Kuwait J. Sci. **41** (2) pp. 127-149, 2014

Time heuristics ranking approach for recommended queries using search engine query logs

R. UMAGANDHI * AND A. V. SENTHIL KUMAR **

* *Associate Professor and Head, Department of Computer Technology, Kongunadu Arts and Science College, Coimbatore. umakongunadu@gmail.com*
** *Director, MCA, Hindusthan College of Arts and Science, Coimbatore. avsenthilkumar@yahoo.com*

ABSTRACT

It is obvious that web search queries given by the user are always short and ambiguous. Mostly the shorter length queries do not satisfy the users real information need and may not produce the results properly. Query Recommendation is a technique based on the real intent of the user and to provide the alternate queries to frame the queries in the future. The proposed work recommends the queries for four types of users in three ways (1) Favourite queries of the user are identified and they are recommended. (2) Users who have similar interest are clustered; the recommendation is given from the access logs of similar users. (3) Similar queries are clustered; the favourite query of the cluster is identified and it is recommended. The proposed work also ranks the recommended queries based on the preference and access time of the query. The proposed strategies are experimentally evaluated using real time search engine query log.

Keywords: Favourite Query; preferences; t-measure; frequent query pattern; query log.

INTRODUCTION

Search engines always play an important role in the web information seeking process and they are used to retrieve the results in terms of web snippets from the web repository. Generally the query issued in the searching process may belong to one of the categories; Informational query, Navigational query and Transactional Query (Liu *et al.*, 2011b). The user issuing the query and viewing some contents in the web snippet itself belongs to the category of Informational Query (e.g. 'dollar rate'). The user issuing the query and clicking the sequence of URLs in a session by using the hyperlink and getting the information is called as Navigational query. Here the users intent is to see the web sites (e.g. 'Inheritance in Java') and the recommended queries are mostly used. Transactional query performs some online activities (e.g. railway ticket reservation).

The user scans the search result from the top to the bottom according to Joachim *et al.* (2005) and then decides whether the resultant web snippet is relevant or irrelevant. A study carried out by Silverstein (1999) on "private" Alta Vista Query Log has shown that more than 85% of queries contain less than three terms and the average length of the queries are 2.35 with a standard deviation of 1.74. For the second AltaVista log, instead, the average query length is slightly above 2.55. It is to be understood that the shorter length queries do not provide any meaningful, relevant and needed information to the users.

Interpreting the human queries into search keyword is never straightforward (Baraglia *et al.*, 2009). Especially search engine users are inexperienced and they are usually casual users. They have very limited background knowledge about the domain they are searching for. Search engines provide the assistance to frame the queries in the form of automatic query completion (Chirita *et al.*, 2007; Mei *et al.*, 2008) at hitting time and query recommendation (Ma *et al.*, 2010; Li *et al.*, 2012; Baeza-Yates *et al.*, 2005). Human mentality is to get a choice for everything and select an option from the given choice. The proposed work deals with the query recommendation technique. The major contributions of the proposed method are summarized as follows:

- The query log entries are pre-processed and analysed.
- Favourite query of every user is identified from the query log file. Frequently occurred query patterns are generated based on the input query.
- Users with similar intent are clustered. The recommendation is given to the user from the queries and click-thru of similar users.
- Similar queries which either share some common keywords or URLs are clustered. Favourite query of each query cluster is identified and it is recommended.
- The recommended queries are ranked based on the preference and time on which the queries are triggered. The correlation between the ranking orders is evaluated using spearmans correlation coefficient.
- The ranking techniques are evaluated by using existing ranking measures.

The rest of the paper is organised as follows: Section 2 reviews the related work. Section 3 defines the input query log used in the proposed work. Section 4 gives the Architecture of the proposed work. Section 5 discusses the favourite query finder, Generation of query clusters and user clusters for recommendation process. Section 6 discusses the experiments and results. Finally the paper is concluded in Section 7.

RELATED WORKS

Due to the enormous growth of the web and lack of users knowledge, query recommendation becomes an important technique used by the search engine users to get the desired information from the web. CNNIC (2009) search behaviour survey report says that 78.2% of the users change their query by using the recommended queries. Users may select the recommended queries instead of framing the new queries. Recommendation methods are organized into three main categories (Stefanidis *et al.*, 2009; Khemiri & Bentayeb, 2013): Content-based approach (Khemiri & Bentayeb, 2012) gives the recommendation based on the past queries and navigational behaviour of individual user; Collaborative approach (Golfarelli *et al.*, 2011) is based on the preferences of other similar users that the queries from similar users are recommended; Hybrid approach (Stefanidis *et al.*, 2009) combined both content-based and collaborative approach.

The input of the 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.

Table 1. Query Recommendation Approaches - A Comparison

Research Works		Stefanidis *et al.*, 2009	Chatzopoulou *et al.*, 2009	Khoussainova *et al.*, 2010	Golfarelli *et al.*, 2011	Khemiri *et al.*, 2012	Proposed Method
Recommendation Type	Content Based	√				√	√
	Collaborative	√	√	√	√		√
Recommendation Time	While			√		√	
	After	√	√		√		√
Recommendation Input Data	Log file	√	√	√		√	√
	User Profile					√	

The proposed method follows hybrid approach, which provides the recommendation after querying and it uses the query log file as the input. The queries are recommended from the access log of the similar users. Much research has been done in query expansion, Query suggestions and Query recommendations. Query expansion techniques utilize the dictionary to get the expansion of the query keyword, but query recommendation refers the query log. Search behaviour of the users is analysed by using the query log and semantic meaning of the query have been described by Baeza-Yates *et al.*, (2005). Neelam & Sharma (2010). describes the recommendation using the semantic meaning of the

input query and the semantics have been identified from yourdictionary.com. Cucerzan & White (2007) finds the query keywords similarity and click URL similarity. Many researchers have used this similarity measure to cluster the similar queries. Liu *et al.* (2011a) have recommended the query in which keywords are recommended because of their appearance in clicked snippets instead of similarity with previous one. The recommendation process is analysed based on the users perspective. The recommendation is based on the snippet click model and there is a possibility for redundant recommendations.

QUERY LOG

Search engine leaves the search information to the user for further references in query logs. Query log is an important repository, which records the users search activities. The mining of these logs can improve the performance of search engines (Neelam & Sharma, 2010). In order to give the recommendations to frame the future queries, the search histories in the query log are analysed. To evaluate this work, query log of AOL (American onLine) search engine data set from 2006-03-01 to 2006-05-31 (zola. di. unipi. it / smalltext /datasets.html) is considered. The search histories are organized under the attributes:

AnonID, Query, QueryTime , ItemRank , ClickURL

Umagandhi & Senthilkumar (2013) described the above attributes. Table 2 shows the sample log entries in the data set.

Table 2. Sample log entries

AnonID	Query	QueryTime	ItemRank	ClickURL
1038	tow truck	2006-03-01 23:17:31	NoClick	NoRank
1038	kris stone	2006-03-15 23:19:22	NoClick	NoRank
227	psychiatric disorders	2006-03-02 17:30:36	1	http://www.merck.com
227	Cyclothymia	2006-03-02 17:34:08	1	http://www.psycom.net

In the first two rows, the user 1038 either obtains the information from the web snippets itself or is not satisfied with the result; hence the user does not click any URL. Other rows contain the data for all the attributes. If the user clicks more than one URL from the returned result for a single query, then there will be successive entries in the access log. The query log entries are pre-processed (Umagandhi & Senthilkumar, 2013) and the unique queries are retrieved. An ID is assigned to each unique query. Some of the basic definitions used in the recommendation process are given below.

Support: (Han & Kamber, 2006) An item set X has support s in T if s% of the transactions in T contains X. Support of query Q is the number of times the Query triggered by the user U.

Preference: The user $\in U$ may express a preference for the query $q \in Q$, which is denoted by Preference (u, q) and lies in the range [0, 1]. Here day wise and query wise preference is considered to obtain the Preference (u, q).

$$Day\_Preference(u,q) = \frac{(Number\ of\ days\ q\ triggerd\ by\ u)}{(Total\ number\ of\ days)} \tag{1}$$

$$Query\_Preference(u,q) = \frac{(Number\ of\ times\ q\ triggerd\ by\ u)}{(\sum_{i=1}^{n} Number\ of\ times\ q_i triggered\ by\ u)} \tag{2}$$

$$Preference(u,q) = \alpha * Day\_Preference(u,q) + \beta * Query\_Preference(u,q) \tag{3}$$

Where n = number of queries given by u. Preference is normalized by using the constant parameters α and β, where $\alpha + \beta = 1$.

Time Schema: Time Schema $T = (R, C)$ where R is set of access logs with time attribute and C is the Constraint. For example, It is to be considered that the Time Schema *year: 2006, month:* {*3, 4, 5*}, *day:* {*1, 2, 3,..., n*} where n = {*31 for month = 3, 5 and 30 for month = 4*} with the constraint that evaluate $< y, m, d >$ to be "true" only if the combination gives a valid date in the range of 2006-03-01 to 2006-05-31.

Time Cluster: Time Cluster $D = (Q, T)$, where Q is the identifier assigned to each unique query in the access log and T is the time period on which Q is triggered. For example, the time cluster for our data set $D = (Qi, T)$ 1, 2, ... , 54, number of unique queries triggered in the first 200 access log is 54.

t-measure: If query q_1 is accessed in two different time periods t_1 and t_2 (t_1 occurs earlier than t_2) then the t-measure of q_1 at t_1 is lesser than the t-measure of q_1 at t_2.

$$t - measure(q_i) = Cluster\ number\ (q_i) / \sum_{i=1}^{n} i \tag{4}$$

Where n = number of clusters. For example consider the day wise query cluster in Table 3.

Table 3. Query cluster of the user 1038

Cluster Number	Date	Query ID.
1	2006-03-01	7,52, 51, 18, 80, 17
2	2006-03-14	8,10
3	2006-04-02	7,9
4	2006-05-01	7

The query 7 occurs in three clusters namely cluster 1, cluster 3 and cluster 4. The t-measure of the query 7 is

t-measure (7 at cluster 1) = 1 / (1 + 2 + 3 + 4) = 0.1

t-measure (7 at cluster 3) = 3 / (1 + 2 + 3 + 4) = 0.3

t-measure (7 at cluster 4) = 4 / (1 + 2 + 3 + 4) = 0.4

t-measure assigns the weight according to the earlier or recent access. t-measure for the query 7 at the first cluster is 0.1, second cluster is 0.2, third cluster is 0.3 and at the last cluster is 0.4.

$$Total\ t-measure(q) = \sum_{i=1}^{n} t-measure(q\ at\ cluster\ i) \tag{5}$$

Where $n =$ number of clusters where the query q appears. For example the total t-measure of the query 7 is

$$Total\ t-measure(7) = \sum_{i=1}^{4} t-measure(7\ at\ cluster\ i) = 0.8$$

Sum of the t-measure of the clusters is equal to 1. That is

$$Sum(t-measure) = \sum_{i=1}^{n} t-measure(cluster\ i) = 1 \tag{6}$$

Where $n =$ number of clusters. For example, total t-measure for the clusters in Table 3 is $0.1 + 0.2 + 0.3 + 0.4 = 1$.

ARCHITECTURE

Figure 1 describes the architecture for time dependent recommendations. The user submits the query through search engine interface. The users request and their navigational behaviours are recorded in the query log file. The user scans the search result from the top to the bottom and decides that the retrieved results are either relevant or irrelevant for their request. Sometimes the user scans the search result and will be satisfied with the information available in the abstract of the web snippets itself. For these cases the user does not click any URL, so the message "NoClick" is assigned to the attribute ClickURL.

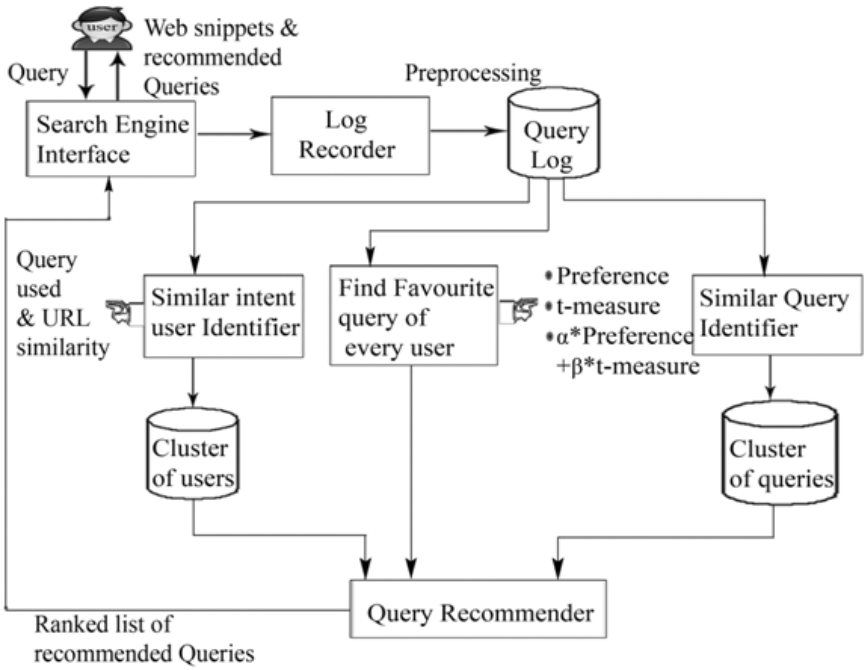

Fig. 1. Architecture of the proposed work

The pre-processed log entries are stored in the query log file. The first module in Figure 2 describes the Similar Intent User Identifier; the similarity between the users is identified by using the day wise query access. The input queries and the Clicked URLs are used to identify the similar users. Then the queries from the similar users are given as the recommendation for the input query. Second module finds the favourite query of the user; here the favourite or popular query of every user is identified using the preference of the query and t-measure. The favourite query is the first choice in the recommendation process. Third module is similar query identifier, the similar queries based on keywords and URLs are identified and clustered. This cluster recommends the similar queries for the input query.

QUERY RECOMMENDATIONS

The proposed work generates the recommendation for the input query in the following manner:

The favourite query of the user is identified by analyzing the access behaviour and it is recommended which is the first choice of the recommendation for all the queries issued by the user. The query and access behaviour of various users

are analysed and the similarity matrix is generated. Similar users are clustered using the Agglomerative Hierarchical Clustering Algorithm (Beeferman & Berger, 2000). This cluster is used to generate the recommendation. Finally the similar queries are clustered based on the query concept. This concept based cluster is used for recommendation. Fig. 2 shows the recommendation process of different kinds of users. Consider user U and query Q, the user U may belongs to the following categories:

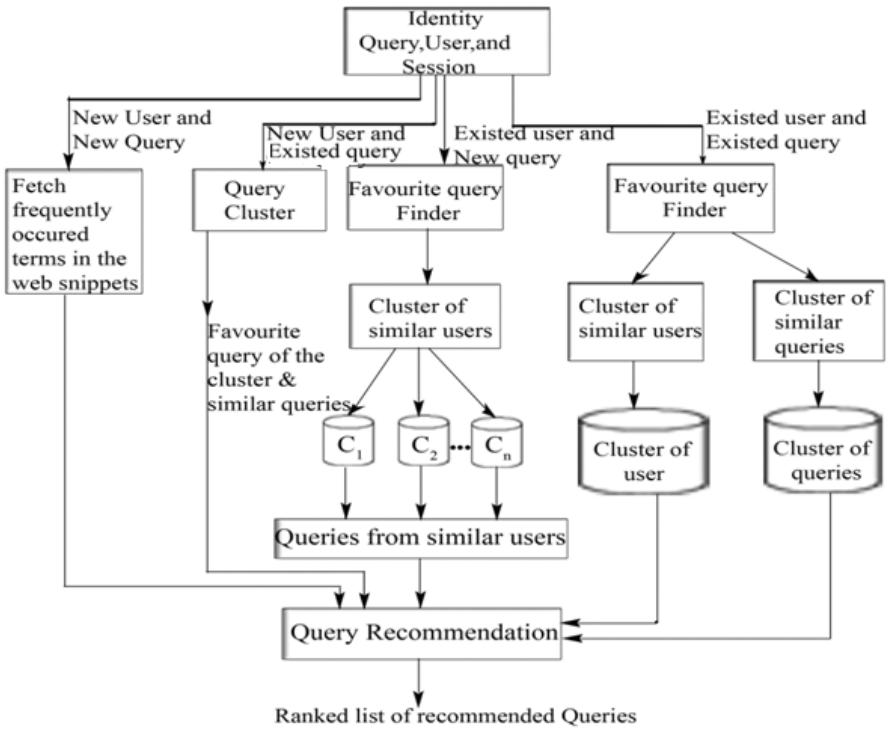

Fig. 2. Recommendation process of different users

Case 1: U is new to the search process, that is, the query log receives the first entry of *U* and *Q* is also new. In this situation, either the top concepts retrieved from the web snippets or constant recommendations from the database are recommended.

Case 2: U is new and the query keyword *Q* already exists. It is not possible to find the favourite query of the user, real intent and similar users. Based on *Q*, similar queries are identified and clustered. Each cluster forms one query concept. This query cluster is used to provide the recommendations to the user *U*. Here the process is purely collaborative recommendation.

Case 3: U already exists in the query log but *Q* is new. Here the favourite query of the user is identified. Similar users are clustered and the cluster is used for recommendation. Here the recommendation process is hybrid approach.

Case 4: U and *Q* existed in the query log. In this situation, favourite query of the user, similar queries from the query cluster and the access log of similar users are used in the recommendation process. Here also the recommendation is a hybrid one.

Favourite Query Finder

A query is said to be favourite when it occupies major portion of the search requests in the access log of the user. Algorithm Favourite Query Finder identifies the favourite query and is stored in the form of pairs.

Algorithm Favourite Query Finder

Input: set of user wise access logs

Output: user wise favourite query

begin

Step 1: Identify the distinct queries fired by the user

Step 2: For (each distinct user *u*)

For (each distinct query *q*)

- Calculate the Preference of *q*
  Preference (u, q) $= \alpha$ * Day_Preference (u, q) $+ \beta$ * Query_Preference (u, q)
- Find the day wise cluster of queries
- Calculate t-measure of *q* using $t\text{-}measure(q_i) = Cluster\ number(q_i) / \sum_{i=1}^{n} i$
- Calculate the combined measure
  Preference with t-measure $= \alpha$ * preference (u, q) $+ \beta$ * t-measure (q)

Step 3: Store $< AID, Favourite\ query >$ pairs

Step 4: Return the maximum weighted query as the favourite query

end

The user and his activities around 5 days are considered where $Qi, 1 \leq i \leq 6$ are the queries triggered by the user on Dayj, $1 \leq j \leq 5$.

Day 1 - Q1, Q3, Q4 Day 2 - Q1, Q4, Q5 Day 3 - Q1, Q2, Q3, Q6

Day 4 - Q3, Q4, Q5 Day 5 - Q1, Q2, Q6

The queries Q1, Q3 and Q4 are given by the user on Day1. Table 4 depicts the support, confidence, preference and t-measure for the above day wise activities.

Table 4. Preference & t-measure

Query	Support	Confidence	Preference	t-measure
Q1	4	80	0.525	0.733
Q2	2	40	0.263	0.533
Q3	3	60	0.394	0.533
Q4	3	60	0.394	0.466
Q5	2	40	0.263	0.4
Q6	2	40	0.263	0.533

Based on the preferences, the queries are ranked as {Q1, (Q3, Q4), (Q2, Q5, Q6)}. Here the queries Q3 and Q4 are grouped because they have equal preference. Preferences alone not produce the good ranked list for queries; It may be considered that the t-measure, which describes recently accessed queries is weighted high compared with the weight of earlier access. The queries are ranked as {Q1, (Q2, Q3, Q6), Q4, Q5} based on t-measure. The query is favourite when its preference is high and also it has recent access. The queries are ranked by considering the combined measure

$$Preference\ with\ t-measure = \alpha * preference(u,q) + \beta * t-measure\ (q) \quad (7)$$

Table 5 shows combined measure value for different α values. If $\alpha = 0.5$ then $\beta = 0.5$ and the queries are ranked as {Q1, Q3, Q4, (Q2, Q6), Q5}.

Table 5. Preference with t-measure values

Query	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$
Q1	0.712	0.671	0.629	0.587	0.546
Q2	0.506	0.452	0.398	0.344	0.29
Q3	0.519	0.491	0.464	0.436	0.408
Q4	0.459	0.444	0.43	0.416	0.401
Q5	0.386	0.359	0.332	0.304	0.277
Q6	0.506	0.452	0.398	0.344	0.29

Table 6 shows the changes in the ranking order according to the α value. For all the cases, irrespective of the constant parameters α and β, favourite query of the user is *Q1*. The queries *Q2* and *Q6* have equal weight and the query *Q5* is least accessible.

Table 6. Ranking of Queries

α	β	**Ranking of queries**
0.1	0.9	Q1, Q3, (Q2, Q6), Q4, Q5
0.3	0.7	Q1, Q3, (Q2, Q6), Q4,Q5
0.5	0.5	Q1, Q3, Q4, (Q2, Q6), Q5
0.7	0.3	Q1, Q3, Q4, (Q2, Q6), Q5
0.9	0.1	Q1, Q3, Q4, (Q2, Q6), Q5

Table 7 shows the changes in the ranking order of 6 queries by using the ranking techniques preference, t-measure and preference with t-measure. Average ranking is assigned to the queries when they have same measure. For example, the queries Q3 and Q4 have the same preference 0.394; hence the rank 2.5 is assigned for Q3 and Q4 instead of 2 and 3 respectively.

Table 7. Ranking order

Original	**Preference**	**t-measure**	**Preference + t-measure (α = 0.5)**
1	1	1	1
2	5	3	4.5
3	2.5	3	2
4	2.5	5	3
5	5	6	6
6	5	3	4.5

Now the correlation between the ranked queries is obtained by using Spearman's rank correlation (Wilks, 2011) measure ρ where

$$\rho = 1 - \frac{(6\sum D^2)}{n(n^2-1)} \tag{8}$$

Where $D = R1\text{-}R2$ and $-1 \leq \rho \leq 1$ when $\rho = -1$ the ranks are negatively correlated and $\rho = 1$ the ranks are positively correlated. Correlation value between the techniques is

$$\rho\ (preference, t-measure) = 0.5571$$

$$\rho\ (preference, combined\ measure) = 0.94285$$

$$\rho\ (t-measure, combined\ measure) = 0.72857$$

The ranking of queries produced by considering preference and preference with t-measure are approximately same because its rank correlation is 0.94285. Friedman test (Wilks, 2011) is applied between the ranking techniques to test the hypothesis

H_0: There is no difference between the ranking order and the results are significant

H_a: The ranking orders produced by the techniques are different and the decision rule is

Reject H_0 if $M \geq$ critical value at $\alpha = 5\%$

The differences between the sum of the ranks is evaluated by calculating the Friedman test statistic M from the formula

$$M = \frac{(12 \sum R_j * R_j)}{nk(k+1)} - 3n(k+1) \tag{9}$$

Where n = number of rows, k = number of columns and R_j = Sum of ranks in R_j.

Friedman Statistic value M for the ranking values in Table 7 is 57.5. Critical value of M for 6 rows and 3 columns at $\alpha = 5\%$ is 7.0.

$\therefore$ M > critical value of M, hence reject null hypothesis and accept Ha.

Ranking order of queries on Table 7 is compared based on the rank measures (Huang & Ling, 2005) Euclidean Distance, Manhattan Distance, Area under Curve, Ordered Area Under Curve and Accuracy. For a balanced ordered ranked list with *n* queries (half positive and half negative), actual ranked position is greater than *n/2* as a positive example; and the rest as negative. Table 8 depicts the position of the positive and negative queries. Table 9 shows the 6 rank measures for 3 techniques.

Table 8. Positive and Negative positions

Technique	Q1	Q2	Q3	Q4	Q5	Q6
Original	-	-	-	+	+	+
Preference	-	+	-	-	+	+
t-measure	-	-	-	+	+	+
Preference + t-measure	-	+	-	-	+	+

Table 9. Rank measures for 3 techniques

Rank Measures	Error rate	Accuracy	Euclidean Distance	Manhattan Distance	AUC	OAUC
Preference	33%	0.66	3.535	6	0.77	0.78
t-measure	0%	1	3.464	6	1	0.82
Preference + t-measure	33%	0.66	3.391	7	0.77	0.77

The technique t-measure ranks the queries in a best manner and its error rate is 0% and the accuracy value is 1 remaining techniques have equal error rate of 33% and accuracy is 0.66. Based on Error rate and accuracy, the techniques are ordered as

t-measure > Preference and Preference + t-measure

When the distance is considered, again t-measure has minimum distance compared with others. Based on AUC and OAUC, techniques are ordered as

t-measure > Preference + t-measure > Preference

Mean Reciprocal Rank (MRR) is calculated (Liu *et al.*, 2011b) for ranking order of queries given in Table 7,

$$MRR(technique) = (\sum_{i=1}^{q} \frac{1}{ri})/q \tag{10}$$

Where q = number of queries and r_i = rank of the i^{th} query

MRR (precision) = 0.4, MRR (t-measure) = 0.3944 and MRR (precision + t-measure) = 0.407 The mean reciprocal rank is high, when both the precision and t-measure are considered for ranking the queries.

Similarity measures

Next, the similarity between the users and queries are generated. The similarity between the queries in terms of keywords and URLs are calculated using Formula (11) and (12). This is similar to Jaccard Coefficient (Thada & Joshi, 2011)

$$Keyword\ Similarity(Qi, Qj) = \frac{keywords\ (Qi \cap Qj)}{keywords\ (Qi) + keywords\ (Qj)} \tag{11}$$

If the queries Qi and Qj share some common terms in their keywords then the queries are similar.

$$\text{URL Similarity } (\text{Qi}, \text{Qj}) = \frac{\text{count}(\text{URL}(\text{Qi}) \cap \text{URL}(\text{Qj}))}{\text{count}(\text{URL}(\text{Qi})) + \text{count}(\text{URL}(\text{Qj}))} \quad (12)$$

The function count is used to find the number of URLs clicked for the given query. The URL count is calculated by using the algorithm HASHURLCOUNT (Umagandhi & Senthilkumar, 2009). The combined similarity measure is calculated by using the Formula (13).

$$\textit{Combined Similarity}(Qi, Qj) = \alpha * \textit{Keyword Similarity}(Qi, Qj) + \beta * \textit{URL Similarity}(Qi, Qj) \quad (13)$$

The positive and negative concept similarities may also be considered when there is an attribute concept available in the query log file (Umagandhi & Senthilkumar, 2012).

Clustering of Users

The users who have similar intents are clustered. This cluster recommends the queries for the user from the access logs of similar users. For example, the day wise access of all the users may be considered. Table 10 contains the access information of 5 users with 6 queries. The value 1 indicates that the query *Qi* is accessed by the *user j* where $1 \leq i \leq 6$ and $1 \leq j \leq 5$.

Table 10. Query accessed by 5 users

User/ Query	Q1	Q2	Q3	Q4	Q5	Q6
U1	1	0	1	1	0	0
U2	1	0	1	1	0	0
U3	1	0	1	1	1	0
U4	0	0	1	1	0	0
U5	0	1	0	0	1	1

Next, the similarity matrix is generated between the users on day1 using the asymmetric binary similarity measure.

	*U*1	*U*2	*U*3	*U*4	*U*5
*U*1	–	1	0.75	0.67	0
*U*2	1	–	0.75	0.67	0
*U*3	0.75	0.75	–	0.5	0.17
*U*4	0.67	0.67	0.5	–	0
*U*5	0	0	0.17	0	–

The users are clustered based on average similarity. There are two resultant clusters: Cluster 1 contains {*U1, U2, U3, U4*} and Cluster 2 contains {*U5*}. Similarly, we have to find the user wise cluster for all the days, which is shown in Table 11.

Table 11. Day wise user cluster

Day	Clusters
Day 1	C1 = {U1, U2} C2 = {U3,U4} C3 = {U5}
Day 2	C1 = {U1, U2, U3} C2 = {U4} C3 = {U5}
Day 3	C1 = {U1} C2 = {U2,U3} C3 = {U4} C4 = {U5}
Day 4	C1 = {U1} C2 = {U2,U3} C3 = {U4,U5}
Day 5	C1 = {U1, U2,U3} C3 = {U4,U5}

PrefixSpanBasic (Umagandhi & Senthilkumar 2013) algorithm generated the frequently clustered users with minimum support 2.

	*U*1	*U*2	*U*3	*U*4	*U*5
*U*1	–	3	2	0	0
*U*2	3	–	4	0	0
*U*3	2	4	–	1	0
*U*4	0	0	1	–	2
*U*5	0	0	0	2	2

The users {*U1, U2, U3*} are in one cluster and they have similar query access and the users {*U4, U5*} are in another cluster. The queries are recommended as a Collaborative one; recommendation for the user *U1* is from the similar queries of the users *U2* and *U3*. Similarly for the user *U4*, the recommendation is from the user *U5*.

Clustering of Queries

Irrespective of users and time stamps, the queries are clustered based on the combined similarity measure given in formula (13). The favourite query of every query cluster is to be found by using the algorithm Favourite Query Finder. When the input query is encountered in the searching process, first the cluster should be found where the query belongs to and the favourite query of that cluster should be recommended.

EXPERIMENTAL RESULTS

The algorithms have been implemented in JDK 1.6.0_24. All the experiments have been performed in Intel Core i3 processor 2.53 GHz with Windows 7 Home Premium (64-bit) and 4 GB RAM. The proposed work has been evaluated by considering the experimental data from AOL search engine query log from 1-3-2006 to 31-5-2006 (zola.di.unipi.it /smalltext/ datasets. html). The dataset is stored in SQL Server, which contains 1975811 log entries and 19131507 words from ˜650k users in 174 MB over three months; based on system memory and its speed and the pre-processed log entries for the user 1038 is only considered. Figure 3 shows the analysis of the length of the query keywords. Totally 181 distinct queries out of 902 log entries are issued by the user 1038. This analysis shows that 80% of the queries are redundant and the users intents are same at some point. The user 1038 has clicked 318 distinct URLs for 181 distinct queries. Thus in average, the user clicked 2.28 pages per query and 69% of the query keywords contain less than 3 terms and the average query length is 2.56.

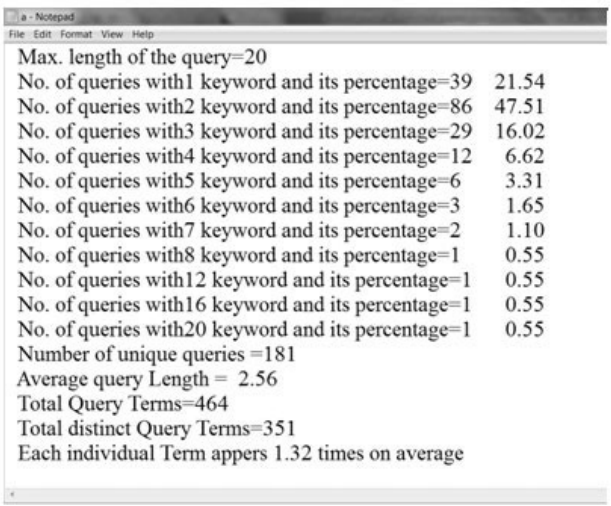

Fig. 3. Analysis of Query Length of the user 1038

The first 200 log entries contain the users with anonymous ID 227, 309, 366, 647, 706, 1038, 808 and 144. Table 12 shows number of access for every user, month wise. For example, 1038 has 458 accesses with URL click and 444 accesses with no clicks. Number of access in March is 423, April is 17, May is 18 and totally 458 queries are given by the user 1038. The user 1038 has accessed more compared with other users in the search log. The log entries for the user 1038 are analysed.

Table 12. Month wise Access

AID	Number of Access	Number of Access (ClickUrl also contains NoClick)	Number of Unique Queries	March	April	May
227	62	212	38	27	2	33
309	34	104	18	20	5	9
366	1	6	1	1	0	0
647	25	41	19	20	4	1
706	47	76	22	29	13	5
1038	458	902	181	423	17	18
808	7	26	5	1	0	6
144	5	21	4	1	3	1

To find the favourite query of the user 1038, the queries which were triggered more than 10 times are to be considered and the preference and t-measure should be calculated for all the queries which are shown in Table 13.

Table 13. Preference and t-measure

Q. Id.	Query	Support	Preference	t-measure	both
1	didier drogba	17	0.070539	0.0461	0.05832
2	How to take optygen	10	0.041494	0.1106	0.076047
3	Joe afful	71	0.294606	0.6762	0.485403
4	Liam George	18	0.074689	0.1321	0.103394
5	Low kupono	20	0.082988	0.0337	0.058344
6	Mzbel	11	0.045643	0.0523	0.048972
7	Omar jarun	12	0.049793	0.0184	0.034096
8	Optygen	19	0.078838	0.0830	0.080919
9	Optygen soccer	11	0.045643	0.0307	0.038172
10	Padraig drew	10	0.041494	0.04	0.040747
11	Samuel kuffour	10	0.041494	0.0184	0.029947
12	Sharlie joseph	20	0.082988	0.0676	0.075294
13	Shane mcfaul	12	0.049793	0.0030	0.026396

By analysing Table 13, the favourite query of the user 1038 is "Joe afful" and it is triggered 71 times i.e. 7.8% of total query. Spearmans rank correlation between the rank measures is

$\rho(preference, t-measure) = 0.4107$

$\rho(preference, combined\ measure) = 0.59478$

$\rho(t-measure, combined\ measure) = 0.95467$

The ranking of queries produced by considering t-measure and preference with t-measure are close, because its rank correlation is 0.95467. Figure 4 shows the preference, t-measure and preference with t-measure values for the top 13 queries given by the user 1038.

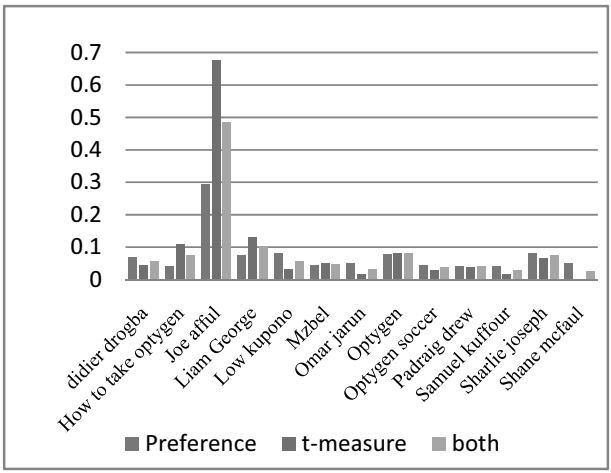

Fig. 4. Top 13 queries of 1038

Table 14 shows the ranking order of 13 queries issued by the user 1038 by considering preference, t-measure and both t-measure and preference. Figure 5 shows the changes in query ranking.

Table 14. Ranking of queries

Original	Preference	t-measure	Preference + t-measure
didier drogba	Joe afful	Joe afful	Joe afful
How to take optygen	Low kupono	Liam George	Liam George
Joe afful	Sharlie joseph	How to take optygen	Optygen
Liam George	Optygen	Optygen	How to take optygen
Low kupono	Liam George	Sharlie joseph	Sharlie joseph

Cont. Table 14. Ranking of queries

Original	Preference	t-measure	Preference + t-measure
Mzbel	didier drogba	Mzbel	Low kupono
Omar jarun	Omar jarun	didier drogba	didier drogba
Optygen	Shane mcfaul	Padraig drew	Mzbel
Optygen soccer	Mzbel	Low kupono	Padraig drew
Padraig drew	Optygen soccer	Optygen soccer	Optygen soccer
Samuel kuffour	How to take optygen	Omar jarun	Omar jarun
Sharlie joseph	Padraig drew	Samuel kuffour	Samuel kuffour
Shane mcfaul	Samuel kuffour	Shane mcfaul	Shane mcfaul

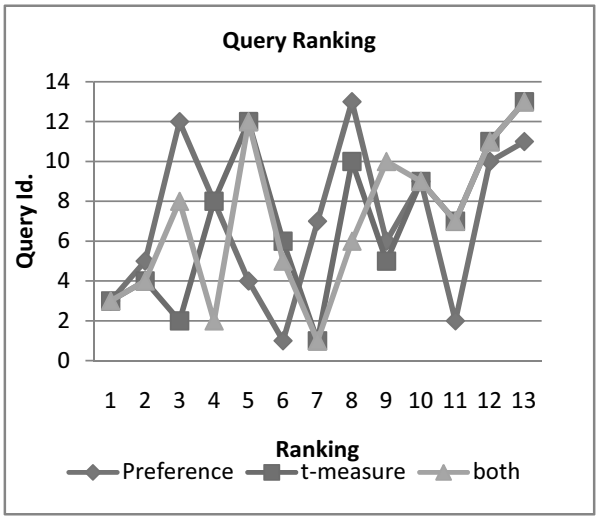

Fig. 5. Ranking of Queries

Next, the query cluster has to be generated for the queries issued by the user 1038. Table 15 shows some of the query cluster which is clustered based on the similarity measure and clustering process given in the Section 5. Favourite query in every query cluster is shown in bold characters. Finally the similar intent users are identified and clustered.

Table 15. Sample query clusters

Volkswagon	Ghana	Optygen	cover letter	sample letter
Vw	**ghanaweb**	bigsoccer.com	sample cover letters	**sample cover letters**
volkswagon credit	Mzbel	optygen soccer	after interview letter	sample thank you letters
volkswagon cars	news in Ghana	Optygen	**what to ask on an interview**	sample resume template
Volkswagen	ghana news	how to take optygen	questions to ask employer in interview	follow up letters
vw golf	landmarks part of ghana africa	mediotiempo.com	examples of skill resumes	thank you letters for business interviews
vw jetta	ghana names	mexican soccer gear	business clothes	cover letter for internship application
vw.com	ghana history	**bigsoccer.com**	interview letters	email cover letter samples
www.volkswagon.com	castro ghana	corithians soccer team	resume format	examples of skill resumes
volkswagon price	www.ghanaweb.com	shane mcfaul	career choice activities	interview letters

CONCLUSIONS

In this paper, the problem of recommending queries to better capture the search intents of search users has been investigated. The recommendation is given to different categories of users namely (i) user and query is new (ii) user is new but the query is existing (iii) user is exiting but the query is new and (iv) both user and query are existing. The recommendation strategy is based on the favourite query of the user, favourite and similar query from the query cluster and similar queries from the user cluster. The proposed work also generates the frequently occurred query patterns. Through this pattern, recommendation is given to the user. The recommended queries are ranked based on the preference, t-measure and preference with t-measure. Experimental results show that, both preference and t-measure are considered and the recommended queries are ranked well. In the near future, the ranking and clustering process may be combined in a unified algorithm.

REFERENCES

Baeza-Yates, R., Hurtado, C. & Mendoza, M. 2005. Query recommendation using query logs in search engines. In Current Trends in Database Technology-EDBT 2004 Workshops, Springer Berlin Heidelberg. 588-596.

Baraglia, R., Castillo, C., Donato, D., Nardini, F. M., Perego, R. & Silvestri, F. 2009. Aging effects on query flow graphs for query suggestion. In Proceedings

of the 18th ACM conference on Information and knowledge management, 1947-1950.

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

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

China Internet Network Information Center. 2009. CNNIC Search behavior survey report, http://research.cnnic.cn/html/1253600840d1370.html.

Chirita, P., Firan, C. S. & Nejdl, W. 2007. Personalized query expansion for the web. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval.7-14.

Cucerzan, S. & White, R. W. 2007. Query suggestion based on user landing pages. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. 875-876.

Golfarelli, M., Rizzi, S. & Biondi, P. 2011. myOLAP: An approach to express and evaluate OLAP preferences. IEEE Transactions on Knowledge and Data Engineering. **23**(7):1050-1064.

Han, J. & Kamber, M. 2006. Data mining concepts and techniques. Second Edition. Elsevier.

Huang, J. & Ling, C. X. 2005. Rank measures for ordering. in knowledge discovery in databases. Springer Berlin Heidelberg. 503-510.

Joachims, T., Granka, L., Pan, B., Hembrooke, H. & 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. 154-161.

Khemiri, R. & Bentayeb, F. 2012. Interactive query recommendation assistant. IEEE 23rd International Workshop on Database and Expert Systems Applications (DEXA). 93-97.

Khemiri, R. & Bentayeb, F. 2013. FIMIOQR: Frequent item sets mining for interactive olap query recommendation. In DBKDA, the Fifth International Conference on Advances in Databases, Knowledge & Data Applications. 9-14.

Khoussainova, N., Kwon, Y., Balazinska, M. & Suciu, D. 2010. SnipSuggest: context-aware autocompletion for SQL. Proceedings of the VLDB Endowment. **4**(1):22-33.

Li, R., Kao, B., Bi, B., Cheng, R. & Lo., E. 2012. DQR: a probabilistic approach to diversified query recommendation. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management. 16-25.

Liu, Y., Miao, J., Zhang M., Ma, S. & Ru, L. 2011a. How do users describe their information need: Query recommendation based on snippet click model. Expert Systems with Applications. **38**(11):13847-13856.

Liu, Y., Ni, X., Sun, J. T. & Chen, Z. 2011b. Unsupervised transactional query classification based on webpage form understanding. In Proceedings of the 20th ACM international conference on Information and knowledge management. 57-66.

Ma, H., Lyu, M. R. & King, I. 2010. Diversifying query suggestion results. In Proceedings of AAAI. 10.

Mei, Q., Zhou, D. & Church, K. 2008. Query suggestion using hitting time. In Proceedings of the 17th ACM conference on Information and knowledge management. 469-478.

Neelam, D. & Sharma, A. K. 2010. Rank optimization and query recommendation in search engines using web log mining techniques. Journal of Computing. **2**(12).

Silverstein, C., Marais, H., Henzinger, M. & Moricz, M. 1999. Analysis of a very large web search engine query log. In ACM SIGIR Forum. **33**(1): 6-12.

Stefanidis, K., Drosou, M. & Pitoura, E. 2009. You may also like results in relational databases. Proc. PersDB, Lyon, France.

Thada, M. V. & Joshi, M. S. 2011. A genetic algorithm approach for improving the average relevancy of retrieved documents using jaccard similarity coefficient. International Journal of Research in IT & Management. 4.

Umagandhi, R. & Senthilkumar, A. V. 2009. Approaches to find URL click count from Search Engine Query Logs. International Journal of Computer Information Systems. **4**(6): 30-36.

Umagandhi, R. & Senthilkumar, A. V. 2012. Concept based time independent query recommendations from search engine query logs. Proceedings of the International Conference on computer Applications and Advanced Communications, Sep 17-18, WARSE, Singapore.

Umagandhi R & Senthilkumar A V. 2013. Time dependent approach for query and url recommendations using search engine query logs. IAENG International Journal of Computer Science. **40**(3).

Wilks, D. S. 2011. Statistical methods in the atmospheric sciences.Vol. 100.

Submitted : *15/04/2013*
Revised : *29/09/2013*
Accepted : *06/10/2013*

منهج استدلالات الترتيب الوقتية لاقتراح استفهامات باستخدام سجلات استفهام محرك بحث

*** ر. اوماغاندي و **أ. ف. سنثيل كومار**

*استاذ مساعد ورئيس قسم تكنولوجيا الحاسوب - كلية كونغونادو للعلوم والاداب
**مدير (MCA) - كلية هندوستان للعلوم والآداب - كويمباتور

خلاصة

من الواضح ان استفهامات البحث على الوب التي يطرحها المستخدم عادة ما تكون مختصرة ومحيرة، كما ان معظم الاستفهامات القصيرة لا تمثل حاجة المستخدم الحقيقية للمعلومات ولا تقدم النتائج المرضية. إن مقترح الاستفهام هو طريقة تعتمد على النية الحقيقية للمستخدم في طرح استفهامات بديلة، وذلك بهدف تأطير الاستفهامات بالمستقبل. يقترح العمل المقدم استفهامات لأربعة انواع من المستخدمين من خلال ثلاثة طرق هي: (1) يتم التعرف على واقتراح الاستفهامات المفضلة لدى المستخدم. (2) يصنف المستخدمون الذين لهم نفس النوايا في مجموعة ويقدم اقتراح شامل وفق سجلات الوصول للمستخدمين المتشابهين. (3) تصنف الاستفهامات المتشابهة ويتم اقتراح افضل استفهام يمثل بقية الاستفهامات ويشملها. كما يرتب العمل المقدم الاستفهامات المقترحة وفق افضلية وقت الوصول للاستفهام. هذا وسيتم تقييم الطريقة المقترحة من خلال تجارب تجرى على سجل استفهام محرك بحث يعمل بالوقت الحقيقي.

كلمات مفتاحية: الاستفهام المفضل، الافضليات، فحص قياس- ت، نمط الاستفهام المتكرر، سجل الاستفهام.

المجلة التربوية

مجلة فصلية، تخصصية، محكّمة
تصدر عن مجلس النشر العلمي - جامعة الكويت

رئيس التحرير: أ. د. عبدالله محمد الشيخ

تُنشر:

- البحوث التربوية المحكمة
- مراجعات الكتب التربوية الحديثة
- محاضر الحوار التربوي
- التقارير عن المؤتمرات التربوية وملخصات الرسائل الجامعية

- تقبل البحوث باللغتين العربية والإنجليزية.
- تنشر لأساتذة التربية والمختصين بها من مختلف الأقطار العربية والدول الأجنبية.

الاشتراكات:

في الكويت: ثلاثة دنانير للأفراد، وخمسة عشر ديناراً للمؤسسات.
في الدول العربية: أربعة دنانير للأفراد، وخمسة عشر ديناراً للمؤسسات.
في الدول الأجنبية: خمسة عشر دولاراً للأفراد، وستون دولاراً للمؤسسات.

توجه جميع المراسلات إلى:

رئيس تحرير المجلة التربوية - مجلس النشر العلمي ص.ب. ١٣٤١١ كيفان - الرمز البريدي 71955
الكويت هاتف: ٢٤٨٤٦٨٤٣ (داخلي ٤٤٠٣ - ٤٤٠٩) - مباشر: ٢٤٨٤٧٩٦١ - فاكس: ٢٤٨٣٧٧٩٤
E-mail: joe@ku.edu.kw