A Novel Approach for Organizing Web Search Results using Ranking and Clustering

Neelam Duhan
Department of Computer Engineering
YMCA University of Sc. & Technology
Faridabad, India

A. K. Sharma
Department of Computer Engineering
YMCA University of Sc. & Technology
Faridabad, India

ABSTRACT
World Wide Web is considered the most valuable place for Information Retrieval and Knowledge Discovery. While retrieving information through user queries, a search engine results in a large and unmanageable collection of documents. Web mining tools are used to classify, cluster and order the documents so that users can easily navigate through the search results and find the desired information content. A more efficient way to organize the documents can be a combination of clustering and ranking, where clustering can group the documents and ranking can be applied for ordering the pages within each cluster. Based on this approach, in this paper, a mechanism is being proposed that provides ordered results in the form of clusters in accordance with user's query. An efficient page ranking method is also proposed that orders the results according to both the relevancy and the importance of documents. This approach helps user to restrict his search to some top documents in particular clusters of his interest.

Keywords
Document Clustering; PageRank; Web Mining; Weighted PageRank; World Wide Web

1. INTRODUCTION
WWW is one of the popular information resources for text, image, audio, video, and metadata. It is estimated that WWW has expanded by about 2000 % since its inception and is doubling in size every six to ten months [1]. With the rapid growth of information sources available on the WWW and growing needs of users, it is becoming difficult to manage the information on the web and satisfy the user needs. Hence the need to employ some information retrieval techniques to find, extract, filter and order the desired information.

Search engines play a major role in information retrieval and are used by majority of users to find information from WWW. Some commonly used search engines are Google, msn, yahoo search etc. The typical user queries issued to a search engine tend to be imprecise i.e. very short and expressed in an ambiguous manner resulting in a large number of documents generally retrieved in the form of ranked list. Google [2] is an example of this type of search engine. It has been found out that more than 50% of the search engine users consult no more than first two screens of results [3]. To get the required information, the user may have to sift through a very large list of documents displayed by the search engine, posing the problem of information overkill thus necessitating the need to look for alternative techniques for document presentation.

Today, the main challenge in front of a search engine is to efficiently harness web information and present relevant results to the user. Web mining is a potential candidate to meet this challenge as it can mine the WWW for interesting associations or groupings among the web documents leading to better organization of search results.

R. Cooley et al [4] and Dr. M. H. Dunham [5] divide web mining into three categories namely *web content mining, web structure mining and web usage mining.* Web Content Mining emphasis on the content of web page instead of its embedded links. Web Structure Mining is used to generate structural summary about the Web sites and Web pages. Web Usage Mining tries to discover user navigation patterns from web data and the useful information from the secondary data derived from the interactions of the users while surfing on the web.

In this paper, a *Clustering and Ranking* mechanism is being proposed that uses both Web Content as well as Web Structure Mining to represent the documents in a concise manner. In the next section, two important page ranking algorithms: PageRank [6, 7] and Weighted PageRank [8] have been reviewed. It also provides a discussion on document clustering. Section 3 explains the proposed method in detail; while in Section 4, some practical work has been presented to demonstrate the proposed method. Section 5 concludes the paper with some light on future work.

2. RELATED WORK
Most of the search engines use *Page ranking* algorithms, which can arrange the documents in order of their relevance, importance and content score. Some search engines also apply Web Mining techniques such as clustering, classification, association rule discovery and categorization to filter, classify as well as group their search results. Many page ranking algorithms [9, 10] have been proposed in the literature such as *HITS, Clever, PageRank, Weighted PageRank, Page Content Rank.* Some algorithms rely only on the link structure of the documents i.e. their popularity scores (web structure mining), some look for the content of the documents with respect to the user query (web content mining), while others use a combination of both i.e. they use links as well as the content of the document to assign a rank value to the concerned document. Two main page ranking methods and document clustering techniques have been discussed as follows:

2.1 PageRank Algorithm

PageRank [6, 7, 11] was developed at Stanford University by Larry page (cofounder of Google search engine) and Sergey Brin. Google uses this algorithm to order its search results in such a way that important documents move up in the results of a search while moving the less important pages down in its list. This algorithm states that if a page has some important incoming links to it, then its outgoing links to other pages also become important, thus it takes backlinks into account and propagates the ranking through links. When some query is given, Google combines precomputed PageRank scores with text matching scores to obtain an overall ranking score for each resulted web page in response to the query. Although many factors determine the ranking of Google search results but PageRank continues to provide the basis for all of Google's web search tools.

A simplified version of PageRank is defined in (1):

$$PR(u) = c \sum_{v \in B(u)} \frac{PR(v)}{N_v} \quad (1)$$

where u represents a web page, $B(u)$ is the set of pages that point to u. $PR(u)$ and $PR(v)$ are rank scores of page u and v, respectively. N_v denotes the number of outgoing links of page v, c is a factor used for normalization.

In PageRank, rank score of a page p is evenly divided among outgoing links. Values assigned to the outgoing links of page p are in turn used to calculate the ranks of the pages to which page p is pointing. Example showing distribution and assignment of page ranks is illustrated in Figure 1.

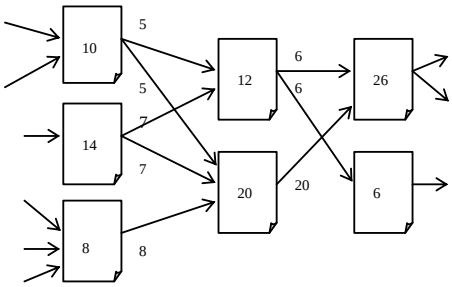

Figure 1. Distribution of PageRank

Later PageRank was modified keeping in view the Random Surfer Model [11] which states that not all users follow the direct links on WWW. The modified version is given in (2).

$$PR(u) = (1-d) + d \sum_{v \in B(u)} \frac{PR(v)}{N_v} \quad (2)$$

where d is a damping factor [6] that is usually set to 0.85. d can be thought of as the probability of users' following the direct links and $(1 - d)$ as the page rank distribution from non-directly linked pages.

2.1.1 Example illustrating working of PR

To explain the working of PageRank, let us take an example hyperlinked structure shown in Figure 2, where A, B and C are three web pages. The PageRanks for pages A, B, C can be calculated using (2) as shown below.

PR(A)= (1-d)+d((PR(B)/2+PR(C)/2) (2a)
PR(B)= (1-d)+d(PR(A)/1+PR(C)/2) (2b)
PR(C)= (1-d)+d(PR(B)/2) (2c)

By calculating the above equations with d=0.5 (say), the page ranks of pages A, B and C become:

PR(A)=1.2, PR(B)=1.2, PR(C)=0.8

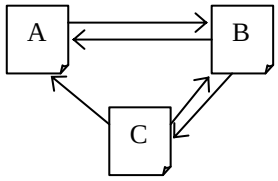

Figure 2. Example Hyperlinked Structure

2.1.2 Iterative Method of Page Rank

It is easy to solve the equation system for a small set of pages to determine the page rank values but the web consists of billions of documents and it is not possible to find a solution by inspection method. In iterative calculation, each page is assigned a starting page rank value of 1 as shown in Table 1 and many iterations could be followed to normalize the page ranks. It may be noted that in this example, PR(A)=PR(B)>PR(C). Experiments have shown that rank value of a page converges to a reasonable tolerance in the roughly logarithmic (log n).

Table 1. Iteration Method of PageRank

Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	1.25	0.81
2	1.21	1.2	0.8
3	1.2	1.2	0.8
4	1.2	1.2	0.8
…	…	…	…

2.2 Weighted PageRank Algorithm

Wenpu Xing and Ali Ghorbani [8] proposed an extension to standard PageRank called *Weighted PageRank (WPR)*. It assumes that more popular the web pages are, more linkages other web pages tend to have to them or are linked to by them. This algorithm assigns larger rank values to more important pages instead of dividing the rank value of a page evenly among its outgoing linked pages. Each outlink page gets a value proportional to its popularity or importance and this popularity is measured by its number of incoming and outgoing links. The popularity is assigned in terms of weight values to the incoming and outgoing links, which are denoted as $W^{in}(v,u)$ and $W^{out}(v,u)$ respectively. $W^{in}(v,u)$ (given in (3)) is the weight of $link(v, u)$ calculated based on the number of incoming links of page u and the number of incoming links of all reference (outgoing linked) pages of page v.

$$W^{in}_{(v,u)} = \frac{I_u}{\sum_{p \in R(v)} I_p} \quad (3)$$

where I_u and I_p represent the number of inlinks of page *u* and page *p*, respectively. $R(v)$ denotes the reference page list of page *v*. $W^{out}(v,u)$ (given in (4)) is the weight of $link(v, u)$ calculated based on the number of outlinks of page *u* and the number of outlinks of all reference pages of page *v*.

$$W^{out}_{(v,u)} = \frac{O_u}{\sum_{p \in R(v)} O_p} \quad (4)$$

where O_u and O_p represent the number of outlinks of page *u* and page *p*, respectively. Considering the weight values, the original PageRank formula (2) is modified as given in (5).

$$WPR(u) = (1-d) + d \sum_{v \in B(u)} WPR(v) W^{in}_{(v,u)} W^{out}_{(v,u)} \quad (5)$$

2.2.1 Example illustrating working of WPR

To illustrate the working of WPR refer again to Figure 2. The Weighted PageRank equations (see (5)) become:

$$WPR(A) = (1-d) + d\,(WPR(B).W^{in}_{(B,A)}.W^{out}_{(B,A)} + WPR(C).W^{in}_{(C,A)}.W^{out}_{(C,A)})$$

$$WPR(B) = (1-d) + d(WPR(A).W^{in}_{(A,B)}.W^{out}_{(A,B)} + WPR(C).W^{in}_{(C,B)}.W^{out}_{(C,B)})$$

$$WPR(C) = (1-d) + d\,(WPR(B).W^{in}_{(B,C)}.W^{out}_{(B,C)})$$

The weights of incoming as well as outgoing links can be calculated as:

$W^{in}(B,A) = I_A/(I_A+I_C) = 2/(2+1) = 2/3$
$W^{out}(B,A) = O_A/(O_A+O_C) = 1/(1+2) = 1/3$

Similarly other values after calculation are:

$W^{in}(C,A) = 1/2$ and $W^{out}(C,A) = 1/3$
$W^{in}(A,B) = 1$ and $W^{out}(A,B) = 1$
$W^{in}(C,B) = 1/2$ and $W^{out}(C,B) = 2/3$
$W^{in}(B,C) = 1/3$ and $W^{out}(B,C) = 2/3$

After substituting d= 0.5 and above calculated weight values, weighted page ranks obtained are:

WPR(A)=0.65 , WPR(B)=0.93, WPR(C)=0.60

It can be seen here that WPR(B)>WPR(A)>WPR(C). It shows that the resulting order of pages obtained by PageRank (Section 2.1) and WPR is different.

It may be noted from the literature that page ranking algorithms have become more and more efficient in order to achieve higher precision, but they have not been made to satisfy the needs of all users in a concise manner. The returned documents should be arranged in a user friendly manner. The paper focuses on other techniques like clustering also to represent the results according to the user needs.

2.3 Document Clustering

Clustering [12] divides a set of objects into groups such that the objects in the same group are similar to each other. In the context of web document clustering [13, 14], objects are replaced by documents and are grouped together based upon some measure like similarity of content or of hyperlinked structure. As discussed earlier, most of the search engines return a large and unmanageable list of documents containing the query keywords. Finding the required documents from such a large list is usually tedious, often impossible. As a solution, the search engines can group a set of returned documents with the aim of finding semantically meaningful clusters, rather than a list of ranked documents. Web clustering may be based on content alone, may be based on both content and links or may only be based on links.

Popular clustering techniques like k-means and the hierarchical clustering [15] can be used for Web document cluster analysis, but these algorithms assume that each document has a fixed set of attributes that must appear in all documents. Similarity between documents can then be computed based on these attribute values. One approach for Web document cluster analysis: Suffix Tree Clustering (STC) [13] uses a phrase-based clustering approach rather than using single word frequency.

It has been found that almost 30% of all web pages are very similar and about 22% are virtually identical to other pages. One can cluster the pages based upon their similarity of content as each document possibly has a set of terms and associated frequencies, which can be used for clustering the pages. Similarity between Web pages usually means content-based similarity [16]. It is also possible to consider link-based similarity and usage-based similarity. Link-based similarity [17] is related to the concept of co-citation and is primarily used for discovering a core set of web pages on a topic. Usage-based similarity is useful in grouping pages or users into meaningful groups. In the proposed work, focus is on content-based similarity which is based on comparing the textual content of the web pages. There are two ways to define content-based similarity between the documents as given below.

1. Resemblance:
Resemblance of two documents is defined to be a number between 0 and 1 with 1 indicating that the two documents are virtually identical and any value close to 1 indicating that the documents are very similar.

2. Containment:
Containment of one document in another is also defined as a number between 0 and 1 with 1 indicating that the first document is completely contained in the second.

A critical look at the available literature highlights the following limitations in the existing approaches of search result organization: First, they give emphasis to links of the resultant pages and the link based calculations do not relate the documents to the user query. Second, no algorithm exists to combine the link score and content score of the page into a single score. Third, most of the existing approaches return millions of documents in an ordered format, whole of which are generally not accessed by users. Finally, all rank based approaches give equal emphasis to inlinks as well as outlinks of pages, which is not true in practice. Therefore, keeping in view these limitations, an efficient ranking and clustering based approach has been proposed which takes advantage of the importance of inlinks over outlinks, as well as provides an efficient user browsable result organization in response to the user query.

3. THE PROPOSED ALGORITHM FOR CLUSTERING AND RANKING

Though most of the search engines are using information retrieval and knowledge discovery tools to filter, order, classify or cluster their search results, still user may have to make extra efforts to find the required documents. There is a need of a navigational tool, which could insure the relevance of the document according to user needs as well as represent the documents in user browsable and understandable manner. As a solution, they can be put into hierarchy of query related clusters, which must be diversified as much as possible. Moreover, the documents in each cluster can be ranked to represent them according to their relevancy. Such organization enables the user to effectively limit his search area looking at the group with higher query-document similarity as well as going through some of the documents contained within interesting groups.

3.1 Proposed Architecture

Traditional search engines work on the basis of matching query keywords with the documents and the presence of the keyword means that document is to be returned to the user. The most important component of a search engine (see Figure 3) is a crawler [2] (also called a robot/spider) that traverses the hypertext structure of the WWW and downloads the web pages. The downloaded pages are routed to an indexing module that builds the index based upon the keywords present in the pages. When a user fires a query in the form of keywords on the interface of a search engine, it is retrieved by the query processor component, which after matching the query keywords with the index returns the URLs of the pages to the user.

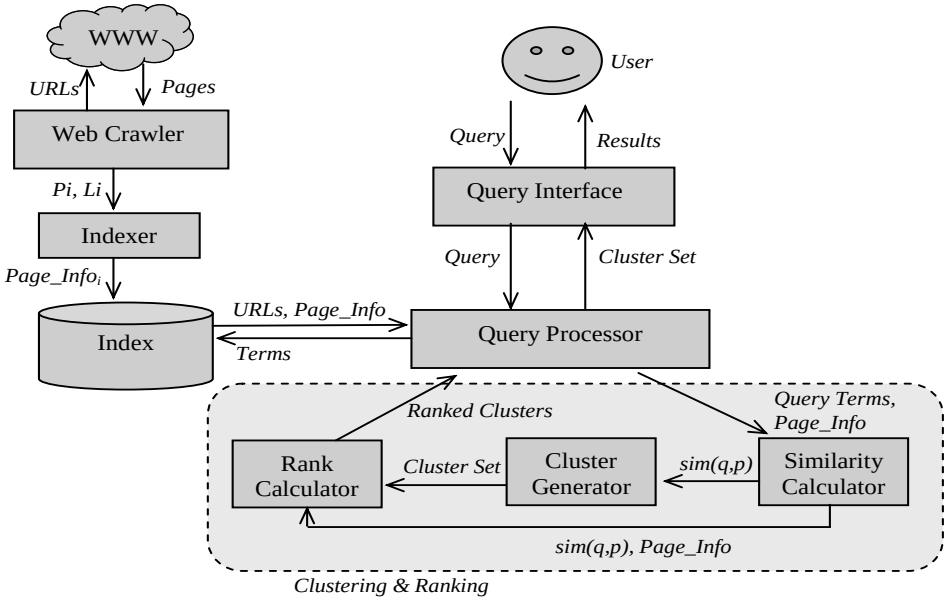

Figure 3. Modified Architecture of Search Engine with Clustering and Ranking Module

The proposed work suggests a *Clustering and Ranking* module to the existing search engine architecture, which further contains three sub-modules as shown in Figure 3. Now in the modified system, crawler downloads the web pages, passes these pages and their link information (in-links and out-links) to the indexer. The indexer in turn builds the index and stores the entire page information (including URL of the page, terms, in-links, out-links, positions of terms, frequency of terms etc.) alphabetically according to the terms present in the page.

The entire process (See Figure 3 and 4) from giving the user query to getting the results can be explained in the following steps:

Step 1: *Get the URLs of the pages:*
When user fires a query to the search engine, *query processor* matches the query terms (after removing prefixes, suffixes, sentence boundaries, non-word tokens, punctuation marks, etc) with the index and gets the URLs of the pages and their entire page information, which it stores in its local database.

Step 2: *Provide a similarity value sim(q, p) to each returned document:*
Query processor passes this page-information and the query terms to the *similarity_calculator* module for finding, how much the page is matching to the user query.

Step 3: *Use sim(q, p) to cluster the documents:*
The similarity values of the pages returned by the *similarity_calculator* module are used by the *cluster_generator* module, which groups the pages into a number of clusters.

Step 4: *Provide a rank score WSR(p) to the documents of each cluster:*

Finally, the *rank_calculator* component works for each cluster to further rank the pages within each cluster using the similarity and the link information. These clusters are resulted to the user by the query processor.

3.2 The Algorithm

The proposed algorithm (see Figure 4) works on the basis of four steps mentioned in the previous sub-section. The working of similarity_calculator, rank_calculator and the cluster_generator components are described below to better understand the algorithm.

3.2.1 Similarity_calculator

Similarity of the document with the query means: what query terms are present in the document, where they are present and how many times? There are many measures to find out the similarity between query and the page, and even between two pages also; but we are using a measure which depends on the weights of the query terms in the user query and in the document. In information retrieval, documents are ranked with these similarity values but we are using this measure for clustering as well as for ranking.

Similarity of a page *p* with reference to the query *q* can be measured with a value called similarity_value denoted by *sim(q, p)* generally lying between 0 and 1. This similarity model [18] is being calculated using cosine between vector of query terms and vector of document terms as given in (6).

$$sim(q,d) = \frac{\sum W_{q,i} * W_{d,i}}{\sqrt{\sum W_{q,i}^2} \times \sqrt{\sum W_{d,i}^2}} \quad (6)$$

where $W_{q,i}$ and $W_{d,i}$ are the weights of term t_i in query q and document d respectively. $W_{q,i}$ can be found by simply counting the number of occurrences of the term t_i in user query, while $W_{d,i}$ is the frequency of the term t_i in the document.

3.2.2 Rank_calculator

The proposed ranking named *WSR (Weight and Similarity based Rank)* considers both back as well as forward links to find the rank of the page as WPR (Weighted PageRank) does; but unlike WPR, WSR does not give equal importance to the inlinks and outlinks to find the weight of a link (see (3) and (4)). Rather it calculates a single weight for a link instead of two. It may be noted that in general, back-links contribute more towards the importance of a page rather than forward-links. WSR follows the same and gives more importance to the inlinks of a page.

The weight of a link (v,u) in WSR is denoted by $W^{link}(v,u)$, which measures the importance of the backlink page *v* of a page *u* based upon the inlinks and outlinks of reference pages of the page *v* as shown in (7).

$$W_{(v,u)}^{link} = \frac{\alpha I_u + \beta O_u}{\alpha(\sum_{p \in R(v)} I_p) + \beta(\sum_{p \in R(v)} O_p)} \quad (7)$$

where I_u and O_u represent the number of inlinks and outlinks of page *u*, *R(v)* is the set of pages pointed out by page *v*. α and β are the constants deciding the importance of inlinks and outlinks respectively. They are not set equal, rather $\alpha > \beta$ (more importance to inlinks) and $\alpha+\beta=1$.

The rank_calculator module can take any value of α and β, but these should be set such that $0.5<\alpha<1$ and $0<\beta<0.5$. For example,

Algorithm: Clustering_with_Ranking
Input: User query *Q*, Set *P* of *n* returned pages, Maximum size of a cluster (*m*).
Output: Cluster set (*C*) with ranked pages in each cluster.
// Start of algorithm

1. Get the page_info of *n* pages.
2. for ($p_i \in P$, $1 \le i \le n$)
   calculate $sim(q,p_i)$ *//Similarity_calculator (uses Q and page_info)*
   $low = \min(sim(q,p_i))$
   $up = \max(sim(q,p_i))$
3. Decide optimal values of α and β. *//Rank_calculator*
4. If (|page_set| ≤ m) *//initially page_set is P & subsequently updated in step 4.*
   Create a single cluster with all pages in the page_set.
   Give a new cluster_id to all the pages of cluster.
   Order all $p \in page\_set$ according to their '*WSR(p)+sim(q,p)*' values.
   C= C ∪ cluster_id; *//initially |C|=0 & C=∅*
   else
   mid= (low+up)/2
   partition the page_set into two sub-sets having $low \le sim(q,p) \le mid$ and $mid \le sim(q,p) \le up$
   for (each sub-set)
   Update *low* and *up* values *//low remains same & up=mid in first sub-set*
   //low=mid & up remains same in second sub-set
   Goto step 4.
5. for (each $Cj \in C$)
   for ($p_i \in Cj$, $1 \le i \le |Cj|$)
   calculate $WSR(p_i)$ *//using link_info and sim(q,r), $r \in$ inlink pages of p*
6. Arrange $Cj \in C$ in descending order of their *low (or up)* values.
7. Return the ordered cluster set *C*.

Figure 4. The Algorithm for Clustering and Ranking Mechanism

a value of α=0.74 says that inlinks of a page u contribute 74% and outlinks of the same page contribute 26% towards the rank of the page u. Experiments to decide the optimal value of α and β are given in Section 4. The numerator in (7) measures the rank contribution to a page u from itself, while denominator gives the contribution from all the backlinked pages of page u. the fraction of both calculates the weight of a link based both upon the inlinks and outlinks.

Now the refined formula for the rank called *WSR* is given in (8).

$$WSR(u) = (1-d) + d(\sum_{v \in B(u)} WSR(v).W^{link}_{(v,u)}.sim(q,v)) \quad (8)$$

This formula provides relevant as well as important documents at the top of the search results because it considers both the links and content of the pages.

3.2.3 Cluster_generator

This component of the search engine is responsible for generating groups of the returned documents. The clustering is purely based on the similarity values of the pages with respect to the user query. The number of clusters is not predefined; they are automatically adjusted depending upon the number of retrieved pages. What this module has to decide is the maximum number of pages (say *m*) that can be in a cluster.

First, the range of similarity values is observed and lower value as well as upper value of similarity is identified. If the number of returned pages is less than or equal to *m*, a single cluster is formed and all pages are assigned the same new cluster_Id; otherwise the similarity range is equally partitioned and complete page set is divided into two sets according to the similarity values of the pages lying within the new partitioned ranges. The condition on the sizes of subsets is checked again and the process of assigning new cluster_Ids is repeated until we get all the returned pages being assigned to any one of cluster.

The complete process of Clustering and Ranking is illustrated by the algorithm given in Figure 4. It may be noted that the pages in each cluster are ordered according to the new rank as given below:

$$Rank(p) = WSR(p) + sim(q, p) \quad (9)$$

This type of ranking considers not only the relevance and importance of the page, rather the relevance of its back-links too. Now the user is returned with a set of clusters with ranked pages within each cluster. It's up to the user that which cluster he examines based on the similarity-ranges and what pages he opens based on the rank values.

4. EXPERIMENTAL RESULTS

Some practical work has been carried out to find the suitable values for the constants α and β. An example scenario is also taken to better understand the work. To illustrate the proposed work, let us take the same hypothetical example of Figure 2.

4.1 Deciding α and β

By applying (7) on the example hyperlinked structure of Figure 2, taking values of α ($0.5<\alpha<1$) and corresponding values of β, the weights of all the links can be calculated as shown in Table 2 (here α and β are taken at some regular intervals to show some of the values).

Table 2. Link Weights wrt different values of α and β.

α	β	$W^{link}(A,B)$	$W^{link}(B,A)$	$W^{link}(B,C)$	$W^{link}(C,B)$	$W^{link}(C,A)$
0.51	0.49	1.000	0.503	0.497	0.570	0.430
0.57	0.43	1.000	0.523	0.477	0.560	0.440
0.63	0.37	1.000	0.543	0.457	0.551	0.449
0.69	0.31	1.000	0.563	0.437	0.542	0.458
0.75	0.25	1.000	0.583	0.417	0.533	0.467
0.81	0.19	1.000	0.603	0.397	0.525	0.475
0.87	0.13	1.000	0.623	0.377	0.517	0.483
0.93	0.07	1.000	0.643	0.357	0.509	0.491
0.99	0.01	1.000	0.663	0.337	0.501	0.499

The results can be shown graphically (see Figure 5) to represent the variation of link weights corresponding to α and β. It can be seen that the link weight in each case is either increasing or decreasing, but their coinciding range i.e. where all the weights get stable serves as a suitable measure to find value of α and β, here it comes out to be α= *0.75 to 0.80* and β= *0.25 to 0.20*. Another way to decide α and β is to find the centroids of the area covered by each graph and taking mean of the corresponding α values. Rank_calculator uses these optimal values to calculate the rank of the pages.

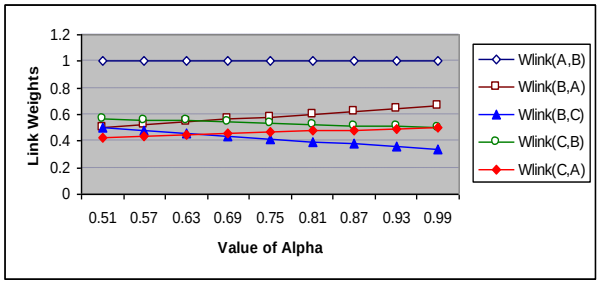

Figure 5. Variation of link weights with respect to α (and β)

4.2 Calculation of Similarity Values

From the practical point of view, here the results on an example query are analyzed. The similarity values show how much a document is relevant to the given query. Let us consider the assumptions given below and in Table 3:

User Query q= Data Mining Techniques for Data Warehouses
Total number of keywords in the query= 5

Table 3. Sample Values for a Hypothetical Example

Page	*Total Terms*	*Freq (data)*	*Freq (warehouse)*	*Freq (mining)*	*Freq (technique)*
A	1000	25	10	5	2
B	2000	25	0	5	3
C	500	10	5	2	0

By using (6), following values are obtained:

Sim(q, A)= 0.92, Sim(q, B)= 0.85, Sim(q, C)= 0.89

These values show that page A is most and page B is least similar to the query. For a large number of returned documents (usually greater than *m*), clusters can be formed by using the cluster_generator algorithm.

4.3 Rank Calculations

The link weights and similarity values are used to find the rank of the returned pages (here A, B and C) by using (8). The resultant rank equations can be solved by iteration method. Taking *d=0.5*, α*= 0.78*, β*=0.22 (*say*)* and substituting weights and similarity values calculated before, the ranks obtained are as given below:

WSR(A)= 0.920, WSR(B)= 1.088, WSR(C)= 0.697

These rank values can easily be differentiated and ordered because they are not very close to each other, but this was not possible in *PR* (Section 2.1) and *WPR* (Section 2.2). Here it may be noted that WSR(B)>WSR(A)>WSR(C).

If we let the maximum size of a cluster to be 2 pages (i.e. m= 2), then two clusters are formed with pages A and C assigned to first (most similar cluster) and B to second (less similar cluster). Within the cluster, they are ordered based on both their similarity and rank. The actual rank of the pages in the cluster is given below.

Rank(A)= WSR(A) + sim(q, A)= 1.84
Rank(B)= 1.938
Rank(C)= 1.587

Therefore C= {C1, C2}, where:

C1= {A, C} with $0.92 \leq sim(q,p) \leq 0.885$, Rank(A)>Rank(C)
C2= {B} with $0.885 < sim(q,p) \leq 0.85$

In actual, where, a large number of documents are returned, significant results can be obtained.

4.4 Comparison of Page Ranking Algorithms

A critical look at the available literature highlights several differences in the basic concepts used in each ranking algorithm. The proposed *Clustering and Ranking* method finds groups as well as ranks of the returned pages to organise them in an efficient and user friendly manner as opposed to the ordered list returned by PageRank and WPR algorithms. If only ranking is concerned, the proposed WSR algorithm is an iterative algorithm but unlike PR and WPR, it uses both the link structure and content of the returned documents and as a result, it returns relevant as well as important pages on the top of the list. Besides considering similarity of the page itself, it also uses the similarity of the back-link pages to find the rank of the concerned page. The comparison summary of the three ranking algorithms PR, WPR and WSR is given in Table 4.

The proposed mechanism was tested on the results obtained from the Google for several different queries (See Appendix A). It may be noted that Google combines the text matching scores to give a page its final order. But the similar pages were seen not to lie in the top of the resultant list. According to the proposed algorithm, more similar pages can be grouped into the same cluster and ordered over there with the new ranking mechanism, thus reducing the search space.

5. CONCLUSION

The usual search engines generally result a large number of pages in response to user queries, while the user always wants to get the best in a short span of time so he/she does not bother to navigate through all the pages to get the required ones. The proposed Clustering and Ranking mechanism gives a way to organize the search results in the form of clusters, the pages in each cluster are further ranked to provide the most relevant and important pages on the top of the cluster. The clustering is purely based on similarity measure, while ranking is based on similarity as well as on the link information of the page. Obviously, the proposed method takes extra time and space than PR and WPR because of extra calculations, but relevancy and importance of the returned results compensate this extra effort. User search space also decreases and he can get the required content in short time. As a future guidance, some history mechanism can be incorporated in

Table 4. Comparison of Ranking Methods

Algorithm→ **↓Parameters**	**PageRank (PR)**	**Weighted PageRank (WPR)**	**Weight & Similarity based Rank (WSR)**
Main Technique Used	Web Structure Mining	Web Structure Mining	Web Structure and Content Mining
Description	Computes scores at indexing time not at query time. Results are sorted according to importance of pages.	Computes score at indexing time, unequal distribution of score , pages are sorted according to importance	Computes scores at query time and used to rank the pages after cluster formation. Relevant as well as important pages are returned.
I/P Parameters	Backlinks	Backlinks, forward links	Backlinks, forward links, α and β *, similarity values
Working levels	N*	1	1
Complexity	O(log N)	< O(log N)	>O(log N)
Relevancy	Less	Less (higher than PR)	High
Importance	High	High	High
Quality of result	Yes	Higher than PR	Very high
Limitations	Equally distributes rank to the outlink pages, Considers only the importance of pages.	Relevancy is ignored, inlinks and outlinks are considered equally.	More time and space complexity because of computing ranks on the fly.

*N: number of web pages, α is inlink and β is outlink constant.

to the Clustering and Ranking method, which stores the user queries and resultant cluster summaries in a local database such that next time, less efforts are made for the identical user queries.

6. APPENDIX-A

The appendix shows the analysis of a user query on Google by taking first 25 resultant pages. The analysis was done in March, 2010. The query fired is:

Q=data mining

Results of Google are analyzed and it can be seen that more similar pages lie downwards in the list (bold and underlined similarity values) and mostly unseen by the user. These pages can be grouped in the same cluster and ranked over there by the proposed approach.

7. REFERENCES

[1] Naresh Barsagade, "Web usage mining and pattern discovery: A survey paper". CSE 8331, Dec, 2003.

[2] http://www.google.com/technology/index.html.

[3] A. Spink, D. Wolfram, B.J. Jansen, T. Saracevis, "Searching the Web: The public and their queries". Journal of the American Society for Information Science and Technology 52 (3), 2001, 226-234.

[4] R.Cooley, B.Mobasher and J.Srivastava, "Web mining: Information and pattern discovery on the World Wide Web". In 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), 1997.

[5] M. H. Dunham, Companion slides for the text, "Data mining: Introductory and advanced topics". Prentice Hall, 2002.

[6] L. Page, S. Brin, R. Motwani, T. Winograd, "The pagerank citation ranking: Bringing order to the web". Technical report, Stanford Digital Libraries SIDL-WP-1999-0120, 1999.

[7] C. Ridings and M. Shishigin, "Pagerank uncovered". Technical report, 2002.

[8] Wenpu Xing and Ali Ghorbani, "Weighted PageRank Algorithm". Proceedings of the second annual conference on Communication Networks and Services Research (CNSR'04), 2004 IEEE.

[9] Neelam Duhan, A. K. Sharma, Komal Kumar Bhatia, "Page Ranking Algorithms: A Survey". In proceedings of the IEEE International Advanced Computing Conference (IACC), 2009.

[10] Jaroslav Pokorny, Jozef Smizansky, "Page Content Rank: An approach to the Web Content Mining".

[11] http://pr.efactory.de/e-pagerank-algorithm.shtml

[12] J. Han, M. Kamber, Data Mining: Concepts and Techniques. Academic Press, London, Morgan Kaufmann Publishers, San Francisco.

[13] O. Zamir, O. Etzioni. "Web document clustering: A feasibility demonstration". Proceedings of the 19th International ACM SIGIR Conference on Research and Development of Information Retrieval (SIGIR'98), 46-54, 1998.

[14] Hiroyuki Toda, Ryoji Kataoka, "A search result clustering method using informatively named entities". WIDM 2005.

[15] D. J. Lawrie and W. B. Croft, "Generating hierarchical summaries for web searches". Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, 2003.

[16] Taher H. Haveliwala, Aristides Gionis, Dan Klein, Piotr Indyk, "Evaluating strategies for similarity search on the Web". WWW2002, May, 2002, Honolulu, Hawaii, USA.ACM 1-58113-449-5/02/0005.

[17] J. Kleinberg, "Authorative sources in a hyperlinked environment". Proceedings of the 23rd annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998.

[18] Miguel Gomes da Costa Júnior, Zhiguo Gong, "Web Structure Mining: An introduction". Proceedings of the IEEE International Conference on Information Acquisition, 2005, China.

Appendix A. Results of the query 'data mining' and Similarity Variations

Google order	URL*(p)*	Google PR	freq (data)	Freq (mining)	Sim *(p,q)*
1	http://en.wikipedia.org/wiki/Data_mining	6	226	78	0.899109
2	http://books.google.co.in/books?id=QTnOcZJzlUoC&printsec=frontcover&dq=data+mining&source=bl&ots=3fmEfoRlRg&sig=cYgoixo2zWDkFtXrcqxnfA8TT1o&hl=en&ei=xnafS8S6FYGzrAfr6viEDg&sa=X&oi=book_result&ct=result&resnum=2&ved=0CBkQ6AEwAQ	0	19	5	0863779
3	http://datamining.typepad.com/	6	49	27	0.960564
4	http://www.thearling.com/text/dmwhite/dmwhite.htm	4	29	26	**0.998516**
5	http://books.google.co.in/books?id=AfL0t-YzOrEC&printsec=frontcover&dq=data+mining&source=bl&ots=UuYZxQ7sA0&sig=nw9acDQ_hBhyyN0woSFeaTF4olE&hl=en&ei=xnafS8S6FYGzrAfr6viEDg&sa=X&oi=book_result&ct=result&resnum=6&ved=0CCQQ6AEwBQ#v=onepage&q=&f=false	0	20	6	0.880471
6	http://www.oracle.com/technology/products/bi/odm/index.html	6	86	45	0.95435
7	http://www.autonlab.org/tutorials/	4	7	5	0.986394
8	http://books.google.co.in/books?id=SNDfRPYomLYC&printsec=frontcover&dq=data+mining&source=bl&ots=sWNXlhZVX7&sig=qNQU1o1WH7LHQ-0QPPrgMlux-6s&hl=en&ei=xnafS8S6FYGzrAfr6viEDg&sa=X&oi=book_result&ct=result&resnum=9&ved=0CCwQ6AEwCA#v=onepage&q=&f=false	0	16	5	0.885832
9	http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm	5	20	5	0.857493
10	http://infolab.stanford.edu/~ullman/mining/mining.html	5	2	2	**1**
11	http://searchsqlserver.techtarget.com/sDefinition/0,,sid87_gci211901,00.html	5	75	55	0.988372
12	http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm	5	58	26	0.934488
13	http://infolab.stanford.edu/~ullman/mining/mining.html	5	34	0	0.707107
14	http://www.google.co.in/search?q=data+mining&hl=en&tbs=blg:1&tbo=u&ei=HnufS_y3MNK4rAeo4bm1Dg&sa=X&oi=blogsearch_group&ct=title&resnum=11&ved=0CDgQrgQwCg	0	24	5	0.836461
15	http://www.megaputer.com/data_mining.php?gclid=CL2IjsukvaACFQowpAodwyATTg	0	3	3	**1**
16	http://searchsqlserver.techtarget.com/sDefinition/0,,sid87_gci211901,00.html	5	79	27	0.897789
17	http://www.thearling.com/	5	85	58	0.982638
18	http://www.the-data-mine.com/	5	66	60	**0.998868**
19	http://www.dmoz.org/Computers/Software/Databases/Data_Mining/	5	13	80	0.811369
20	http://databases.about.com/od/datamining/a/datamining.htm	4	5	1	0.83205
21	http://www.datamining-conf.org/	5	84	53	0.975342
22	http://www.statsoft.com/textbook/data-mining-techniques/	5	12	79	0.805278
23	http://www.intelegencia.com/	3	41	8	0.829437
24	http://www.exforsys.com/tutorials/data-mining.html	4	47	34	**0.987364**
25	http://www.zementis.com/?gclid=CPLfl5qmvaACFYcvpAodyRjJsw	0	2	2	**1**