Web Document Clustering and Visualization Results of Semantic Web Search Engine Using V-Ranking

S. K. Jayanthi and S. Prema

Abstract— **As the number of available Web pages grows; it is become more difficult for users finding documents relevant to their interests. Clustering is the classification of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. Because of the short lengths of queries, approaches based on keywords are not suitable for document clustering. This paper describes a new Web Document Clustering method that makes use of user logs which allow identifying the documents the users have selected for a query. The similarity between two queries may be deduced from the common documents the users selected for them. This research paper show that a combination of both content based and session based clustering [1] is better than using either method alone. The clustered documents are arranged based on V-Ranking. In this research work, it has been proposed to display the result in visual mode of semantic search engine using V (Visual) - Ranking algorithm and bookshelf data structure. This paper proposes a semantic web search results in visualize web graphs, representations of web structure overlaid with information and pattern tiers by providing the viewer with a qualitative understanding of the information contents.**

Index Terms— **Book Shelf Data Structure, Content based Clustering, Session Based clustering, Visualization, (Visual)-Ranking.**

I. Introduction

Content based clustering affect the precision of the search engines. In many cases, the answers returned by search engines are not relevant to the user's information need, although they do contain the same keywords as the query. These new systems try to "understand" the user's question i.e. it should understand the sense of human beings in order to suggest similar questions that other people have asked and for which the system has the correct answers.

This research paper proposes a new approach to query clustering based on user login entry i.e. session based clustering. If user clicked on the same documents for different queries, then these queries are similar. If a set of documents is often selected for the same queries, then the terms in these documents are; to some extent, related to the terms of the queries. This study demonstrates the usefulness for a search engine, of session based clustering.

In this paper it is proposed a clustering based approach to support the comprehension of web applications. The approach is based on a clustering process that first computes the dissimilarity between the web pages using Latent Semantic Indexing, a well known information retrieval technique, and then group's similar pages. To automate the clustering process a prototype has been also implemented. The results obtained by applying the different clustering algorithms on the static pages of three web applications developed using JSP technology.

Documents clustered based on both content based and session based clustering are ordered based on V-Ranking. It is named so because the ordered documents are arranged in the shelf of book shelf data structure.Finallly the web search result is executed in visual mode. So, the ranking algorithm is named as visual ranking.

The main focus of this paper is the processing of the results coming from an information retrieval system. Although the relevance depends on the results quality, the effectiveness of the results processing represents an alternative way to improve the relevance for the user. Given the current expectations this processing is composed by an organization step and a visualization step. Then the proposed approach organizes the results according to their meaning using a Bookshelf Data Structure, and visualizes [2] them in a 3D scene to increase the representation space. This paper deals with the processing of query results. This processing, still neglected in some information retrieval systems, is becoming more and more important and essential. The two main points to reach this goal are a good document organization and an effective visualization. Concerning these two aspects, the main directions of this paper are a Clustering method and a 3D visualization.

II. Review of Work Related To Document Clustering

The first group of related clustering approaches is certainly those that cluster documents using the keywords they contain. In these approaches, in general, a document is represented as a vector in a vector space formed by all the keywords [3]. As to clustering algorithms, there have been mainly two groups: hierarchical and non-hierarchical. Hierarchical agglomerative clustering (HAC) algorithm and k-means are representatives of the two groups [4]. Special attention is paid to such words in question answering (QA) [5] [6], where they are used as prominent indicators of question type. Because of the limitations of keywords, people have been looking for additional criteria for document clustering. One of them is the hyperlinks between documents.

Manuscript received October 11, 2010; revised January 5, 2011. This work was supported in part by the Department of Computer Science, K.S.R College of Arts and Science and Research work under Vellar College for Women, Bharathiar University.

Dr. S.K.Jayanthi is with Computer Science Department as Associate Professor and Head , Vellalar College for Women (Autonomous), Erode, Tamilnadu, India(e-mail: jayanthiskp@gmail.com).

S.Prema is with Computer science Department as Asst.Prof, K.S.R. College of Arts and Science,Tiruchengode-637215, Namakkal district, Tamilnadu, India . (e-mail: prema_shanmuga@yahoo.com).

The hypothesis is that hyperlinks connect similar documents. This idea has been used in some early studies in IR[7][8].More recent examples are Google (http://www.google.com) and the authority/hub calculation of Kleinberg [9]. Relevance feedback in IR is a typical exploitation of cross-references. It is typically used to reformulate the user's query [3]. It is also suggested that relevance feedback may be used as follows: if two documents are judged relevant to the same query, then there are reasons to believe that these documents talk about the same topic, and therefore can be included in the same cluster.

III. Document Clustering

Document Clustering is a process used to discover frequently asked questions or most popular topics on a search engine. This process is crucial for search engines based on question-answering. Because of the short lengths of queries, approaches based on keywords are not suitable for query clustering. This paper proposes session based clustering that makes use of user login entry which allows the search engine to identify the documents the users have selected for a query as in Fig. 1.

A. Content based clustering:

If two queries contain the same or similar terms, they denote the same or similar information needs. Obviously, the longer the queries, content based clustering is more reliable. However, as queries are short, this principle alone is not sufficient.

Keywords are all words, except function words included in a stop-list. All the keywords are stemmed using the Porter algorithm [10]. The keyword-based similarity function is defined as follows:

$$\text{similarity}_{\text{content}}(a,b) = \frac{C(a,b)}{Max(c(a),c(b))}$$

where c (.) is the number of the keywords in a query, c (a, b) is the number of common keywords in two queries.

If query terms are weighted, the cosine similarity [3] can be used instead:

$$\text{similarity}_{\text{w\_content}(a,b)} \frac{\sum_{i=1}^{k} dw_i(a) \times dw_i(b)}{\sqrt{\sum_{i=1}^{m} w_i^2(a)} \times \sqrt{\sum_{i=1}^{n} w_i^2(b)}}$$

where $dw_i(a)$ and $dw_i(b)$ are the weights of the i-th common keyword in the query a, and b respectively and $w_i(a)$ and $w_i(b)$ are the weights of the i-th keywords in the query a and b respectively. tf * idf is used for keyword weighting. The above measures can easily be extended to phrases. Since phrases are a more precise representation of meaning than a single word, the user can obtain a more accurate calculation of query similarity. For example, the two queries "operating systems" and "Unix" are very close queries. Their similarity is 0.33 on the basis of keywords. If the user recognizes "Operating Systems" as a phrase and takes it as a single term, the similarity between these two queries is increased to 0.5. The calculation of phrase-based similarity is similar to noun phrase recognizer based on syntactic rules and statistics [11][12]. Another way is to use a phrase dictionary. In Encarta, there is such a dictionary, containing a large number of phrases and proper nouns that appear in Encarta documents. In the future, it will be supplemented by an automatic phrase recognizer based on activity syntactic and statistical analysis.

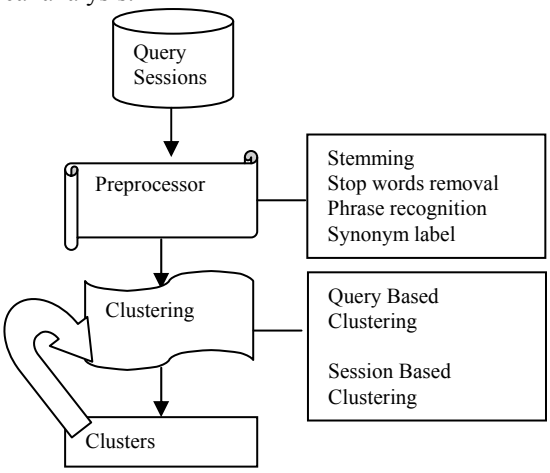

Figure 1. Flow chart of the clustering process

B. Session-based query clustering

Session-based query clustering is a strategy, which employs the evidences in query sessions to deduce and detect users' search intentions. Many search engines have accumulated a large amount of web query logs, from which one can find out what the query is, and the web pages the user has selected to browse. Typically, a query session is made up of the query and the subsequent activities the user performed, which can be extracted from query logs.

Let D(a) and D(b) be the set of documents the system presents to the user as search results for the queries a and b respectively. The document set that users clicked on for the queries a and b may be seen as follows:

$$\text{D-C (a)} = \{d_{a1}, d_{a2}, \ldots, d_{ai}\} \subseteq \text{D (a)}$$
$$\text{D-C (b)} = \{d_{b1}, d_{b2}, \ldots, d_{bj}\} \subseteq \text{D (b)}$$

Similarity based on session based follows the following principle. If $\text{D\_C (a)} \cap \text{D\_ C (b)} = \{d_{ab1}, d_{ab2}, \ldots, d_{abk}\} \neq \phi$, then documents $d_{ab1}, d_{ab2}, \ldots, d_{abk}$ represent the common topics of queries a and b. Therefore, a similarity between the queries a and b is determined by $\text{D\_C (a)} \cap \text{D\_C (b)}$.

Similarity through Document Hierarchy:

Let F (di, d j) denote the lowest common parent node for documents di and d j , L(x) the level of node x, L_ Tot the total levels in the hierarchy . The conceptual similarity between two documents is defined as follows:

$$s(d_i,d_j) = \frac{L(F(d_i,d_j))-1}{L\_Tot-1}$$

Let $d_i (1 \leq i \leq m)$ and d_j $(1 \leq j \leq n)$ be the clicked documents for queries a and b respectively, and nd(a) and nd(b) the number of document clicks for each query. The hierarchy-based similarity is defined as follows:

$$similarity_{session\_based}(a,b)=\frac{1}{2}?\left(\frac{\sum_{i=1}^{m}\left(max_{j=1}^{n}s(di,dj)\right)}{nd(a)}\quad\frac{\sum_{j=1}^{n}\left(max_{i=1}^{m}s(di,dj)\right)}{nd(b)}\right)$$

C. Combined Measures

Content-based and Session-based document clustering should be defined to take advantage of both strategies. A simple way to do this is to combine both measures linearly, as follows:

$similarity=\alpha*similarity_{content\text{-}based}+\beta*similarity_{session\text{-}based}$

IV. Bookshelf Data Structure

Bookshelf data structure [13] as in Fig. 2 has been introduced for community formation, which stores the inverse indices of the WebPages. This data structure is formed by combining a matrix and list with dynamically allocated memory. This is an extended data structure of hash table and bi-partite core [5], which is used to store base domain and sub-domain indices of various communities. A recent study [5] shows that 81.7%of users will try a new search if they are not satisfied with the listings they find within the first 3 pages of results. However it would be too restrictive to only consider the first 30 results (10 results per page). Indeed this study has been done on search engines with linear results visualization (ordered lists) and users may want to see more results on visualizations like web graphs [14].

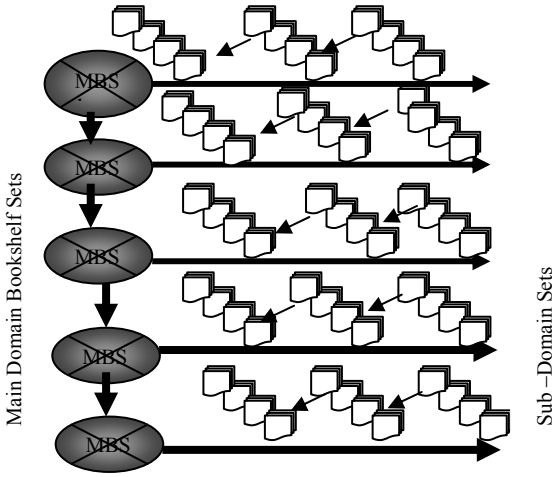

Figure 2. Bookshelf Data structure

V. V-Ranking

The celebrated PageRank algorithm (Brin and Page, 1998) is a method for ranking the vertices in a graph according to their relative structural importance. The main idea of PageRank is that whenever a link from vi to vj exists in a graph, activity vote from node i to node j is produced, and hence the rank of node j increases. Besides, the strength of the vote from i to j also depends on the rank of node i: the more important node i is, the more strength its votes will have. Alternatively, PageRank can also be viewed as the result of a random walk process, where the final rank of node i represents the probability of a random walk over the graph ending on node i, at a sufficiently large time. Let G be a graph with N vertices $v_1, \ldots, v_n$ and d_i be the outdegree of node i; let M be a N×N transition probability matrix, where$M_{ji}=1/d_i$ if a link from i to j exists, and zero otherwise. Then, the calculation of the Page Rank vector Pr over G is equivalent to resolving Equation P=cMPr+(1-c)v, v is a N × 1 vector whose elements are 1/N and c is the so called damping factor, a scalar value between 0 and 1.The damping factor, usually set in the [0.85..0.95] range, models the way in which these two terms are combined at each step. Directed graph representing web of 6 pages is shown in fig.3

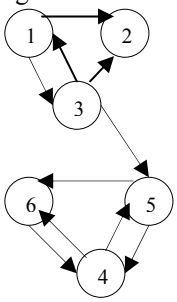

Figure 3. Directed graph representing web of 6 pages

For clustering the initial preference or boundary, condition should be specified, Hubbell's interest is clique detection, an early study of spectral graph clustering [15].The index r represents the relationships between input and output of goods in each industry.r=i+r m,Where i is a initial preference or boundary condition.

The pseudo code of V-Ranking algorithm for effective web search results based on Java Language [16] using the packages util (HashMap , Iterator, List , Map) and Jama (Matrix) is described below.

```
V- Ranking
Initialize Damping Factor as 0.85
Assign  List params = new ArrayList();
begin
   Ranking ranking = new Ranking();
   Print  ranking.rank("C");
end

   /*To Solve the equation of ax=b, a is the generated
matrix. X is the page ranks matrix. b is a n*1 matrixwhich all
the values are equal to the damping factor. */

   Rank(String pageId)
begin
   generateParamList(pageId);
   Matrix a = new Matrix (generateMatrix());
   Initialize parameter size to array B;
   Repeat  for  i = 0 and  i < params.size()
   Assign arrB[i][0] = 1 - Damping_Factor
end
Matrix b = new Matrix (arrB)     // To get the page rank
Begin
Initialize Matrix x = a.solve (b)
   Initialize index and count value as zero
Iterate till param=null
   Check current referencePage with next related page
```

```
    If current page equals pageid then
    Assign count value to index
    Increment count value
    Return value
  end
    /* To returns list of the related pages  */
  generateParamList(pageId)
    if  parameter value ! = pageId
    then add  pageId
    String[] inc = getInboundLinks(pageId)
   // Get list of the inbound pages
  // Add the inbound links to the params list and do same for
inbound links
    for (int i = 0; i < inc.length; i++)
  begin
    if  parameter value ! in inbound range then
    generateParamList(inc[i])
  end
    // Return list of the inbound links to a given page.

getInboundLinks(String pageId)    //This simulates a simple
page collection
  begin
  assign map value to  new HashMap()
    map the web pages 'A','B',C'
    return map(pageId)
  end

  // Returns list of the outbound links from a page.
  getOutboundLinks(pageId) // This simulates a simple page
collection
  begin
  assign map value to  new HashMap()
  map the web pages 'A','B',C'
  return map(pageId)
  end
```

VI. Websites As Graphs

The user after analyzing the result of semantic web search prefer to produce the result as web graph as in Fig.4 with color specification for nodes like blue: for links (the A tag),red: for tables (table, tr and td tags),green: for the DIV tag, violet: for images (the IMG tag),yellow: for forms (form, input, text area, select and option tags),orange: for line breaks and block quotes (br, p, and blockquote tags),black: the HTML tag, the root node, gray: all other tags .

VII. Conclusion

In this paper it has been presented a clustering based approach to identify pages similarity at the content level. The approach is based on a process that first computes the dissimilarity between web pages using LSI and then groups' similar pages using clustering algorithms that have been widely employed in the past to comprehend legacy web applications. For clustering of documents both content based clustering and session based clustering techniques is used. The clustered documents are arranged in bookshelf data structure for effective and easy information retrieval. The clustered documents are ranked using V (Visual) Ranking algorithm and the final result is displayed in visual mode. To automate the identification of groups of similar pages, the approach has been implemented in a Java prototype. This paper proposes an effective method for organizing and visualizing web search results.

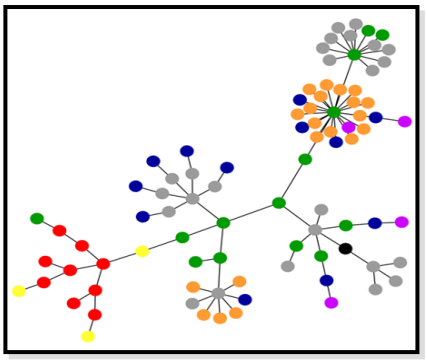

Figure 4.Web graph simulated result of search engine results

References

[1] Ji-rong Wen,Jian-Yun Nie,and Hong-Jiang Zhang ," Query Clustering Using User Logs",*ACM Transactions on Information Systems*, Vol. 20, No. 1, January 2002, Pages 59–81

[2] S.K.Jayanthi and S.Prema,"Facilitating Efficient Integrated Semantic Web Search with Visualization and Data Mining Techniques", International Conference on Information & Communication Technologies, Germany: Springer-Verlag.September, 2010, pp. 437 – 442.

[3] G.Salton and M.J. McGill 1983, *Introduction to Modern Information Retrieval,* McGraw-HillNew York, NY.

[4] R.C.Dubes, and A.K.Jain ," Algorithms for Clustering Data", Prentice-Hall, Englewood Cliffs, NJ. 1988.

[5] V. A .Kulyukin, K. J. Hammond, and R. D. BURKE,"Answering questions for an organization online", *In Proceedings of AAAI 98*. 532–538. 1998.

[6] R.Srihari, and W.LI, "Question answering supported by information extraction", *In Proceedings of TREC8*, 75–85. 1999.

[7] E.Garfield, *Citation Indexing: Its Theory and Application in Science, Technology and Humanities*, 2nd ed. The ISI Press, Philadelphia, PA. 1983.

[8] M. M. Kessler," Bibliographic coupling between scientific papers". *In American Documentation, 14*, 1, 10–25. 1963.

[9] J.Kleinberg, "Authoritative sources in a hyperlinked environment". *In Proceedings of the9th ACM SIAM International Symposium on Discrete Algorithms*. ACM Press, New York, NY, 1998,pp.668–677.

[10] M.Porter, An algorithm for suffix stripping. *Program*, *14*, *3*, 1980. pp. 130–137.

[11] E. De Lima, and J.Pedersen," Phrases recognition and expansion for short, precision-biased queries based on a query log ". *In Proceedings of the 22nd Annual International ACM SIGIRConference on Research and Development in Information Retrieval*. ACM Press, New York, NY, 1999,pp.145–152.

[12] D. D. Lewis, and W. B. Croft, "Term clustering of syntactic phrases". *In Proceedings of the13th Annual International ACMSIGIR Conference on Research and Development in Information Retrieval.* ACM Press, New York, NY, 1990,pp. 385–404.

[13] Vijaya kathiravanan," E-mail id harvester to retrieve E-mail addresses of domain experts" ,In: National Conference On Current Trends in Computer Applications ,2009.

[14] S.K.Jayanthi and S.Sasikala ,"Link Spam Detection based on DBSpamClust with Fuzzy C-Means Clustering", *Journal.* International Journal of Next-Generation Networks (IJNGN), Vol IV Dec 2010,pp.1-8.

[15] H. Charles Hubbell," An input-output approach to clique identification.Sociometry", 28(4):377-399, 1965.

[16] http://www.javadev.org/fles/ranking.zip.

Dr.S.K.Jayanthi received the M.Sc., M.Phil., PGDCA, Ph.D in Computer Science from Bharathiar University in 1987, 1988, 1996 and 2007 respectively. She is currently working as an Assosiate Professor, Head of the Department of Computer Science in Vellalar College for Women. She secured District First Rank in SSLC under Backward Community.

Her research interest includes Image Processing, Pattern Recognition and Fuzzy Systems. She has guided 18 M.Phil Scholars and currently 4 M.Phil Scholars and 4 Ph.D Scholars are pursuing their degree under her supervision. She is a member of ISTE, IEEE and Life Member of Indian Science Congress. She has published 5 papers in International Journals and one paper in National Journal and published an article in Reputed Book. She has presented 14 papers in International level Conferences/Seminars, 16 papers in National level Conferences/Seminars and participated in around 35 Workshops/Seminars/Conferences/FDP.

S.Prema, currently working as a Assistant Professor in K.S.R. College of Arts & Science has received the B.Sc., M.C.A., M.Phil., from the Periyar University in 2001,2004,2008 respectively and now pursuing Ph.D in computer science at Bharathiar University. Her area of Doctoral research is Web mining. She has been a First rank holder under Periyar University in B.Sc Programme.

She has published one paper in international journal of Advances in Computational Sciences and Technology (ACST). She has presented 10 papers in International Conferences/Seminars, 18 papers in National Conferences/Seminars and participated in 4 National Conferences/Seminars, 7 workshops and 1 FD