Obtaining Personalized and Accurate Query Suggestion by Using Agglomerative Clustering Algorithm and P-QC Method

Saurabh Sharma[1]
M.Tech Research Scholar, CSE Deptt.
MMEC (Maharishi Markandeshwar University Mullana), Ambala

Neeraj Mangla[2]
Assistant Professor, CSE Deptt
MMEC (Maharishi Markandeshwar University Mullana), Ambala

Abstract

Personalized search is the important research area and its main aim is to resolve the ambiguity of the query terms issued by the user. Most of the queries supplied by the user to search engine tends to be short and ambiguous, so they are unable to express the user's actual needs. To overcome this problem, we are creating user profiles to capture the user's personal preference and in this way we can identify the actual goal of the input query. In this paper, agglomerative clustering algorithm is used to find the queries that are close to each other conceptually. We are considering relationship between users, queries and concepts to obtain accurate and more personalized query suggestions for the user. By applying our approach we are getting better precision and recall values when compared with previous query clustering methods.

Keywords- *Search Engine, Personalization, Agglomerative Clustering, Click Through, Web-Snippets.*

1. Introduction

It has become increasingly difficult for web search engines to find information that satisfies users individual needs because the amount of information on the web continuously growing. Million of users interact with search engine daily, they issue queries, follow some of the links in the web snippets, click on ads, spend some time on the pages, reformulate their queries and perform other actions. Personalized search is the better way to improve the search quality by customizing the search results for people with different information needs [5]. Many recent research efforts have focused on this area. Query Clustering is a process used to discover frequently asked questions or most popular topics on a search engine and is crucial for search engines based on question- answering.

Due to large amount of information available on the web, it is very difficult to find related information that satisfies the user needs. The queries given by the user to search engine are usually short and ambiguous. According to a survey average query length on a popular search engine was found to be 2.35 terms [12]. Most of the search engine provides query suggestions to help the user and to formulate more effective queries. So when a user enters the query, the lists of the terms that are semantically related to the given query are provided to the user to help the user in identifying the terms that user wants. But unfortunately these systems provide the same suggestion to the same query without considering the user's interest [2].

2. General Approach

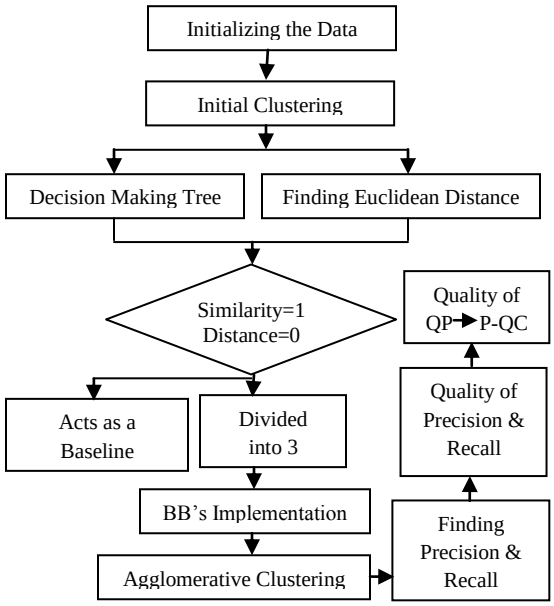

Figure 1. **Flow Diagram of Our Method**

Figure1 shows the flow diagram of our method. First of all we have initialized the data collected from the web snippets returned from the search results. Web-Snippets denotes the title, summary and URL of a web page returned by search engines. After the initialization the initial clustering is done. In initial clustering, Euclidean distance between every interval of query has been calculated and dataset is divided into clusters by the help of Decision making tree. If the query clusters obtained are exactly same then the value of similarity will be 1 and if the Euclidean distance calculated is 0 then it will be the idle case and will act as a baseline for our approach. If similarity is not equal to 1 and Euclidean distance is not equal to 0 then the query clusters will be divided into 3 ranges and labels are assigned to these ranges. In community merging, BB's is implemented on all three ranges and on the basis of similarity and agglomerative clustering algorithm has been applied. After applying agglomerative clustering algorithm Precision, Recall and Cut off values are calculated. Calculate the quality of precision and recall on basis of QU (Quality Unit), QW (Quality Weight) and QC (Quality Cut Off). Calculate the quality of P-QC (Precision-Quality Cut off) on the basis of QP (Quality Precision).

3. Beeferman and Berger's Agglomerative Clustering Algorithm

To evaluate the performance of our approach we are using standard recall-precision measures. BB's Agglomerative Clustering Algorithm is used as the baseline to compare with Personalized Concept Based Clustering Approach [2]. In BB's graph-based clustering, a query-page bipartite graph is first constructed with one set of the nodes corresponding to the set of submitted queries, and the other corresponding to the sets of clicked pages. If a user clicks on a page, a link between the query and the page is created on the bipartite graph. After obtaining the bipartite graph, an agglomerative clustering algorithm is used to discover similar queries and similar pages [8]. During the clustering process, the algorithm iteratively combines the two most similar queries into one query node, then the two most similar pages into one page node, and the process of alternative combination of queries and pages is repeated until a termination condition is satisfied [12]. To compute the similarity between queries or documents on a bipartite graph, the algorithm considers the overlap of their neighboring vertices as defined in the following equation:

$$\text{Sim(x,y)} = \begin{cases} \dfrac{|M(x) \cap M(y)|}{|M(x) \cup M(y)|}, if\ |M(x) \cup M(y)| > 0 \\ 0 \qquad\qquad , \text{otherwise,} \end{cases}$$

where M(x) is the set of neighboring vertices of x, and M(y) is the set of neighboring vertices of y. Intuitively, the similarity function formalizes the idea that x and y are similar if their respective neighboring vertices largely overlap and vice versa.

Now by using the noise-tolerant similarity function, the similarity between two vertices always lies between [0, 1]. The similarity for two vertices is zero, if they share no common neighbor, and the similarity between two vertices is 1, if they have exactly the same neighbor vertices.

4. Concept Derivation

Before explaining the Personalized Concept-Based Query Clustering, we are describing the concept derivation method. This method contains the three steps. In first step, concepts are extracted using the web-snippets returned from search engine. Second, relationship between the concepts can be obtained by mining concept relation. Finally by using derived concepts, user's click through's and concept relations, we can create user concept preference profile [7].

4.1. Concept Abstraction Using Web-Snippets

After a query is submitted to a search engine, a list of web-snippets is returned to the user. This Concept Derivation method is inspired by well known problem of finding the frequent item sets in Data Mining [15]. We assume that if a keyword/phrase exists frequently in web-snippets of a particular query, it represent an important concepts related to the query because it co-exists in close proximity with the query in the top documents. We are using support formula for measuring the interestingness of a particular keyword or phrase (l_i) with respect to returned web-snippets arising from a query q.

$$\text{support}(l_i) = \text{sf}(l_i)/n.|l_i|$$

where n is total no. of web-snippets returned, $\text{sf}(l_i)$ is the snippet frequency of keyword/ phrase l_i (i.e. number of snippets containing l_i) and $|l_i|$ is the number of terms in keyword/ phrase l_i. We treat l_i as a concept for the query q supplied by the user.

First of all we extract all the keyword and phrase from the web-snippets returned by the query. After getting a set of keywords (l_i), we compute the support

for all l_i (support (l_i)). If the support of a keyword/ phrase (l_i) is bigger than the threshold s (support (l_i) > s), we treat l_i as concept of the query q. For example, if we extract the concepts for the query "apple", stop words, such as "the", "of", "we", etc are first removed from the snippets. Maximum length of a concept is limited to seven words. These not only reduce the computational time, but also avoid extracting meaningless concepts.

4.2. Building the Relation between Concepts

We assume that two concepts from a query q are similar if they coexist frequently in the web-snippets arising from the query q. According to the assumption, we apply the following well known [2] signal to noise formula from data mining to establish the similarity between terms X_1 and X_2. This similarity value lies between [0, 1].

$$\text{Sim}(X1,X2)=\frac{n.df(X1\cup X2)}{df(X1)df(X2)}\Big/\log n$$

where n is the total number of documents in the corpus, df(X_1 U X_2) is the joint document frequency of X_1 and X_2, and df(X) is the document frequency of the term X. In the search engine two concepts C_i and C_j could coexist in a web-snippet in the following situation:

First, C_i and C_j coexist in title. Second, C_i and C_j coexist in the summary. Third, C_i exists in the title, while C_j exists in the summary (or vice versa). Therefore by modifying the signal to noise ratio formula for the three different cases:

$$\text{Sim}_R(C_i, C_j)=\text{Sim}_{R,title}(C_i,C_j)+\text{Sim}_{R,summary}(C_i,C_j)+\text{Sim}_{R,other}(C_i,C_j)$$

where $\text{Sim}_R(C_i, C_j)$ is the similarity between concepts C_i and C_j, which is composed of $\text{Sim}_{R,\,title}(C_i, C_j)$, $\text{Sim}_{R,\,summary}(C_i, C_j)$ and $\text{Sim}_{R,\,other}(C_i, C_j)$ as follows:

$$\text{Sim}_{R,title}(C_i,C_j)=\alpha.\log\left(\frac{n.sf_{title}(C_i\cup C_j)}{sf_{title}(C_i)sf_{title}(C_j)}\right)\Big/\log n$$

$$\text{Sim}_{R,summary}(C_i,C_j)=\alpha.\log\left(\frac{n.sf_{summary}(C_i\cup C_j)}{sf_{summary}(C_i)sf_{summary}(C_j)}\right)\Big/\log n$$

$$\text{Sim}_{R,other}(C_i,C_j)=\left(\alpha.\log\frac{n.sf_{other}(C_i\cup C_j)}{sf_{other}(C_i)sf_{other}(C_j)}\right)\Big/\log n$$

where n is the total number of web-snippets returned, $sf_{title}(C_i \text{ U } C_j)$ is joint snippet frequency of concepts C_i and C_j in document title, $sf_{title}(C)$ is snippet frequency of concept C in document titles, $sf_{summary}(C_iUC_j)$ is joint snippet frequency of C_i and C_j in document summaries. $sf_{summary}(C)$ is snippet frequency of concept C in document summaries, $sf_{other}(C_iUC_j)$ is joint snippet frequency of concept C_i in a document title and C_j in the document's summary (or vice versa) and $sf_{other}(C)$ is the snippet frequency of concept in either document summaries or document titles.

4.3. Constructing User Concept Preference Profile

Before deriving user concept preference profile, concept relationship graph is derived first. This graph is derived without considering the user click throughs. In this way we can obtain the possible concept space with the help of graph that arise from users queries [3]. This concept space will cover more than actual user needs. For example, when we search for query "apple", the concept space derived from web-snippets contains the concepts such as "mac", "ipod", "iphone", "recipe". Now if the user is interested in the concept "ipod" and user clicks on the pages that contain the concept "ipod", then the click through should support the concept "ipod" and its neighboring concepts. But the weight of the unrelated concepts and their neighborour's should remain zero [4]. So we propose the following formula for capturing users interestingness w_{ti} on extracted concepts t_i when we click the web-snippet s_j, denoted by click(s_j) can be represented as follows:

$$\text{click}(s_j)\Rightarrow\forall t_i\in s_j.w_{t_i}=w_{t_i}+1$$

$$\text{click}(s_j)\Rightarrow\forall t_i\in s_j, w_{t_j}=w_{t_j}+\text{sim}_R(t_i,t_j)\text{ if }\text{sim}_R(t_i,t_j)>0,$$

Here s_j denotes the clicked web-snippet, w_{ti} shows the interestingness weight of the concept t_i and the t_j is the neighborhood concept of t_i .According to the above formula, when the user clicks on web-snippet s_j then the weight of the concepts t_i that appears in s_j is incremented by 1. It shows the user's interest on the concepts that are found in clicked page s_j.

5. Proposed Algorithm

Using the concepts extracted from web-snippets, we are obtaining personalized and accurate query suggestions by using Agglomerative clustering algorithm and P-QC Method. By using this method ambiguous queries can be classified into different query clusters. Concept-based user profiles are employed in the clustering process to achieve personalization effect. Our technique is composed of following steps: 1) Initialization of data. 2) Initial

clustering with the help of decision making tree and Euclidean distance. 3) Community Merging.

First step of our method is initialization of data. After initialization data set is converted into query clusters with the help of decision making tree and calculates the distance between every interval of query. If query cluster obtained are similar then Similarity=1. If similarity=1 and Euclidean Distance=0, then it will be idle case or acts as a baseline. If similarity≠1 and Euclidean Distance≠0, then query clusters are divided into three ranges. After initial clustering BB's is implemented on the basis of similarity and agglomerative clustering algorithm is applied. Then precision, recall, cut off values, quality of precision and recall on basis of QU (Quality Unit), QW (Quality Weight) and QC (Quality Cut Off) and quality of P-QC (Precision-Quality Cut off) on the basis of QP (Quality Precision) has been calculated.

1. Initialization of data.
2. Initial Clustering
    - Obtain the query cluster from dataset with the help of decision making tree.
    - Calculation of Euclidean distance between every interval of query.
    - If query cluster obtained are similar then Similarity=1.
    - If similarity=1 and Euclidean Distance=0, then it will be idle case or acts as a baseline.
    - If similarity≠1 and Euclidean Distance≠0, then query clusters are divided into three ranges (Minimum, Maximum and Average).
3. Community Merging
    - Implementation of BB's on the basis of similarity.
    - Merging the similar query clusters in BB's.
    - Train the features of all queries.
    - Apply the Agglomerative Clustering Algorithm.
    - Finding Precision, Recall and cut off values.
    - Calculation of quality of precision and recall on basis of QU (Quality Unit), QW (Quality Weight) and QC (Quality Cut Off).
    - Calculation of quality of P-QC (Precision-Quality Cut off) on the basis of QP (Quality Precision).

6. Experimental Results

In this section, we are evaluating the performance of the proposed clustering method for obtaining related queries using user clickthroughs. In section 6.1 we describe the method for collecting the required clickthrough data. In Section 6.2, we compare the performance of QU, QW, and QC methods.

Table 1. **Statistics of the Clickthrough Data Collected in the Experiment**

Statistics	
Number of users	**50**
Number of queries assigned to each user	**5**
Number of test queries	**250**
Number of unique queries	**250**
Maximum number of retrieved URL's for a query	**140**
Maximum number of extracted concepts for a query	**279**
Maximum number of extracted words for a query	**1203**
Number of URL's retrieved	**15390**
Number of unique URL's retrieved	**1006**
Number of concepts retrieved	**14490**
Number of unique concepts retrieved	**7098**
Number of words retrieved	**179843**
Number of unique words retrieved	**24567**

6.1. Statistics of Clickthrough Data

We evaluated the performance of the proposed clustering method for obtaining related queries using user clickthroughs. We have considered total 50 numbers of users. The number of queries assigned to each user is 5. So number of test queries are 250. Table1 shows the statistics of clickthrough data.

When a query is submitted to the middleware, the top 100 search results from Google are retrieved, and the web-snippets of the search results are displayed to the users. Since most users would examine only the top 10 results, our concept extraction method, digging deep into the first 100 results, will discover concepts related to the query that would otherwise be missed by the users. The users were asked to click on the web-snippets of the returned results that are relevant to the queries. The clickthrough data collected are used to measure the performance of the concept-based clustering method. Test queries are separated into predefined clusters.

6.2. Comparison of QU, QW and QC Method

We compared the performance of our proposed algorithm using QU, QW, and QC methods. QU method is the original input of our baseline algorithm, which serves as a baseline for comparison. QW method uses query-word bipartite graph, which is similar to the query-concept bipartite graph in that they are both constructed using proposed Algorithm. The difference is that the former contains all words (excluding stop words) from the web snippets and the latter contains the extracted concepts. QW and QC methods are necessary, since they allow us to study the benefits of concept extraction. The three methods are also employed to cluster the collected data. The results are compared to our predefined clusters for precision and recall. Given a query q and its corresponding query cluster {q1, q2, q3.....} generated by a clustering algorithm, the precision and recall are computed using the following formulas:

$$precision(q) = \frac{|Q\_relevant \cap Q\_retrieved|}{|Q\_retrieved|}$$

$$recall(q) = \frac{|Q\_relevant \cap Q\_retrieved|}{|Q\_relevant|}$$

where $Q\_relevant$ is the set of queries that exist in the predefined cluster for q, and $Q\_retrieved$ is set of the related queries {q1, q2, q3.....} generated by the algorithm. The performance of the three methods is compared using precision-recall figures and best $F\text{-}measure$ values.

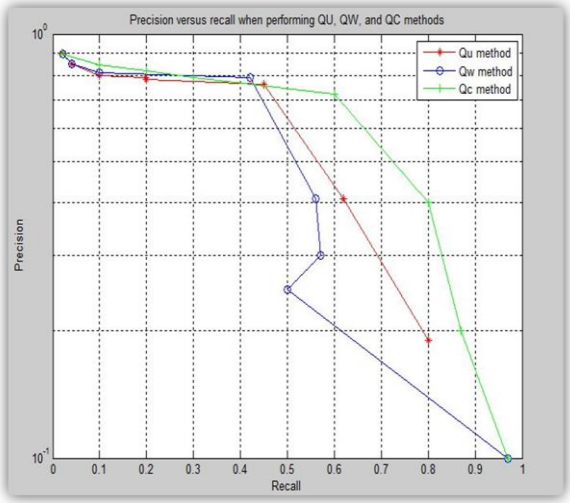

Figure 2. **Precision versus Recall when Performing QU, QW and QC Method**

Figure2 shows Precision versus recall when performing QU, QW, and QC methods. We observe that QC method yields better recall rate than QU method, while preserving high precision rates. This can be attributed to the fact that the average number of overlapping concepts between the queries is much higher than the URL overlap rate.

As a result, related queries that cannot be discovered by URL overlap can be brought together by our QC method, thus improving the recall rate. Table2 shows Statistics of the Precision and Recall values Collected in QU, QW, and QC Method.

Table 2. **Statistics of the Precision and Recall Values Collected in QU, QW, and QC Method**

QU Method		QW Method		QC Method	
Precision	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
0.90	0.02	0.899	0.021	0.898	0.02
0.85	0.04	0.849	0.041	0.848	0.10
0.80	0.10	0.810	0.100	0.820	0.20
0.79	0.19	0.789	0.420	0.788	0.30
0.78	0.20	0.410	0.560	0.720	0.60
0.76	0.45	0.300	0.570	0.400	0.80
0.41	0.62	0.250	0.500	0.200	0.87
0.19	0.80	0.100	0.970	0.100	0.97

Table 3. **Best F-Measure Values of QU, QW, and QC Method**

Best F-Measure Values			
Method	**Precision**	**Recall**	**F-measure**
QU Method	0.900	0.80	0.847
QW Method	0.890	0.97	0.928
QC Method	0.898	0.97	0.932

Table3 shows the best $F\text{-}measure$ values for the QU, QW, and QC methods. From the results, we can conclude that query clusters obtained using QC method are much more accurate compared to those obtained from QU and QW methods.

Figure3 shows the change of precision, for the three clustering methods. When the cutoff similarity score is around 0.3, the precision obtained using QU method is very close to that of QC method, which is much better than the precision obtained using QW method. We observe that the three methods are able to achieve their optimal precision/recall at different cut off similarity scores.

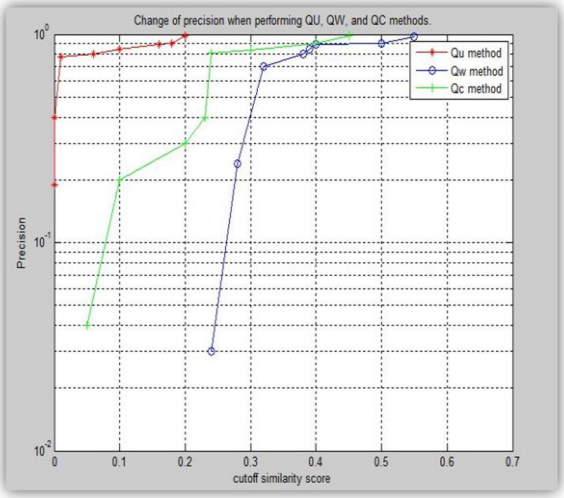

Figure 3. **Change of Precision when Performing QU, QW, and QC Method**

To obtain and compare the best F-measures (i.e., evenly weighted harmonic means of precisions and recalls) for the three different methods, The F-measure is defined by the following formula:

$$F = 2.\frac{precision \times recall}{precision + recall}$$

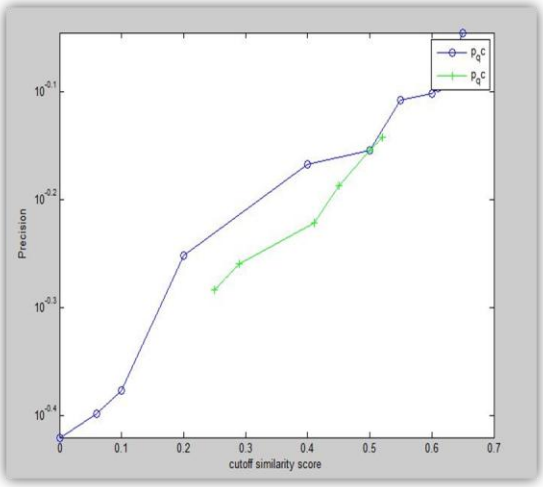

Figure 4. **Change of Precision when Performing P-QC Method**

Figure4 and Figure5 show the change of precision and recall when performing P-QC method. In , Figure5 we observe that the precisions generated by community merging are slightly lower than those generated by initial clustering because some unrelated queries can be wrongly merged in community merging.

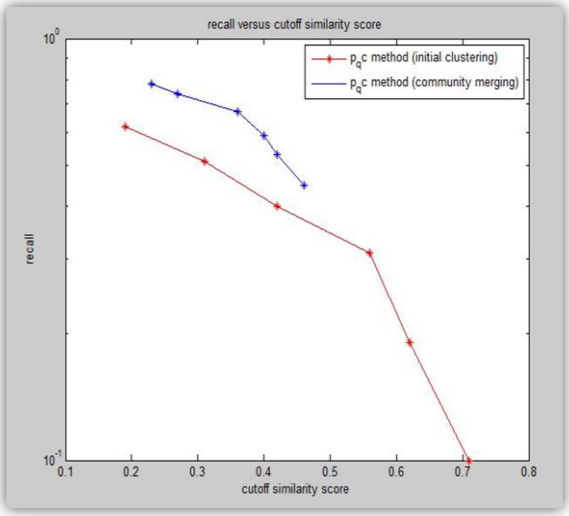

Figure 5. **Change of Recall when Performing P-QC Method**

In Figure5, we observe that the recalls generated by community merging are much higher than those generated by initial clustering because community merging can successfully merge conceptually related clusters together.

Table 4. **Cut-Off Values for Initial Clustering in P-QC Method**

Initial Clustering			
Cut-Off	**Precision**	**Recall**	**F-measure**
0.71	**0.90**	**0.62**	**0.734**
0.62	0.80	0.51	0.622
0.62	0.79	0.51	0.619
0.56	0.78	0.40	0.528
0.56	0.70	0.40	0.509
0.42	0.68	0.31	0.425
0.42	0.56	0.31	0.399

We can easily see from Figure5 and Figure6 that only a small fraction of precision is used to trade for a much better recall in community merging. Table4 shows the cut-off values for initial clustering in P-QC Method. Table5 shows the cut-off values for community merging in P-QC Method.

Table 5. **Cut-Off Values for Community Merging in P-QC Method**

Community Merging			
Cut-Off	**Precision**	**Recall**	**F-measure**
0.46	**0.72**	**0.78**	**0.748**
0.42	0.70	0.74	0.719
0.42	0.70	0.74	0.719
0.40	0.65	0.67	0.659
0.40	0.65	0.67	0.659
0.36	0.60	0.59	0.595
0.36	0.60	0.59	0.595

7. Conclusions

As search queries are ambiguous, there are several effective methods for search engines to provide query suggestions on semantically related queries in order to help users formulate more effective queries to meet their diversified needs. In this dissertation, we have proposed a new personalized concept-based clustering technique that is able to obtain personalized query suggestions for individual users based on their conceptual profiles. The technique makes use of click through data and the concept relationship graph mined from web-snippets, both of which can be captured at the back end and as such do not add extra burden to users. An adapted agglomerative clustering algorithm is employed for finding queries that are conceptually close to one another. Our experimental results confirm that our approach can successfully generate personalized query suggestions according to individual user conceptual needs. Moreover, it improves prediction accuracy and computational cost compared to BB's algorithm, which is the state-of-the-art technique of query clustering using clickthroughs for the similar objective.

There are several directions for extending the work in the future. Click through data and concept relationship graphs can be directly integrated into the ranking algorithms of a search engine so that it can rank results adapted to individual users' interests.

8. Acknowledgment

I sincerely express my deep gratitude to Assistant Prof **Neeraj Mangla (Computer Science Engineering Department, Mahrishi Markendshwar Engineering College, Mullana (Haryana), INDIA**), for their constant encouragement, motivation and supervision throughout my Thesis work. His endless support helped me in channeling my efforts and ideas in proper direction.

9. References

[1] Kenneth Wai-Ting Leung, Hing Yuet Fung, Dik Lun Lee, "Constructing Concept Relation Network and its Application to Personalized Web Search" in IEEE Transactions, FEB 2011, pp. 413-424.

[2] Kenneth Wai-Ting Leung and Dik Lun Lee, "Deriving Concept-Based User Profiles from Search Engine Logs" in IEEE Transactions, VOL. 22, and NO. 7, JULY 2010, pp. 969-982.

[3] Kenneth Wai-Ting Leung, Dik Lun Lee, Wang-Chien Lee, "Personalized Web Search with Location Preferences" in IEEE Transactions, JULY 2010, pp. 701-712.

[4] Z. Dou, R. Song, and J.R. Wen, "A Large-Scale Evaluation and Analysis of Personalized Search Strategies," Proc. 16th Int'l World Wide Web Conf. (WWW), 2007, pp. 581-590.

[5] Y. Xu, B. Zhang, Z. Chen, and K. Wang, "Privacy-Enhancing Personalized Web Search," Proc. 16th Int'l World Wide Web Conf. (WWW), 2007, pp. 591-600.

[6] T. Joachims and F. Radlinski, "Search Engines That Learn from Implicit Feedback," Computer, vol. 40, no. 8, 2007, pp. 34-40.

[7] E. Agichtein, E. Brill, and S. Dumais, "Learning User Interaction Models for Predicting Web Search Result Preferences," Proc. 29th Ann. Int'l ACM SIGIR Conf. (SIGIR), 2006, pp. 3-10.

[8] E. Agichtein, E. Brill, S. Dumais, and R. Rango, "Improving Web Search Ranking by Incorporating User Behaviour Information," Proc. 29th Ann. Int'l ACM SIGIR Conf. (SIGIR), 2006, pp. 19-26.

[9] Z. Zhang and O. Nasraoui, "Mining Search Engine Query Logs for Query Recommendation," Proc. 15th Int'l World Wide Web Conf. (WWW), 2006, pp. 1039-1040.

[10] B. Koester, "Conceptual Knowledge Retrieval with FooCA, "ImprovingWeb Search Engine Results with Contexts and Concept Hierarchies," Proc. Sixth IEEE Int'l Conf. Data Mining (ICDM), 2006, pp. 176-190.

[11] B. Smyth et al., "Exploiting Query Repetition and Regularity in an Adaptive Community-Based Web

Search Engine," User Modeling and User-Adapted Interaction, vol. 14, no. 5, 2005, pp. 383-423.
[12] M. Speretta and S. Gauch, "Personalized Search Based on User Search Histories," Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence (WI), 2005, pp. 622-628.
[13] R.A. Baeza-Yates, C.A. Hurtado, and M. Mendoza, "Query Recommendation Using Query Logs in Search Engines," Proc. EDBT Workshop, vol. 3268, 2004, pp. 588-596.
[14] S.M. Beitzel, E.C. Jensen, A. Chowdhury, D. Grossman, and O. Frieder, "Hourly Analysis of a Very Large Topically Categorized Web Query Log," Proc. 27th Ann. Int'l ACM SIGIR Conf. (SIGIR), 2004, pp. 321-328.
[15] V.W. Chan, K.W. Leung, and D.L. Lee, "Clustering Search Engine Query Log Containing Noisy Clickthroughs," Proc. Symp. Applications and the Internet (SAINT), 2004, pp. 305-311.

Saurabh Sharma was born at H.No. 70/11, Chougan, SunderNagar, Distt. Mandi (Himachal Pradesh) in India, on May 21, 1987. He graduated from Dr. JJMCOE Jaysingpur, Distt. Kolhapur (Maharashtra), India, and completed his Masters of Technology (batch 2009-11), CSE Deptt, from MMU, Mullana, Ambala (Haryana), INDIA. He has successfully completed his B.E. Final Year Project-Multi-layer Database Access Control Architecture (MDACA).

Neeraj Mangla is graduated (B.Tech) from JMIT, Radaur, Yamunanagar, post graduate (M.Tech) from the MMEC Mullana, Ambala and persuing P.hd from MMEC Mullana, Ambala, India. His employment experience is about 8 years. He has guided 6 M.tech Thesis and 4 are under guidance. He has published 15 papers in international journals, International and National conferences. He is working as an Assistant Professor in Computer Science Engineering Department, MMEC MM University Mullana, Ambala, Haryana India.