Search Query Recommendations using Hybrid User Profile with Query Logs

R.Umagandhi
Associate Professor and Head
Department of Computer Technology
Kongunadu Arts and Science College
Coimbatore, India

A.V.Senthil Kumar, Ph.D
Director,
Mater of Computer Applications
Hindustan College of Arts and Science
Coimbatore, India

ABSTRACT
The exhaustive information available in the World Wide Web indeed, unfolds the challenge of exploring the apposite, precise and relevant data in every search result. Apparently, in such instances of web-searching, Query Recommendations is the ultimate application in information retrieval. The Query Recommendation technique provides alternative queries to the user to frame a meaningful and relevant query in the future and rapidly satisfies their information needs. Similar query keywords are juxtaposed with the concept based hybrid user profile from the user log, query log and click-thru snippets to re-conduct the recommendation generation phase. The concept based hybrid user profile is used for recommending and re-ranking the queries. The given technique is very efficient and scalable; it is particularly effective in generating suggestions for rare queries and newly occurring queries. Experimental results based on log files and click-through data prove that the proposed algorithm performs well with better outcomes. The proposed strategies are experimentally evaluated using real time search process.

Keywords
Recommended Queries, Concepts, User Log, Query Log, Snippets.

1. INTRODUCTION
Generally the search engine users are naive and frame their query keywords in front of the search engines only. They are having less background knowledge about their information needs. Also the input queries are short and ambiguous [1]. The shorter length queries do not produce the accurate results, so the query recommendations are an essential technique to provide suggestions to the user to frame their queries in the future. Query recommendation helps to describe the user's information needs more clearly so that search engines can return relevant and appropriate answers. Modern researches prove that the analysis of query log and the use of users' behaviour information help to improve query recommendation performance [2] [3]. The real search intent of the user is analysed and retrieved not only by using the query logs [4] but also by the clicked concepts from the web snippets ("Web-snippet" denotes the title, abstract, and URL of a Web page returned by the search engines) and the user's preferences in the search result. NPD 2000 [5] reports that an independent survey of 40000 web users found that after a failed search, 76% of users try to rephrasing their query in the same search engine. This is a situation that the possibilities are more that the user can select the recommended queries.

During the search process, information about the user is collected in two different ways [6] [7]. Implicit profile is automatically created using the search behaviour of the user from the query log. Explicit profile is created by the user by providing the feedbacks explicitly. The main drawback of the query recommendation process using the implicit [8] [9] user's feedback is it is not possible to accurately find the user's real search intent. Google Personalized Search builds a user profile by means of implicit feedback where the system adapts the results according to the search history of the user. The user hesitates to provide all the information explicitly and frequently.

The main drawbacks while the user uses the search engine for the information retrieval are

- Single user uses different computers for searching process, different user IDs are created for a single user. In this case, the query log processor creates a distinct entry with multiple IDs. Here the user's real intent is not analyzed properly.
- If one system is used for searching process by different users then all the users have single user ID. In this case, multiple user behaviours are treated for single user.
- When a system connected with the internet, every time different IP is created and stored in the query log file.
- The identification of IP or user history are machine oriented either the user changes the system or the user is new, the result may go wrong.

The traditional query recommendation techniques may go wrong for the above cases. To avoid the above anomalies we propose this system and it is based on the hybrid user profile. The major contributions of the proposed method are summarized as follows:

- User's preferences in the search engine for the query is analysed from the user log file. Here, the information about the user is recorded implicitly.
- The past queries and search behaviour is analysed from the user's implicit feedback and the user's concept intent is retrieved from the query log.
- The concepts are retrieved from the clicked web snippets and pre-processed. The relationships between the concepts are represented as a similarity matrix.
- The queries are recommended using User log, Query log and Concept log files.
- The click count for every URL against the query supplied by the user is calculated. The favourite query of the user is identified. The recommended queries are re-ranked using the favourite query and URL click count.

Some of the basic terms used in the proposed method are defined below.

Relevant: A document is relevant to the query, when the document contains the query as one of its concept or tag words. That is Similarity (*Query*, {*Concept*, *Tag words*}) $\neq \varnothing$

Consistency [10]: A document is consistent with a query, if it has been selected number of times during the query session.

Recommended Query: Set of queries is recommended to frame the better queries in the future instead of the initial query.

Rank of Recommended Query: The rank of a recommended query r is the position of the query in the recommended list. When Q_i comes before Q_j if Rank (Q_i) > Rank (Q_j)

The rest of the paper is organised as follows: Section 2 reviews the related work; Section 3 defines the Recommendation Technique based on the concept based user profile; Section 4 discusses the experiments and results. Finally Section 5 concludes the paper.

2. RELATED WORK

Queries are keywords in the searching process used to retrieve the necessary information from the millions of web sites. Some times, the issued query may be shorter, bamboozling or ambiguous. In this situation, the shorter keyword does not reflect the real intent of the user exactly and it has paved the way for search engine to retrieve the irrelevant and redundant web snippets [11].

The search engine's query recommendations is personalized using the information about the users in terms of the concept based user profile. Information can be collected from the users in two ways [6] [7]: either explicitly, for instance, asking feedback such as preferences or ratings; or implicitly, for example observing user behaviours such as the time spent reading an on-line document, number of times an URL is clicked and etc. Explicit construction of user profiles has some drawbacks [12]. The users may provide inconsistent or incorrect information, the profile is static whereas the user's interests may change over time, and the construction of the profile places a burden on the user that the user may not wish to provide all the information recursively. On the other hand, implicitly created user profiles do not place any burden to the user. Thus, many research [13] [14] created the user profiles implicitly and provide the recommendations.

User profiles can be used to represent the user's preferences [15] (e.g., Search engines preferred, types of documents) and interests (e.g., Sports, photography). In general, user profiles are under the following categories namely,

- Content-based profiles - The profile is generated from the concepts preferred by the user.
- Collaborator profiles - Grouping users who are having similar interest.
- Rule-based profiles - Rules are created from the answers provided by the users on questions about information usage and filtering behaviour.

Currently, most commercial search engines and lots of research work focus on how to recommend the queries based on users' previous query and click behaviours. The idea is to locate popular queries which are similar with the current query either in content [1] [16] [17] or in click context [18] [19]. This kind of recommendation lacks understanding of users' actual information needs. It does not take current users' search intent into consideration; instead, it uses collaborative recommendation that shares similar interests with other users who propose similar queries. But the proposed method recommends the queries using Content-based profiles and also the Collaborative profiles. The two snippet click models namely global scale snippet click model and a local scale snippet click model and their corresponding recommendation algorithms are described in [20]. Instead of finding the similar keywords from the query log, the real user's information needs are analysed by retrieving the concepts from clicked snippets.

But the proposed user profile recommends the queries and also re-rank the recommended queries based on the intent of the user. The proposed technique generates the concept based user profile from (i) user preferences given explicitly in the log file (ii) clicked snippets shows the user's intent of the present query and (iii) past queries and its click thru from the query log. Hybrid User profile generated in this work is used for many search engine personalization task such as query suggestion at hitting time, query recommendation to frame the future queries and provide the effective search result based on the real intent of the user and also re-rank the recommended queries and search result.

The input of the query recommendation process can be a user profile, query log or an external source like ontology, web pages, etc. The recommendation may be provided before querying, while querying or after querying. Table 1 lists the comparison between the previous techniques and the proposed method.

3. QUERY RECOMMENDATIONS BASED ON CONCEPT BASED HYBRID USER PROFILE

The architecture and the overall process are explained in Section 3.1. The proposed query recommendation technique consists of three steps. First, user information is gathered and stored in the user log which is explained in Section 3.2. Second, the collection of user's implicit feedback from click thru and past queries using the query log is explained in Section 3.3. The important concepts retrieved from the clicked snippets against the given query and finding the concept similarity is explained in Section 3.4. Section 3.5 explained the overall recommendation process and its format. Re-ranking of the recommended queries is given in Section 3.6.

3.1 Architecture for Query Recommendations using Hybrid User Profile

Figure 1 shows the recommendation process of the proposed work. The proposed technique recommends the query in the following manner. The user's information is collected explicitly using the registration form and it is stored in the user log. The registered users either apply the search query to the search engine through this interface or update the general status. The proposed technique pre-processes the query keywords and retrieves the search result from Google by using the given keywords. The proposed technique can be directly integrated into any search engine to provide the query recommendations. The user can also update the status of the user against the query. Here the profile is created for each query of the user. Most existing recommenders [6] [27] generates a single profile for the user and this profile is applied to all the queries given by the user. But the user's preference is not stable and it varies across queries. Finally the set of queries is recommended to the search user along with the resultant web snippets.

Table 1. Query Recommendation Approaches – A Comparison

Research Works	Recommendation Type		Recommendation Time		Recommendation Input Data	
	Content Based	Collaborative	While	After	Log file	User Profile
Stefanidis et al., [21]	✔	✔		✔	✔	
Chatzopoulou et al., [22]		✔		✔	✔	
Khoussainova et al., [23]		✔	✔		✔	
Golfarelli et al., [24]		✔		✔		✔
Khemiri et al., [25]	✔		✔		✔	
R. Umagandhi et al. [26]	✔	✔		✔	✔	
Proposed Method	✔	✔	✔	✔	✔	✔

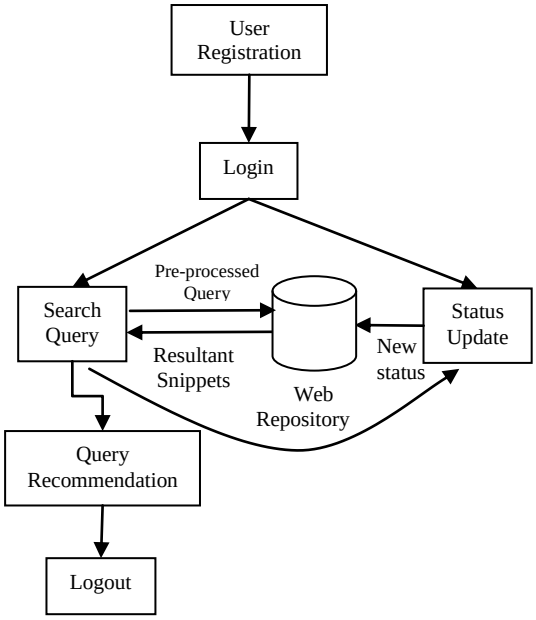

Fig. 1: Proposed Query Recommendation Technique

Figure 2 shows the architecture of the proposed method. The generation of hybrid user profile comprises the following phases,

- Phase 1: Before submitting the initial query to the search engine the user must give their information explicitly through some feedback forms.
- Phase 2: The user's preferences and interests are obtained implicitly by using their search queries and click thru behaviours. The navigational information is stored in the Query Log. The user who are having the similar search intent are analysed from the query log and it is clustered. The queries from the similar users are also given as the recommendation for the initial query of the user. The clustering process is explained in [26].
- Phase 3: The concepts from the clicked snippets are retrieved and the relationship between the concepts is stored in the concept log.

3.2 Generation of User Profile

The information about the user can be collected in two ways. *Explicit user feedback information*: Attributes like login information, shared concepts from search result, query keywords, date and time on which the query is issued and the selection of documents etc are some of the user's information gathered explicitly from the user. But this technique collects the user's information other than the search activity. *Implicit user feedback information*: The user's interaction with the search engine is used to collect and analyse the user's real intent and search behaviour. The positive effect of this method is that the technique does not require any additional effort from the user.

Figure 3 shows the sample information about the user collected explicitly by using the registration form. When the user is registered in the recommender system, it gets the basic details of the user; still the user might give wrong information or infill it. The user may also update the general status or status against each query. The general status is stored in the user log. The updated status is one of the recommended queries. For example the user like the cricketer 'Ms Dhoni' and he update the status regarding him. The next time the user gives the search keyword 'cricket', the search engine will give the result of Cricket and the recommendation list contains 'Ms Dhoni'.

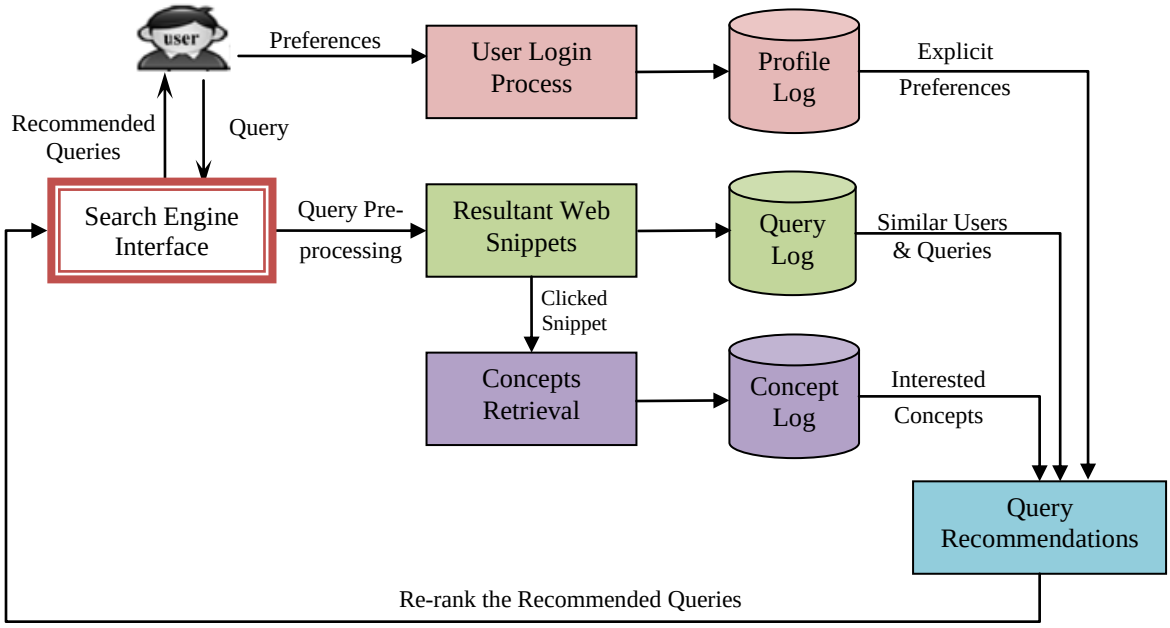

Fig. 2: Architecture of the Query Recommendation Process

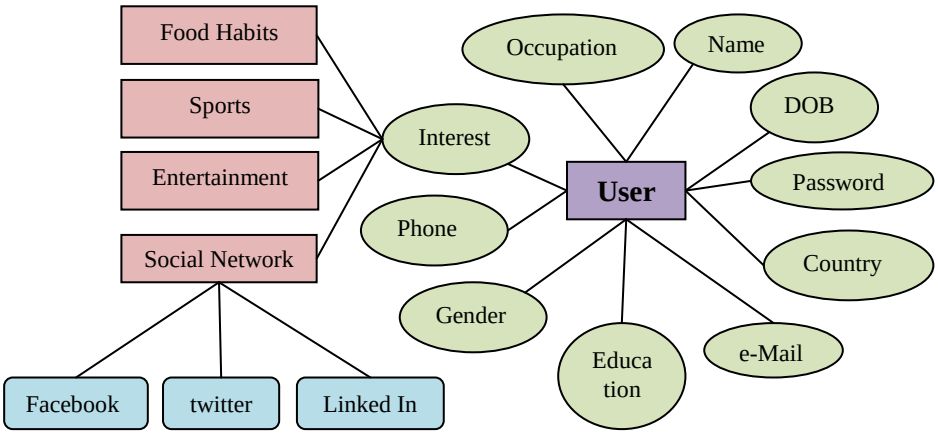

Fig. 3: User Profile Representation

3.3 Processing of Query Log

Search engine leaves the search information to the user for further references in the form of query logs. Query log is an important repository, which records the user's search navigational behaviours. The mining of these logs can improve the performance of the search engines. In order to give the recommendations to frame the future queries, the search histories in the query logs are analysed. The search histories are organized under the attributes:

< UserName Query QueryTime ItemRank ClickURL>

Table 2 shows the attributes and their descriptions used in the data set.

The entries in the query log are analysed. First the users and their sessions are identified and the User's favourite query is generated [26]. The similar users are analyzed and clustered using Agglomerative clustering algorithm [28]. The query keywords from the similar uses and are also given as recommendations.

Table 2. Query Log's Attributes and their Descriptions

Attribute	Description
UserName	Name of the user given in the log in process
Query	The query issued by the user
QueryTime	The date and time on which the query was triggered by the user
ItemRank	If the user clicks on a search result, the rank of the item on which they click is listed.
ClickURL	If the user clicks on a search result, the domain portion of the URL in the web snippets is listed.

3.3 Concepts Extraction

After the query is submitted to the search engine interface, the query keywords are pre processed by using the techniques in [26] and a list of Web snippets [29] is returned to the user. The user scans the retrieved web snippets from the top to the

bottom according to Joachims [30] and then decides which one of the documents is relevant and then clicks it. The important concepts from the clicked documents are retrieved and stored in the concept log. The derived concepts are pre-processed in the way of,

- The concepts are converted into lowercase letters.
- Extra spaces are trimmed.
- All the plurals are converted into singulars. (It is called Lemmatization. Morpha is used for the conversion. It is downloaded from www.informatics.sussex.ac.uk /research)
- Some of the special symbols and words are truncated. (Remove the words like cached, similar and etc. and symbols like @, ., ; and etc.)
- Stop words are removed from the clicked snippets. (List of Google stop words are downloaded from http: //code.google.com/p/andd/downloads/detail?name=stopwords.csv)

If the concept exists frequently on the Web-snippets for a particular query, it is known as an important concept related to that query. The support value is used to find the interestingness of a concept. In the concept extraction, first identifies the support value of the unique concepts of length one. If the support value satisfies the minimum support threshold then the concepts with higher length is generated. The frequently occurred concepts in the clicked web snippets are identified by using support formula [29],

$$Support(C_i) = \frac{sf(C_i)}{n} \quad (1)$$

Where $sf(C_i)$ is snippet frequency, number of clicked web snippets contains the concept C_i, n is the total number of clicked web snippets. If the support of a concept C_i is greater than the threshold *s* then C_i is an important concept related to the query *q*. In our experiments, the concept is an important one, when it occurs in minimum of 50% of the clicked documents. The support value is calculated only for the concepts in the clicked snippets; if it satisfies the threshold *s* then it is treated as important concepts as positive preferences. [29] Set the maximum length of a concept which is limited to seven words because it limits the computational time and also avoids extracting meaningless concepts. The proposed technique is also considers the concepts with a maximum length of seven. Maximum number of concept's combinations to be generated for the query *Q* is

$$Max\_concepts = \sum_{i=1}^{n} 2^{m_i} - 1 \quad (2)$$

Where m_i is number of concepts in the i[th] document and *n* is the number of documents. For example, consider D1, D2 and D3 are clicked documents out of ten snippets which contains,

D1 = {a, b, c, d, e}
D2 = {a, b}
D3 = {a, c}

Maximum number of concept combinations to be generated for the three clicked documents is 31+3+3=37. Number of combinations among the concept is nC_r where *n* is the length of the document that is number of concepts in the document and *r* is the number of words is to be combined. For example, number of concepts to be generated with the length of four in the document D1 is $5C_4$ it is 1. Table 3 shows that the concept patterns and its support value in the documents D1, D2 and D3.

From Table 3 (a), the concepts a, b and c satisfies the threshold value 50% and the support value of d and e are 1. The concepts a, b and c are used to generate the concepts of length 2. The support of the concept '*bc*' is 1, so it is not considered for the next level. Finally, the generated concept patterns and its support value are shown in Table 3 (d). The maximum number of concepts to be generated is 37 with the maximum length of five, but the proposed method generates five concepts with the maximum length of 2.

Next, the relationship between the extracted concepts in the clicked web snippets is identified. Here the similarity measure is used to obtain the relationship between the concepts. The concepts co-exist either in the title, abstract or at the tags. The tags are keywords used to retrieve the web page; it is defined in <meta> tag. The tags are displayed publicly only in few web pages. The format of the <meta> tag is

<meta name = "description" content="a description of your site">

<meta name="keywords" content="relevant keywords about your site">

The combined similarity measure is

$$Sim(C_i, C_j) = \propto \cdot \frac{sf_{title}(C_i \cup C_j)}{sf_{title}(C_i) \cdot sf_{title}(C_j)} + \beta \cdot \frac{sf_{abstrct}(C_i \cup C_j)}{sf_{abstrct}(C_i) \cdot sf_{abstrct}(C_j)} + \gamma \cdot \frac{sf_{tags}(C_i \cup C_j)}{sf_{tags}(C_i) \cdot sf_{tags}(C_j)} + \delta \cdot \frac{sf_{others}(C_i \cup C_j)}{sf_{others}(C_i) \cdot sf_{others}(C_j)} \quad (3)$$

Where $\propto + \beta + \gamma + \delta = 1$ to ensure that the similarity is lies between [0, 1]. $Sim(C_i, C_j)$ is the combined similarity between the concepts C_i and C_j. $sf_{loc}(C_i \cup C_j)$ is the joint snippet frequencies of the concepts C_i and C_j where $sf_{loc}(C)$ is the number of snippets contained the concept *C* and loc={title, abstract, tags, others}. Here 'others' denotes that the different combinations of concept locations. The proposed work considers all the combinations for calculating the similarity. Table 4 lists different combinations of locations where the concepts C_1 and C_2 may appear. For example, in Location number 1, the concept C_1 appears at title whereas the concept C_2 in abstract.

Consider the locations {3, 4, 7, 8, 9 and 10}. If the concepts C_1 and C_2 occur at the location of Title, Abstract or Tags then the joint snippet frequency $C_1 \cup C_2$ is occurred in the maximum of 2 combinations of {(Title, Abstract), (Title, Tags), (Abstract, Tags)}. If any one of the location is empty then the joint snippet frequency $C_1 \cup C_2$ is occurred in the maximum of 1 combination.

The support value of the concept *C* is calculated based on the appearance of *C* in the title, abstract and tags.

$$Support(C_i) = \sum_{loc} \propto_j \cdot \frac{sf_{loc}(C_i)}{n} \quad (4)$$

Table 3 (a). Concepts with 1 word

Concept Pattern	Count & Support
a	3 & 100%
b	2 & 67%
c	2 & 67%
d	1 & 33%
E	1 & 33%

(b) Concepts with 2 words

Concept Pattern	Count & Support
ab	2 & 67%
ac	2 & 67%
Bc	1 & 33%

(c) Concepts with 3 words

Concept Pattern	Count & Support
abc	1 & 33%

(d) Selected Concepts

Concept Pattern	Count & Support
a	3 & 100%
b	2 & 67%
c	2 & 67%
ab	2 & 67%
ac	2 & 67%

Table 4. Locations of the concepts C_1 and C_2

Location No.	Title	Abstract	Tags	Location No.	Title	Abstract	Tags
1	C_1	C_2	-	7	C_2	C_1	C_1
2	C_1	-	C_2	8	C_2	C_2	C_1
3	C_1	C_2	C_2	9	C_1	C_2	C_1
4	C_1	C_1	C_2	10	C_2	C_1	C_2
5	C_2	C_1	-	11	-	C_1	C_2
6	C_2	-	C_1	12	-	C_2	C_1

< Item Name *Favourite Query: Number of times the query triggered by the user*>

<Item Name *Number of times Clicked and its history*>

<Item Name *Number of Clicked web snippets containing the Item*>

<Item Name *list of queries from similar users*>

Fig. 4: Structure of Recommended Queries

Where loc = {Title, Abstract, Tags}. $\propto_j$ is used to normalize the support values in between [0, 1] where $1 \leq j \leq 3$. c_i is the concept and $1 \leq i \leq m$ where m is the number of concepts retrieve from the clicked snippets.

3.5 Query Recommendations using User Profile

Algorithm QRecommender
Input: Query Log, Concept Log and Updated User Log
Output: Set of k recommendations
begin
Step 1: Registration and Log in Process of a user and the user's information is in user log.
Step 2: Pre-process the query keywords. Extract and store the web snippets for the pre-processed query.
Step 3: Analyse the query log about the user's search and navigational behaviour. Identify the favourite query of the user.
Step 4: Extract the concepts from clicked snippets and stored in Concept Log against the query.
Step 5: Analyze the Concept log and recommend the concepts as queries.
Step 6: The recommendations are ranked and self explanatory.
end

Algorithm QRecommender generates top k recommendations for the query Q. The steps given in the algorithm shows that the overall process of the proposed technique.

After identifying the *k* recommended items from different log files for the user *u* and for the query Q, *k* is displayed in the search engine interface. [31] Suggested that the success of recommendations relies on explaining the cause behind them. This is the motivation factor for providing an explanation along with each suggested item, i.e., for explaining why this specific recommendation appears in the top-k list. The recommendations along with their explanations are represented by using a simple template mechanism or tool tip text. The recommended items are the user's favourite query, ranked past queries issued the user by analyzing their real intent from the query log, concepts with high support and the query terms from the similar users. The format of the recommended queries is given in Figure 4. The italic texts are tool tip texts that show the reason why the recommended items are in the list.

In this approach, the used database is processed and stored in 4GB RAM and 320 GB Hard disk using SQL Server as a database engine and .NET Framework is used to design the interface. Figure 5 shows the interconnection between the databases used in the proposed method.

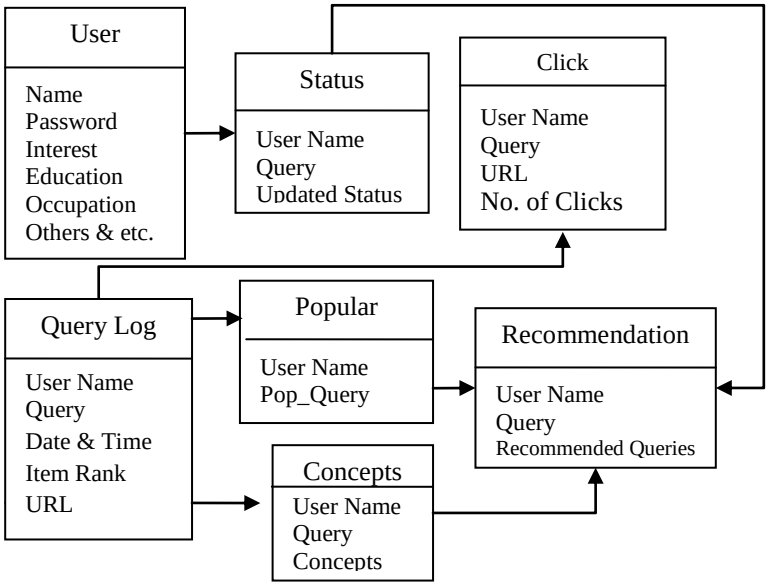

Fig. 5: Data model used in the Query Recommendation Technique

3.6 Ranking of Recommended Queries

The recommended queries from query log are re-ranked using its number of clicks and the concepts are re-ranked using their support value. When the Concepts have equal number of clicks then it is re-ranked according to t-measure. The first recommendation is always the favourite query of the user. Different ranking models are explained in [26]. Figure 6 shows that how the queries are re-ranked in the recommendation.

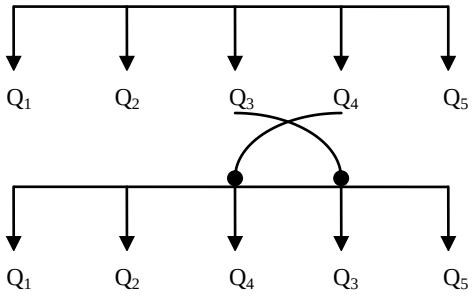

Fig. 6: Re-ranking of Queries

4. EXPERIMENTAL RESULTS

The algorithm is implemented in .NET framework. All the experiments are performed in Intel Core i3 processor 2.53 GHz with Windows 7 Home Premium (64-bit) and 4 GB RAM. To evaluate the performance of the proposed work, we developed an interface for Google to collect the click through data. The interface can be applied to interpret any search engine. The queries are selected from ten different categories and are assigned to 10 different users. Before the interface have been used, the users are instructed to create their basic profile by using the registration form. Table 5 lists the 10 different query categories.

Table 6 shows the statistics of the data environment. If the search query is issued by the unregistered users then the search results from Google is retrieved as it is.

Table 5. Query Categories

Query Id.	Query Category	Query Id.	Query Category
1	Cricket	6	Fruit
2	Hospital	7	Insurance
3	Entertainment	8	Education
4	Travels	9	Gold
5	Mobile	10	News

Set of 50 query keywords under10 different categories are listed in Table 7. Some of the query keywords are overlapping in the categories. For example the query keyword ‘sports’ appears in the categories ‘Cricket’ and ‘Entertainment’. Some of the query keywords have ambiguous meanings. For example the query ‘apple’ refers as a fruit and also computers.

Table 6. Statistics of data environment

Number of Users	10
Number of Query Categories	10
Number of Test Queries	50
Number of Queries assigned to each user	5
Number of unique queries	48
Maximum Number of web snippets considered for the query	10
Maximum Number of URLs retrieved for a query	10
Number of URLs retrieved	449
Number of unique URLs retrieved	422
Number of concepts	2882
Number of unique concepts	1739
Maximum number of extracted concepts for a query	495

The registered user can update their profile for every query. These updates are stored in the database log of the user. For example, the user may like cricketer ‘MS Dhoni’ and he updates the status regarding him. For the next search regarding ‘cricket’, the search engine will give the result of ‘cricket’ and the recommendation will be on ‘Ms Dhoni’ as

Table 7. Query keywords used in the system

Cricket	Hospital	Entertainment	Travels	mobile
cricket	Hospital	Entertainment	travels	mobile
live cricket matches	clinic for women	Cricket	travel agencies at coimbatore	applications of mobile
live cricket score	list of hospitals	Cinema	travels online bus booking	mobile technology
match fixing	health care	entertainment news	travel agency	mobile communication
sports	open heart surgery	Sports	travels time spent	latest mobile phones
Fruit	**Insurance**	**Education**	**Gold**	**News**
fruit	Insurance	Education	gold	News
apple	life insurance corporation	education institutions	gold history	today news
fruit salad	vehicle insurance	school teaching	gold rate	sports news
fruit advantages	insurance amount calculator	higher education	comparison of gold rate in the world	newspaper
vitamin fruits	insurance policies	education in news	gold making	news in media

Table 8. Analysis of 10 Different Query Categories

Query Category	Location	Number of Words	Number of Concepts	Number of Unique Concepts	Number of Redundant Concepts
cricket	Title	362	121	74	47
	Abstract	875	359	242	117
	Both	1237	389	240	149
hospital	Title	248	114	80	34
	Abstract	587	379	298	81
	Both	836	423	300	123
entertainment	Title	308	122	80	42
	Abstract	860	458	347	111
	Both	1173	495	349	146
travels	Title	290	112	75	37
	Abstract	740	308	224	84
	Both	1029	359	247	112
mobile	Title	265	110	77	33
	Abstract	770	416	310	106
	Both	1036	456	321	135
fruit	Title	269	121	83	38
	Abstract	662	406	312	94
	Both	919	465	332	133
insurance	Title	356	90	50	40
	Abstract	732	279	197	50
	Both	1038	312	204	108
education	Title	243	113	81	32
	Abstract	656	424	348	76
	Both	896	472	356	116
gold	Title	274	108	69	39
	Abstract	760	389	287	102
	Both	1039	444	318	126
news	Title	333	105	66	39
	Abstract	761	304	193	111
	Both	1093	338	205	133
Complete Data set	Title	2758	783	477	306
	Abstract	7410	2605	1636	969
	Both	10168	2882	1739	1143

the user has updated. Table 8 shows the number of words, number of concepts, number of unique concepts and number of redundant concepts occurred in our dataset. But the proposed system considers only the concepts in clicked snippets.

Figure 7 shows the variation between the retrieved concepts and unique concepts occurred in different query categories. Figure 8 depicts the words, concepts and unique concepts counts.

Query Category	Concepts	Unique Concepts
Cricket	31.45	61.70
Hospital	50.60	70.92
Entertainment	42.20	70.51
Travels	34.89	68.80
Mobile	44.02	70.39
fruit	50.60	71.40
insurance	30.06	65.38
education	52.68	75.42
gold	42.73	71.62
news	30.92	60.65

Fig. 7: Concepts Vs Unique Concepts

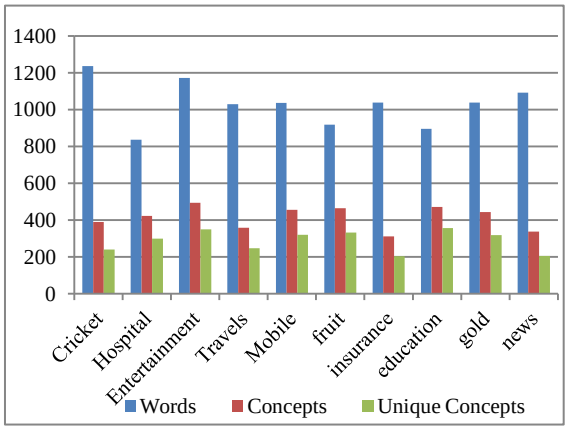

Fig. 8: Analysis of 10 different Query Categories

Table 9 lists the pre-processed concepts from the clicked web snippets for the query 'cricket' issued by the user on 25-6-2013 10:25:11 am. The concepts are pre-processed by using the methods in section 3.4.

Support value of the pre-processed concepts is calculated. The support value which satisfies the minimum threshold is considered for finding the similarity. Table 10 shows that set of concepts extracted for the query 'cricket'.

Table 10. Concepts extracted for the query 'cricket'

Concpet C_i	Support(C_i)	Concpet C_i	Support(C_i)
Indian	2	Ipl	2
Cricket	3	spot fixing	3
Fixing	3	ipl spot	2
Spot	3	ipl spot fixing	2

Figure 9 shows that list of words, concepts, unique concepts and its percentage appeared in the title of the clicked snippets for the query 'cricket'. The concepts from the query Q is consistent [10] if they co-exist frequently in the locations of title, abstract and tags of the web snippets are retrieved for the query Q. Table 11 illustrates the similarity between the concepts which is obtained by using the formula (3) in section 3.4.

Table 11. Similarity Matrix

Concept	*indian*	*cricket*	*fixing*	*spot*	*ipl*	*spot fixing*	*ipl spot*	*ipl spot fixing*
indian	-	0.8	0.25	0	0.2	0	0	0
cricket	**0.8**	-	0	0	0	0.066	0.066	0.066
fixing	0.25	0	-	0.366	0.25	0.366	0.366	0.2
spot	0	0	0.366	-	0.366	0.366	0.366	0.066
ipl	0.2	0	0.25	0.366	-	0.5	0.4	0.4
spot fixing	0	0.066	0.366	0.366	**0.5**	-	0.5	0.366
ipl spot	0	0.066	0.366	0.366	0.4	**0.5**	-	0.4
ipl spot fixing	0	0.066	0.2	0.066	0.4	0.366	0.4	-

Table 9. Pre-processed Concepts in the Clicked Snippets

Document	Title	Abstract	Tags
Clicked Document 1	reputation indian cricket hit fixing scandal ms dhoni	skipper ms dhoni indian cricket reputation tarnished hopeful indias show upcoming	mahendra singh dhoni, msd, dhoni, fixing, scandal, spot fixing, cricket
Clicked Document 2	srinivasan forced quit bcci secretary lele	beleaguered bcci president n.srinivasan eventually step choice honourably exit	bcci president, srinivasan, n.srinivasan, jaywant lele, ipl spot fixing, indian premier league, cricket
Clicked Document 3	ipl fixing sreesanth distract delhi police jail one india	ipl spot fixing kerala pace bowler sreesanth distract police jail ajit chandila ankeet chavan.	sreesanth, ipl spot fixing, cricket, delhi police

Personalized Query Recommendation using Concept Based Hybrid User Profile

Enter the Query : cricket

Enter the Web snippet File Name : E:/cricket/criket title.txt

Clicked Concepts occur in Web snippet's Title

List of Words	List of Concepts	Concepts with Count	Concept with Support (%)
espn	espn	espn 1	Espn 1 33%
cricinfo	cricinfo	cricinfo 1	cricinfo 1 33%
live	live	live 3	live 3 100%
cricket	cricket	cricket 8	cricket 8 100%
score	score	score 4	score 4 100%
commentary	commentary	commentary 1	commentary 1 33%
match	match	match 1	match 1 33%
coverage	news	news 2	news 2 67%
yahoo	coverage	coverage 1	coverage 1 33%
cricket	yahoo	yahoo 1	yahoo 1 33%
live	schedule	schedule 1	schedule 1 33%
cricket	ipl	ipl 1	ipl 1 33%
score	rediffcom	rediffcom 1	rediffcom 1 33%

cricket
news
cricket
schedule
cricket
live
cricket
score
ipl
score
cricket
news
rediffcom

Number of words....:26
Number of Concepts....:12
Number of unique concepts...9
Number of redundant concept...3
Concept cricket occursfrequently at 8 times

Fig. 9: Concepts Extraction

Consider the similarity threshold value is 0.5. Fig. 9 shows the concept relation graph built for the query 'cricket'. The concept pairs which satisfies the threshold value is {(indian, cricket), (ipl, spot fixing), (spot fixing, ipl spot)}. Here we recommend the queries 'indian cricket' and 'ipl spot fixing' for the input query 'cricket' from the concept log. The dotted line in Fig.10 intimates the recommended concepts.

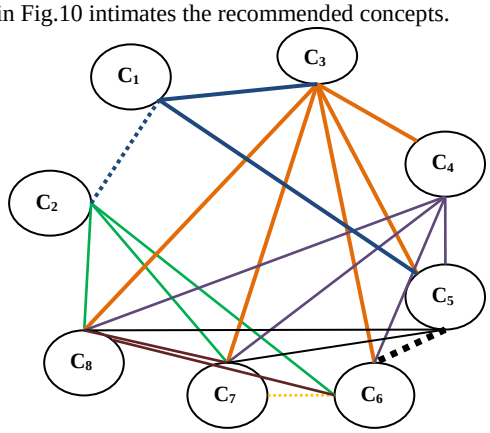

Fig. 10: Concept Relation Graph

The output format of the proposed algorithm is compared with the familiar search engines Google and Bing. The search engines retrieve the static recommendation for the initial query 'cricket' on 8-7-2013 10:12:24. Both return 8 recommendations which have occurred every time. Figure 11 and Figure 12 depict the recommended queries from Google and Bing respectively.

Fig. 11: Recommendation from Google for 'cricket'

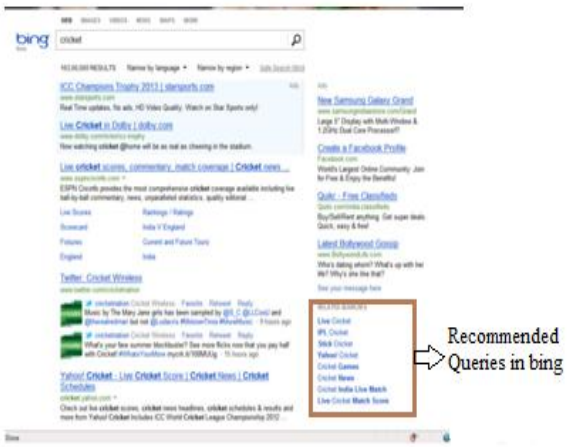

Fig. 12: Recommendation from Bing for 'cricket'

If the user is not a registered then the system retrieves the resultant snippets and recommended queries from Google. But, when the user is logged, the top-k recommendations are displayed as mentioned in Section 3.5. Figure 13 shows the recommendations of the proposed system.

Recommendation Evaluation

The proposed recommendation is evaluated by using an evaluation form. The users are asked to search in one query category. On the evaluation form, the users are asked to give the relevancy score for the recommended queries. For each recommended query, the user had to label it with a relevancy score {0, 1, 2} where 0: irrelevant, 1: partially relevant, and 2: relevant. Table 11 shows the relevancy score for the query 'cricket'. The number of recommended queries is varied and depends on the intent of the user. In Table 12, {R1, R2….R7} indicates the recommended queries. Here R1 is always the favourite query of the user. It may be irrelevant many times.

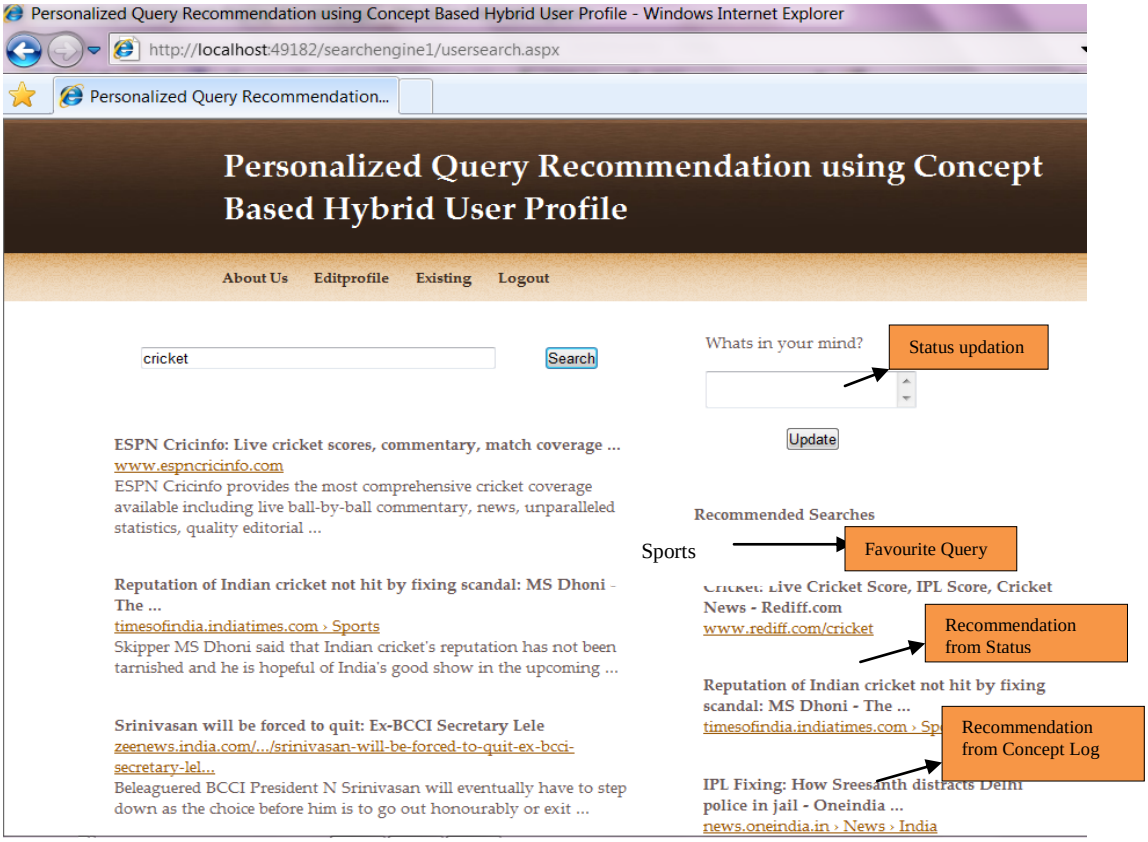

Fig. 13: Recommendations for Registered User

Table 12. User relevancy score

Query : cricket	R1	R2	R3	R4	R5	R6	R7
User 1	0	2	2	2	1	--	--
User 2	1	2	2	1	1	1	0
User 3	0	2	1	1	--	--	--
User 4	0	2	1	--	--	--	--
User 5	2	2	2	1	1	--	--
User 6	2	2	1	1	0	0	--
User 7	0	2	2	1	1	--	--
User 8	0	2	2	2	--	--	--
User 9	1	2	1	1	1	--	--
User 10	1	2	2	1	--	--	--

Figure 14 shows that most of the users scored the recommended queries are either relevant or most relevant. Only 5 users is selected their favourite queries are irrelevant for the initial query 'cricket'.

5. CONCLUSION

The availability of web pages comprises umpteen data in the form of documents, images and multi-media contents. A survey carried out by Netcraft, Internet Services Company, reports that there are 739,032,236 sites in September 2013 and 22.2M which seems more than the month August 2013. Hence search engines play a vital role in web information retrieval process. Query Recommendation provides a set of alternate queries which may be used in future and it satisfies the user's real information need. The proposed technique recommends the queries and the resultant queries are re-ranked. The recommendation process is personalized by favourite query of the user, concepts retrieved from the clicked web snippets, user's explicit profile information and the search behaviour of similar users. The recommendation process of the proposed method is evaluated. This technique can be applied to any recommender systems.

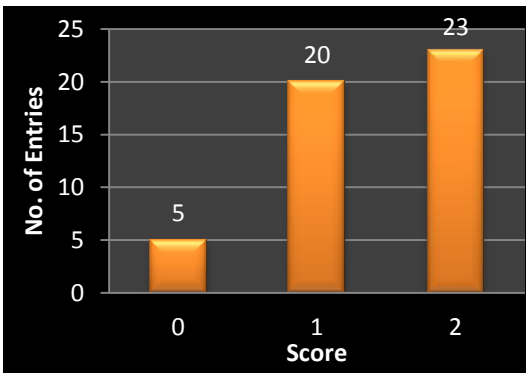

Fig. 14: Recommended Query evaluation

6. REFERENCES

[1] Wen J. R., Nie J. Y. and Zhang H. J. 2001. Clustering user queries of a search engine. In Proceedings of the 10th international conference on World Wide Web, ACM, pp. 162-168.

[2] Baeza-Yates R., Hurtado C. and Mendoza M. 2007. Improving search engines by query clustering. Journal of the American Society for Information Science and Technology 58, No. 12, pp. 1793-1804.

[3] Baeza-Yates R., Hurtado C., Mendoza M. and Dupret G. 2005. Modeling user search behaviour. In Web Congress, IEEE, pp. 10

[4] Grimes C., Tang D. and Russell D. M. 2007. Query logs alone are not enough. In Workshop on query log analysis at WWW.

[5] Hsieh-Yee and Ingrid. 2001. Research on Web search behaviour. Library & Information Science Research, 23, No. 2, pp. 167-185.

[6] Speretta M. and Gauch S. 2005. Personalized search based on user search histories. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, pp. 622-628.

[7] Wen J. R., Nie J. Y. and Zhang H. J. 2002. Query clustering using user logs. ACM Transactions on Information System, 20, No. 1, pp. 59-81.

[8] Zigoris P. and Zhang Y. 2006. Bayesian adaptive user profiling with explicit & implicit feedback. In Proceedings of the 15th ACM international conference on Information and knowledge management, ACM, pp. 397-404.

[9] Sugiyama K., Hatano K. and Yoshikawa M. 2004. Adaptive web search based on user profile constructed without any effort from users, In Proceedings of the 13th international conference on World Wide Web, ACM, pp. 675-684.

[10] Dupret G. and Mendoza M. 2005. Recommending better queries based on click-through data. In Proceedings of the 12th International Symposium on String Processing and Information Retrieval, SPIRE, pp. 41-44.

[11] Silvestri Fabrizio. 2010. Mining query logs: Turning search usage data into knowledge. Foundations and Trends in Information Retrieval, Vol. 4, No. 1, pp. 1-174.

[12] Joachims T., Granka L., Pan B., Hembrooke H. and Gay G. 2005. Accurately interpreting clickthrough data as implicit feedback. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 154-161.

[13] Chen C.C., Chen M. C. and Sun Y. 2002. PVA: A self-adaptive personal view agent. Journal of Intelligent Information Systems, Vol. 18, No. 2-3, pp. 173-194.

[14] Claypool M., Le P., Wased M. and Brown D. 2001. Implicit interest indicators. In Proceedings of the 6th international conference on Intelligent user interfaces, ACM, pp. 33-40.

[15] Pazzani M. J. and Billsus D. 2007. Content-based recommendation systems. In the adaptive web, Springer Berlin Heidelberg, pp. 325-341.

[16] Baeza-Yates R., Hurtado C. and Mendoza M. 2004. Query recommendation using query logs in search engines. In Current Trends in Database Technology-EDBT Workshops, Springer Berlin Heidelberg, pp. 588-596.

[17] Baeza-Yates R. and Tiberi A. 2007. Extracting semantic relations from query logs. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 76-85.

[18] Cucerzan Silviu and Ryen W. White. 2007. Query suggestion based on user landing pages. In Proceedings of the 30th annual international conference on Research and development in information retrieval, ACM SIGIR, pp. 875-87.

[19] Fonseca B. M., Golgher P. B., de Moura, E. S. and Ziviani N. 2003. Using association rules to discover search engines related queries. In Web Congress, IEEE, pp. 66-71.

[20] Liu Y., Miao J., Zhang M., Ma S. and Ru L. 2011. How do users describe their information need: Query recommendation based on snippet click model, Expert Systems with Applications Vol. 38, No. 11, pp.13847-13856.

[21] Stefanidis K., Drosou M. and Pitoura E. 2009. You May Also Like" results in relational databases. In Procedings of the conference on PersDB, Lyon, France.

[22] Chatzopoulou G., Eirinaki M. and Polyzotis N. 2009. Query recommendations for interactive database exploration, In Scientific and Statistical Database Management, Springer Berlin Heidelberg, pp. 3-18.

[23] Khoussainova N., Kwon Y., Balazinska M. and Suciu D. 2010. Snip Suggest: context-aware auto completion for SQL. In Proceedings of the VLDB Endowment, Vol. 4, No. 1, pp.22-33.

[24] Golfarelli M., Rizzi S. and Biondi P. 2011. myOLAP: 2011. An approach to express and evaluate OLAP preferences. IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No. 7, pp. 1050-1064.

[25] Khemiri R. And Bentayeb F, 2012. Interactive Query Recommendation Assistant. 23rd International Workshop on Database and Expert Systems Applications (DEXA), IEEE, pp. 93-97.

[26] Umagandhi R. and Senthil Kumar A.V. 2013. Time Heuristics Ranking Approach for Recommended Queries using Search Engine Query Logs. Kuwait journal of Science and Engineering, communicated.

[27] Agichtein E., Brill E., Dumais S. and Ragno, R. 2006. Learning user interaction models for predicting web search result preferences. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 3-10. ACM.

[28] Beeferman D. and Berger A. 2000. Agglomerative clustering of a search engine query log. In Proceedings of the sixth international conference on Knowledge discovery and data mining, ACM SIGKDD, pp. 407-416.

[29] Leung KW-T. and Dik Lun Lee. 2010. Deriving concept-based user profiles from search engine logs. IEEE Transactions on Knowledge and Data Engineering, Vol. 22, No. 7, pp. 969-982.

[30] Joachims T. 2002. Optimizing search engines using click through data. In Proceedings of the eighth international conference on Knowledge discovery and data mining, ACM SIGKDD, pp. 133-142.

[31] Stefanidis K., Ntoutsi I., Norvag K. and Kriegel H. P. 2012. A framework for time-aware recommendations. In Database and Expert Systems Applications, Springer Berlin Heidelberg, pp. 329-344.