THE INTERNATIONAL JOURNAL OF
SCIENCE & TECHNOLEDGE

A Novel Approach for Improving Results of Search Engine using Query Log, Support Factor and Clustering

Sneha Tuteja
B.Tech. Student, Department of Computer Science & Engineering, MSIT, New Delhi, India
Naresh Kumar
Assistant Professor, Department of Computer Science & Engineering, MSIT, New Delhi, India
Dr. Rajender Nath
Professor, Department of Computer Science & Applications
Kurukshetra University, Kurukshetra, Haryana, India

Abstract:
Enhancing the result standards of Web search and user experience, lately apprehended a lot of interest from the researchers. This paper intends to present a novel method for refining the search engine results corresponding to a user query. The proposed system makes the use of click through data to improve the present rankings of the Web pages and making it easier to distinguish the relevant Web pages from the irrelevant ones which would result in greater allegiance by the users. This is achieved by using the query log of a search engine. This paper provides a method to calculate the similarity of a query with the available Web pages and then cluster them accordingly. To achieve higher precision of results, the clusters are re-arranged once more.

Keywords*: Query, Query Log, Search Engine, Query Similarity, Cluster, Query Clustering, Hits and Page Rank.*

I. Introduction

Billions of users interact with the Internet on a day-to-day basis [1]. They formulate and issue their queries to the search engine, traverse through the search results, click some pages and reformulate their queries. Most of the time they are not able to find the desired information within the top ranked documents displayed. This requires users to refine their queries again and again [2]. But this query refinement is limited to 2-3 words only [3] [4] [5] that does not convey the users' search interest to the search engine. It results in plenty of irrelevant information offered to the Internet users. This keeps them far from the actual results causing the unsatisfaction with the performance of existing search engine results [5][6]. This problem is referred as information overkill [7]. So, the brief queries are unable to provide any meaningful, relevant and desirable information about the users [8]. But according to [9], query log files helps in achieving the desirable information about the user interest. Nowadays, lots of Web applications are working for the prediction of users' navigational behavior by using Web log mining but, very few of them are working for enhancing the search engine results.

This paper proposes an approach for improvising search engine results by using the concept of similarity, clustering and support factor. Similarity is used to find out similar queries among the large number of queries. Clusters are used to group the similar queries at a single place. Support factor is used to rank the resultant links inside a cluster. The main aim of the proposed work is to improve the current ranking algorithms which would certainly meet the user's expectations of higher relevancy. The rest of the paper is organized as: section II describes the related work, section III explain the problems available with the results optimization techniques. Section IV describe the working of proposed architecture. Section V provides the process of experimental set up and their corresponding achieved results and section VI discusses the achieved results and section VII, finally conclude the paper.

2. Related Work

A traditional rank optimization technique uses keywords for clustering the queries. It argues that the queries that contain exact keywords are likely the same and would be satisfied by the same set of Web documents. Authors Neelam Duhan and A.K Sharma [9] proposed a system that used a combination of clicked URLs and keywords for clustering the similar queries. It then calculates the weights of URLs which are further used for calculating new ranks. An algorithm to recommend related Web pages corresponding to user's query had been suggested. Similar approach had been used by [10] [11] [12].

Yinglian Xie and David O'Hallaron in [13], proposed a system based on caching technique. They found that the popular queries with high repetition frequencies were shared among different users. Thereby, placing the Web pages in server/ proxy cache. Further, they brought forward the idea of caching at the user side to fulfil the demands of queries that are referred within the short time span.

The study of [14], presented a system by incorporating implicit user behaviour to improve the ranking of top results in real Web search setting. To improve the effectiveness of retrieval results, authors combined the implicit feedback with the ranking algorithms on large scale operational environment. To do this, they collect a random sample of 3000 queries from search engine query log file. They interact with the system over a period of 8 weeks and they found 1.2 millions unique queries and 12 million individual interactions with the proposed system. They average the results over three random samples that splits of the overall dataset and each split contained 1500 training, 500 validation, and 1000 test queries. At last they summarize their result in tabular form and showed a little bit significant improvement over methods that do not consider implicit feedback. They also reported an improvement of 31 % in total precision of results.

In [15], authors proposed a technique that could discover cluster of similar queries and similar URLs. For this purpose they operated bipartite graph where vertices are used for queries and URLs. Edges were used to join the vertices of this graph. They executed agglomerative clustering to the graph's vertices to determine the related queries and URLs. They designed two algorithms for this purpose and implemented it on Sun UltraSPARC by using 266MHz processor with 1.5 GB of physical memory and took ten hours to perform the operation. At the end they declared that the content aware cluster is able to judge the importance of the contents of all the Web pages.

3. Problem Formulation

There are certain challenges in the above stated techniques that are needed to be overcome for creating a more efficient search mechanism that would provide higher relevancy and these challenges are as follows:

- Very few researches are present related to the optimization of the search results[9].
- Number of different clicked URLs may be small. Since user's feedback is not considered, many noisy search results that are not clicked by any users may be analysed as well [16].
- Natural Language queries are inherently unclear. For example, think of a user submitting the query "canon book". Due to unclarity in the query terms, the results obtained could be related either to religious or to photography[17].
- According to a study ([17]) of 2 months of a query log file data, 32.6% queries were limited to two words and 77.2% queries are limited to three words only. These short queries are often puzzling that provides insignificant knowledge to search engine which may result in few relevant and a bundle of irrelevant information returned to the end user resulting in the problem of information overkill.
- Caching technique, stated in [13] would satisfy only a particular set of clients, ignoring the entire user range.
- Moreover, the use of bipartite approach can judge the importance of the Web pages but it also increases the time complexity of the algorithm [15]. Further this technique was not able to explore prosperous source of knowledge and latent URLs from search engine log files.

So the problems mentioned above are resolved by the proposed approach using clustering and ranking the similar queries.

4. Proposed System

The proposed system of clustering and ranking the Web pages is shown in Figure 1. User enters a query to the search engine interface; search engine provides a list of pre-processed Web pages from its database. Next, the user's queries and navigational behavior are stored in the query log for future references. Data from the query log is transferred to the similarity calculator which calculates the similarity between each query with each other possible query avilable in query log file.

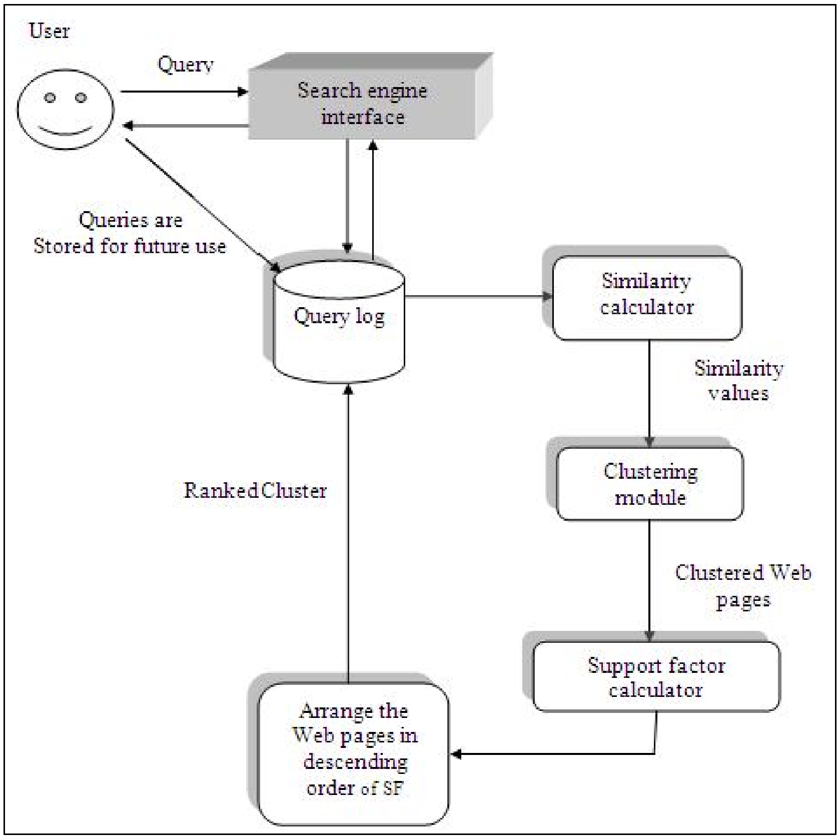


Figure 1: Proposed Architecture for ranking and clustering the search results

Then, based on these similarity values, clustering module clusters the queries into their respective groups. After clusters formation, support factor is calculated which gives the support value for each Web page in the cluster. At last clusters are arranged in descending order of their support values which would be used as a reference in future.
The whole process of clustering the query is organized into following modules:

4.1. Similarity Calculator

This module is used for calculating the similarities between two queries collected from the Web query log. In the proposed system, similarity is defined as the ratio of total number of unique pages between the two queries referred by the users to the total number of common references between the queries [12].

$$\boxed{Similarity\ Calculator = \frac{No.\,of\ \ unique\ pages}{No.\,of\ \ common\ references}}$$

Lower values for similarity means the queries are almost similar and if the similarity value is 0, it means that the queries could be satisfied with the same set of Web pages. On the other hand, if there are not many common references, it would result in higher similarity value. And lastly, if the queries do not have even a single common reference it would be considered as not applicable (NA) for similarity calculation. Further those query that are found NA cannot be grouped in one cluster with other query.

4.2. Clustering Module

Clustering module clusters the queries according to a threshold value decided for the similarity. Queries that have a similarity value less than the threshold value are clustered in one group whereas queries that have a value higher than that of the threshold are grouped in another cluster. Further, there are some queries that do not have any common references between them. As those queries are not compatible with any other query, those are clustered accordingly.

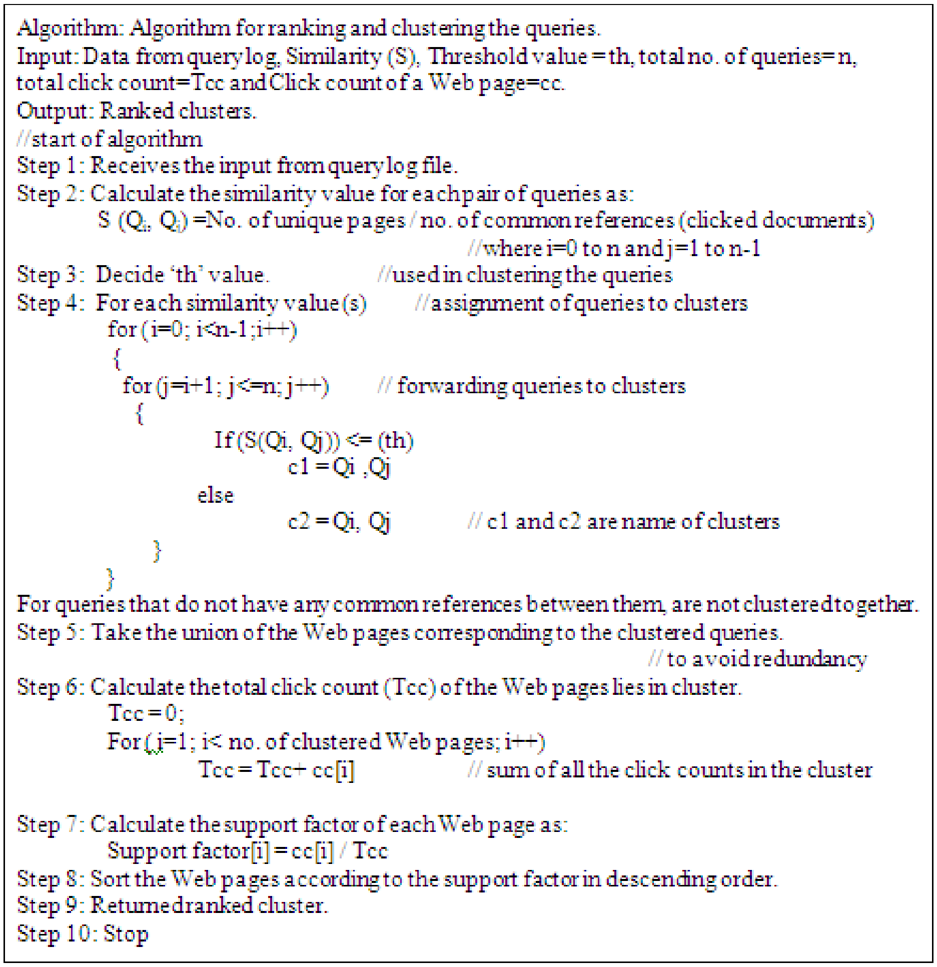
4.3. Support Factor (SF) calculator

Support is the measure of how relevant is the Web page in the cluster. This module is responsible for the union of all the Web pages corresponding to the clustered queries. It measures the support of a Web document as the percent of the clicks of those particular Web pages to the total no of click count of the clustered Web pages [13]. It is estimated from the query log as well.

$$Support\ Factor = \frac{click\ count\ of\ the\ Web\ page}{total\ click\ count\ of\ the\ cluster} * 100$$

4.4. Rank Calculator

Finally this module is used for calculating the new ranks of the Web documents that are clustered before. Ranking is done on the basis of support factor. Web document that has highest support factor is ranked first and so on. The complete process of ranking and clustering is shown in Figure 2.



```
Algorithm: Algorithm for ranking and clustering the queries.
Input: Data from query log, Similarity (S), Threshold value = th, total no. of queries= n,
total click count=Tcc and Click count of a Web page=cc.
Output: Ranked clusters.
//start of algorithm
Step 1: Receives the input from query log file.
Step 2: Calculate the similarity value for each pair of queries as:
          S (Qi,, Qj) =No. of unique pages / no. of common references (clicked documents)
                                                    //where i=0 to n and j=1 to n-1
Step 3:  Decide 'th' value.              //used in clustering the queries
Step 4:  For each similarity value (s)       //assignment of queries to clusters
            for (i=0; i<n-1;i++)
            {
             for (j=i+1; j<=n; j++)       // forwarding queries to clusters
              {
                        If (S(Qi, Qj)) <= (th)
                              c1 = Qi ,Qj
                      else
                              c2 = Qi, Qj           // c1 and c2 are name of clusters
              }
            }
For queries that do not have any common references between them, are not clustered together.
Step 5: Take the union of the Web pages corresponding to the clustered queries.
                                                                    // to avoid redundancy
Step 6: Calculate the total click count (Tcc) of the Web pages lies in cluster.
            Tcc = 0;
            For ( i=1; i< no. of clustered Web pages; i++)
                      Tcc = Tcc+ cc[i]                  // sum of all the click counts in the cluster

Step 7: Calculate the support factor of each Web page as:
            Support factor[i] = cc[i] / Tcc
Step 8: Sort the Web pages according to the support factor in descending order.
Step 9: Returned ranked cluster.
Step 10: Stop
```

Figure 2: Proposed algorithm for ranking and clustering the Web pages

The step 1 of this algorithm, obtains the input from the query log that can be used in step 2 for calculation of the similarity value for each possible pair of queries. A thresh hold value is assumed in Step 3 for clustering the queries. Step 4 checks the similarity value of each query. If this similarity value of selected Web pages is less than the threshold value then these Web pages are clustered in one group. Queries having higher similarity values are clustered within the other group. Step 5, takes the union of all the Web pages to avoid the redundancy of Web pages. Step 6 and step 7 demonstrates the calculation of support value. Finally, step 8 arranges the Web pages in descending order on the basis of the support factor values. Step 9, returned the ranked clusters to the search engine for future reference and step 10 stops the working of algorithm.

5. Experimental Setup

Each search engine maintains its record of given query and the links access by the user in its log file. A typical log file shown in [9] and [18], mainly contains (1) User ID, (2) Queries entered by user, (3) URL clicked by user, (4) click counts and (5) Time of submission of query to the search engine. But, for evaluation of proposed approach authors required only four fields from the above stated fields i.e. user id, entered query, clicked URLs and the number of times the corresponding URL has been clicked. To verify the validity of proposed approach, it is not possible to conduct a detail evaluation on the achieved server data. That's why, only a sample query log with 14 queries with required data is taken into consideration. The following three parameters are tested while performing the experiment i.e. similarity, cluster formation and support factor.

Query no.	User id	Queries	Documents Clicked	Click count
Q1	13509	Samsung Phones	www.samsung.com	5
			www.gsmarena.com	10
			gadgets.ndtv.com	3
Q2	13509	Mobile phones	gadgets.ndtv.com	8
			www.amazon.in	3
Q3	13510	Samsung prices	www.samsung.com	5
			www.gsmarena.com	6
			www.mysmartprice.com	25
Q4	12567	Samsung India	www.samsung.com	5
			gadgets.ndtv.com	5
			www.mysmartprice.com	8
			www.amazon.in	10
Q5	16009	IPods	www.apple.com	10
			www.walmart.com	12
Q6	15688	IPad	www.apple.com	15
			www.bestbuy.com	2
Q7	15689	Samsung Mobile price list	www.samsung.com	14
			www.gsmarena.com	10
			www.mysmartprice.com	23
Q8	13570	Sony	www.sony.co.in	21
Q9	13571	IPad mini	www.walmart.com	7
			www.apple.com	9
			www.bestbuy.com	12
Q10	13571	Samsung Galaxy	www.samsung.com	11
			www.gsmarena.com	10
			www.mysmartprice.com	2
Q11	13571	Nokia phones	www.amazon.in	9
			www.nokia.com	14
			www.microsoft.com	15
Q12	13409	Samsung s	www.gsmarena.com	11
			www.mysmartprice.com	10
			www.samsung.com	11
Q13	13410	Lumia 720	www.nokia.com	12
			www.microsoft.com	5
Q14	12227	IPad air	www.apple.com	5
			www.walmart.com	6
			www.bestbuy.com	7
…	…	….	….	…

Table 1: Sample query log for Practical evaluation

5.1. Similarity

Similarity between two every possible pair of queries is calculated (as per the parameter discussed in section 4) and is shown as below:-

(Q1, Q2)=3/1=3 (Q1,Q3)=2/2=1 (Q1,Q4)=3/2=1.5 (Q1,Q5)=NA (Q1,Q6)=NA (Q1,Q7)=0 (Q1,Q8)=NA (Q1, Q9)=NA (Q1,Q10)=2/2=1 (Q1,Q11)=NA (Q1,Q12)=2/2=1 (Q1,Q13)=NA (Q1,Q14)=NA

Similarly for other queries, the values of similarity was calculated and is avilable with the authors.

5.2. Cluster Generation

Cluster 1 is formed with the help of similarity values calculated above (in section). In cluster 1 every possible pair has a similarity <=1. Hence, the queries are grouped together.

C1-{Q3, Q7, Q1, Q10, Q12}

Apart from all the possible pairs in cluster1, cluster2 contains Q2 and Q4 also has a similarity value 1. But the reason for not grouping them in cluster 1 is the similarity value of Q2 with Q3, Q7 Q10 and Q12. Most of the values are NA (not applicable) so they can’t be clustered together

C2-{Q2, Q4}

In cluster 3, Q5, Q6, Q9 and Q14 are the queries that are not compatible with any other query apart from each other. Similarly, Q5 is not applicable with any other query apart from the queries Q6, Q9 and Q14. Similarly, Q6 is not applicable with any other query apart from Q5, Q9 and Q14.
C3-{Q5, Q6, Q9, Q14}
Similarly Q11 and Q13 are not similar to any other query in the data. But S(Q11, Q13)=0.5 that indicates the queries are quite similar.
C4-{Q11, Q13}
Q8 doesn't have any common references with any other query. So it is clustered in an individual group
C5- {Q8}

5.3. Support factor of the Web pages corresponding to the clustered queries

The support factor of each cluster is calculated according to the parameter discussed in previous section. Table 2 shows the final optimized results of cluster 1. It must be noted that the Support Factor calculation affects the final ranking of Web page inside a cluster and places the most relevant Web page on the top of the cluster.

Links corresponding to cluster 1 (c1)	Total Click-Count	Support (%)
www.samsung.com	46	29.299
www.gsmarena.com	48	30.573
gadgets.ndtv.com	3	1.91
www.mysmartprice.com	60	38.216

Table 2: Implimented Cluster 1

5.4. Proposed ranking for a query in cluster 1

- www.mysmartprice.com
- www.gsmarena.com
- www.samsung.com
- gadgets.ndtv.com

6. Discussion of Results

Furthermore; on the basis of basic concepts used, a critical look on available search engine results optimization techniques provides the differences (shown in Table 3) with the proposed optimization approach. The evaluation of proposed work showed that the ranking and clustering method displayed the results in structured and responsive way as opposed to the results returned by other search optimization techniques. According to the proposed algorithm, more similar query links are grouped into one cluster and unsimilar links in other cluster. Now user can select any desired link of any cluster which will reduce the search space and time also.

Characteristics Parameters	Comparison				
	[19]	[20]	[21]	[9]	Proposed approach
Feature used	Web Log	Data set from different heterogeneous sources	Web log	Web log	Web log
Query sample size	Not known	Not known	6	14	14
Ranking before clustering	Yes	Yes	Yes	Yes	No
Clustering of results	Fuzzy C means	K- Means	Query similarity	Query similarity	Query similarity
Re-ordering of results	No	No	No	No	Yes
Complexity	Yes	Yes	Less than [20]	Less than [21]	Less than [9]
Optimization algorithm used	Probabilistic models	Probability based	Similarity	Similarity	Similarity and support factor
Number of clusters	Not known	Not known in advance	Known	Known	Not Known In advance

Table 3: Comparison between different search engine optimization results

7. Conclusion

This paper has proposed a search engine results optimization technique based on query logs, ranking and clustering the results for implementing effective Web search. The main attraction of this approach was based on determination of similarity of user query words. Based on the obtained similarity value the results were grouped into separate clusters. Moreover, these results were ranked in their respective clusters which provides more important links at the top of displayed results. The proposed algorithm ranks the similar pages in same cluster and unsimilar in other cluster. Now user can select any desired link which reduces the search space of the user. The results obtained from the evaluation of proposed work had been found quite effective in reduction of search space and time.

8. References

1. http://www.internetworldstats.com/stats.htm.
2. Qi He, Daxin Jiang, Zhen Liao, Steven C.H. Hoi, Kuiyu Chang, Ee-Peng Lim and Hang Li, "Web Query Recommendation via Sequential Query Prediction", in proceeding of ICDE '09 Proceedings of the 2009 IEEE International Conference on Data Engineering ISBN: 978-0-7695-3545-6, pp. 1443 – 1454, IEEE Computer Society Washington, DC, USA, 2009.
3. Silverstein Craig, M. Henzinger, H. Marais and M. Moricz, "Analysis of a very large AltaVista query log. Technical Report", in technical notes of Systems Research Center, Compaq Computer Corporation, Department of Computer Science, Stanford University, Stanford ISSN: 0163-5840, 1998.
4. B. J. Jansen, A. Spink, and T. Saracevic, "Real life, real users, and real needs: a study and analysis of user queries on the Web", in Information Processing and Management, pp. 207-227, 2000. DOI: 10.1016/S0306-4573(99)00056-4.
5. J.-R. Wen, J.-Y. Nie, and H.-H. Zhang, "Clustering user queries of a search engine", in proceeding of WWW '01 Proceedings of the 10th international conference on World Wide Web, ISBN: 1-58113-348-0, pp. 162–168, 2001.
6. Brendan O'Connor, "Search Engine Relevance: An Empirical Test", http://blog.doloreslabs.com/2008/04/search-engine-relevance-an-empirical-test/#more-35, accessed April 13, 2008.
7. A. Borchers, Univ. Minnesota, MN Duluth, J. Herlocker, J. Konstan, J. Reidl, "Ganging up on information overload", in Computer, ISSN: 0018-9162, volume: 31, Issue: 4, pp. 106 – 108, 2002.
8. R.Umagandhi and A.V.Senthilkumar, "Time Dependent Approach for Query and
URL Recommendations Using Search Engine Query Logs", in IAENG International Journal of Computer Science,
URL: http://www.iaeng.org/IJCS/issues_v40/issue_3/IJCS_40_3_04.pdf.
9. Neelam Duhan, A.K Sharma."Rank Optimization and Query Recommendation in Search Engine using Web Log Mining Technique", in Journal of computing, ISSN: 2151- 9617, Vol 2, Issue 12, Dec. 2010.
10. Rekha and Sushil Kumar, "Design of Query Suggestion using Rank Updater", in International Journal of Computer Trends and Technology (IJCTT), volume 11, number 5, ISSN: 2231-5381, pp. 220-227, May 2014.
11. Rashmi Rani and Vinod Jain, "Web Search Result using the Rank Improvement", in International Journal of Scientific and Research Publications, ISSN 2250-3153, Volume 3, Issue 5, pp. 1-5, 2013.
12. Kajal Y.VYAS, "Improved Web Search Result Rank Optimization using search engine query Log", Journal of Information, Knowledge and Research in Computer Engineering, ISSN: 0975 – 6760, volume 02, issue – 02, pages 433-436, 2013.
13. Yinglian Xie and David O'Hallaron, "Locality in Search Engine Queries and Its Implications for Caching", in proceeding of INFOCOM 2002, Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, ISSN: 0743-166X, volume: 3, pp. 1238 – 1247, 2002.
14. Susan Dumais, Eric Brill and Eugene Agichtein, "Improving Web Search Ranking by Incorporating User Behaviour Information", in SIGIR '06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, ISBN: 1-59593-369-7, pp. 19-26, 2006.
15. Doug Beeferman and Adam Berger, "Agglomerative clustering of a search engine query log" in Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, ISBN: 1-58113-233-6, pp. 407-416, 2000.
16. Zheng Lu, Hongyuan Zha, Xiaokang Yang, Weiyao Lin and Zhaohui Zheng, "A New Algorithm for Inferring User Search Goals with Feedback Sessions", in Knowledge and Data Engineering, IEEE Transactions volume 25 , Issue 3, pp. 502 – 513, ISSN : 1041-4347.
17. Mirco Speretta and Susan Gauch, "Personalizing Search Based on User Search Histories", in the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, 2005. Proceedings, ISBN: 0-7695-2415-X, pp. 622 – 628, 2005.
18. Kajal Y. Vyas and Kiran R. Amin, "Optimize Rank of Search Engine Query Results
19. Using Log Mining", in Proceeding of the International Conference on Advances in Electronics, Electrical and Computer Science Engineering— EEC 2012, ISBN: 978-981-07-2950-9, pp. 332-336, 2012.
20. K. Poongothai, M. Parimala and Dr. S.Sathiyabama, " Efficient Web Usage Mining with Clustering", in International Journal of Computer Science, ISSN (Online): 1694-0814, volume 8, issue 6, no. 3, 203-209, 2011.
21. Sovers Singh Bisht and Sanjeev Bansal, "Optimization of Web Content Mining with an Improved Clustering Algorithm", in International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 3, Issue 11, pp. 479-483, 2013.
22. Rachna Chaudhary and Nikita Taneja, "A novel approach for Query Recommendation Via query logs", in International Journal of Scientific & Engineering Research, ISSN 2229-5518, Volume 3, Issue 8, pp. 1-6, 2012