International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-4, Issue-1 E-ISSN: 2347-2693

Search Engine Query Grouping using the combination of Time, Text and URL Similarity with Association Rules

Divakar Pandey

Department of Computer Science and Engineering,
(SATI) Vidisha, MP, India

www.ijcseonline.org

Received: Dec/19/2015 Revised: Dec/24/2015 Accepted: Jan/17/2016 Published: Jan/30/ 2016

Abstract— Understanding the characteristics of queries wherever a search engine is failing is very important for improving search engine performance. Previous work for the most part depends on user-interaction options (e.g., click through statistics) to spot such underperforming queries. This paper evaluates the techniques used for users log history query grouping in automatic manner. Automatic query grouping is very useful for lots of software and web application. In this paper we proposes new method for calculating similarity between query using various log record attributes like time, clicked url, text similarity and frequently occurring queries using association rules. This work introduces another strong method for similar query grouping to make web browsing easy and efficient by query recommendation. A comparative evaluation of proposed method with existing work available in literature has also been carried out and the result shows that the proposed method is more effective.

Keywords: : Query reformulation, click graph, web mining, association rules, text similarity.

Introduction

Automatic detection of less related queries for search engines could not perform better and users not able to get appropriate results,such problem are addressed by different authors in their research literature [1][2][3][4]. In Internet surfing environments, massive volumes of query logs area unit promptly come-at-able. This makes it cheap to gather several coaching examples that may be accustomed improve classification performance, improves the capability, find desire related queries from any search engines. So such types of mechanism have to select query from a large possibilities of queries. Some of systems follow ranking parameter for select appropriate query. Such ranking system applied by training given set of sampled query, hence the power of ranking system is limited to that set and therefore ranking algorithm unable to perform effectively.

Earlier effort done on predicting less effective queries has mainly focused user communications (e.g., clickthrough information) [3][4] and collective properties from user activities within query logs and sensors [2]. Even though earlier schemes are pretty doing well, they are not as much helpful from the search engine point of view. The cause is dissatisfaction indications are simply presented a posteriori. This is because of search engines hardly ever comprising the opportunity to hamper and treat the user disappointed results. On the other hand, query effective predictors interaction actions signals and be able to facilitate identify disappointment earlier than the search session ends.

Grouping of queries have concerned important concentration in recent years. many program application softwares use query suggestion. Query agglomeration is a requirement to control properly. Certainly, agglomeration is crucial to release verity value of query logs. Still, agglomeration look for queries successfully are quite tough, due to the high variety and uneven input through users. Search queries are habitually small and unclear in the point of view user wants. Many different queries would possibly talk to one construct, where one query would probably contains many ideas. Some of researchers already introduced agglomeration ways, such as K-Means or DBSCAN unable to guarantee smart results in such a numerous environment. Grouped agglomeration recommends smart results but is they are time consuming relatively process. This paper gives a singular agglomeration scheme maintained a key insight – program results might themselves be used to confirm query relationship. Proposed work suggests query subject similarity and time comparison for better technique that utilize search query logs properly. This query log information may be used to develop quite cost-effective and proper equation for agglomerated queries. It will helpful for various search engines working for user better searching and document searching software's like query recommendation, output ranking, query modification, sessionization, and supportive search.

I. Problem Formulation

This section gives details about various problems have to face while query clustering by user's search history data given in query log. Organize user's query in automatic and efficient manner is tough task. Arrange the query clusters at period of a user's history is hard due to get solution of several reasons. Primary, associated queries won't appear close to one another, as a look like similarly within a duration days or maybe weeks. Usually getting information for the interleaving of pairs for queries and respective clicks from overall log records are totally different search tasks because of user's multitasking behavior [5], simultaneously

working on multiple browser tabs, and infrequently changing search topics.

Time	Query
10:51:48	saturn vue
10:52:24	hybrid saturn vue
10:59:28	snorkeling
11:12:04	barbados hotel
11:17:23	sprint slider phone
11:21:02	toys r us wii
11:40:27	best buy wii console
12:32:42	finance statement
12:22:22	wii gamestop

Time	Query
12:59:12	saturn dealers
13:03:34	saturn hybrid review
16:34:09	bank of america
17:52:49	caribean cruise
19:22:13	gamestop discount
19:25:49	used games wii
19:50:12	tripadvisor barbados
20:11:56	expedia
20:44:01	sprint latest model cell phones

a) User's search history

Group 1	Group 2
saturn vue hybrid saturn vue saturn dealers saturn hybrid review	snorkeling barbados hotel caribean cruise tripadvisor barbados expedia
Group 3	**Group 5**
sprint slider phone sprint latest model cell phones	toys r us wii best buy wii console wii gamestop gamestop discount used games wii
Group 4	
finance statement bank of america	

b) Query Collection

Fig. 1. Browsing history log record of a user on internet surfing within a day can be arranged as query groups.

Figure 1(a) shows the related queries "hybrid saturn vue" along with "Saturn dealers" could be divided from other searched dissimilar queries. This restricts the usefulness of methods which uses time or order to categorize similar queries. Second, connected queries won't be textually similar. as associate in case, in Fig. 1(b),the similar queries "tripadvisor barbados" along with "Caribbean cruise" into group a combination of words in common. Therefore,

relying completely on string similarity is to boot deficient. Finally, as users may also manually alter their varied query groups, any automatic query grouping possesses to respect the manual efforts or edits by the users.

Every query cluster could also be a group of queries by an identical user that measures query relevant to each different around normal data he desires. These query groups are dynamically updated according to outcomes from user launches new search words and respective new query collections are formed according to similarity in time. This paper mainly aims to improve these query collection known as groups as users activities perform while searching records over internet within any day. This activity information may be very useful to create groups of similar queries as given in figure 1(b). The first collection of query are all queries which are similar to Saturn vehicles in any form. Other group mention in figure is Barbados hotel similar queries within its collection.

II. Association Rule Mining Queries

Assume that items could be a finite set of symbols denoted by capital letters, e.g., Items={A,B,C…}. A transactional info could be a assortment of rows wherever every row could be a set of things. An itemset could be a set of items. A row r supports and itemset S if S⊆ r. The support (denoted support(S)) of an itemset S is that the multi-set of all rows of the info that support S. The frequency of an itemset S is |support(S)|/|support(∅)| and is denoted F (S). We regularly use a string notation for itemsets, e.g., AB for {A,B} . An association rule is denoted X⇒Y where Y≠ ∅, X∩Y= ∅, X⊆ items is the body of the rule and Y⊆ items is the head of the rule. The support and frequency of a rule are defined as the support and frequency of an itemset X∪Y. A row r supports a rule X⇒Y, if it supports X∪Y. A row r is an exception for a rule X⇒Y, if it supports X and it does not support Y. The confidence of the rule is conf(X⇒Y)= *F*(X⇒Y)/ *F(X)*= *F(X*∪Y)/ *F(X).* The confidence of the rule gives the conditional probability that a transaction supports *X* ∪*Y,* when it supports *X*. A rule with a confidence of one has no exception and is called a logical rule.

III. Query Similarities for Search Logs

The idea is to initially get frequent item sets for every query victimization existing association rule mining algorithms either by horizontal or vertical approach. Once we discover frequent query itemsets in log dataset then we have a tendency to develop the machinery to stipulate the query association supported web search logs. Our live of association is meshed toward capturing two important needed parameters of similar queries, known as

1) Queries that often appear on as reformulations and

2) Queries which are input by users in search engine always click on similar other pages.

This section gives detail about introduction of different types of four search behavior graphs that capture similar properties. Following that, we've got a bent to indicate but we'll use these graphs to cipher query association and also the approach we'll incorporate the clicks following a user's query therefore on reinforce our association metric

A. Search Behavior Graphs

In this work we uses some special types of graphs can be generated using users search log records. The Query Reformulation Graph (QRG),represents the association between a mix of queries that unit of measurement likely reformulations of each totally different. The Query Click Graph (QCG), represents the association between two queries that often cause clicks on similar URLs. The Text Similarity Graph (TSG), where we assume that two query teams area unit similar if queries are similar interm of text appear within these query.The Query Association Graph (QAG), that relies on the standard to a couple of query issued in succession among the search logs unit of measurement strongly associated. All four graphs unit of measurement made more than the equivalent collection of graph vertices VQ, having queries that appear during a minimum of 1 among the graphs, but their edges unit of measurement made public otherwise.

1) Query Reformulation similarity

One way to identify relevant queries is to suppose query reformulations that unit of measurement sometimes found within the query logs of a probe engine. If two queries that unit of measurement issued consecutively by many users occur multiple times, such queries are needed to be reformation with each other totally different. To measure the association between any two queries searched by user, the time similarity metric can help for utilize time stamp information between launching queries within user search log record as time similarity. The proposed work make use of time similarity information, if two queries search simultaneously within very less time interval means these query must be similar in any respect. The math frequency there upon two different queries appear next to each, totally different at intervals the whole search log record for all users exists in web log.

$$w_r(q_i, q_j) = \frac{count_r(q_i, q_j)}{\sum_{(q_i,q_k)\in s_{QR}} count(q_i q_k)}$$

2) Query Click similarity

A different thanks to capture relevant queries from the search logs is to contemplate queries that area unit possible to induce users to click often on an equivalent set of URLs. for instance, though the queries "ipod" and "apple store" don't share any text or seem temporally march on a user's search history, they're relevant as a result of their possible to possess resulted in clicks regarding the ipod product.

So as to get this type of similarity, measure of likely similar queries, there is need to construct a graph known as the Query Click Graph (QCG).We have a tendency to first begin (VU,EC), by considering this graph is bipartite graph consists of click information with its URL from user CG=(VQ)[6]. CG consists two distinct collection of nodes can be similar queries VQ, and its connected URLs, VU can be separated from the web search logs. There's a position EC, if (qi, uk) query was issued and address Great Britain was clicked by some users. This method assign weight every edge (qi, Great Britain) by the amount of your time query qi is searched followed by uk URL'S clicked then information can be stored as count(qi,uk). Hence to separate out rare pairs employing a threshold Tc. During this means, victimisation the CG, need to establish pairs of search word or query result to click on same URL'S. This graph derives our query click graph, QCG= (Vq, Eqc), wherever the vertices represets the query from log records and edges can be represents as qi to qj direct link of these two queries a minimum of one address, uk. Hence cost of edge (qi,qj) of QCG graph can be givea as wc(qi,qj) according to the weighted uneven Jaccard similarity [7] as follows:

$$w_c(q_i, q_j) = \frac{\sum_{u_k} min(count_c(q_i, u_k), count_c(q_j, u_k))}{\sum_{u_k} count_c(q_j, u_k)}$$

This captures the intuition that qj is additional associated with qi, if additional of qi's clicks associate to link URL are clicked for qj.

3) Text Similarity Graph

On a special note, we have a tendency to might assume that two query teams area unit similar, if queries are similar in term of text appear within these query. The common text words are normally occurs within two different queries. we will so outline the subsequent two text primarily based cosine values that can be employed in place of sim. Simtext(sc,si) is outlined because numbers of common word appears within qc and qj as follows:

$$(S_c, S_i) = \frac{|words(q_c) \cap words(q_i)|}{|words(q_c) \cup words(q_i)|}$$

4) Query Association Similarity

This technique relies on the standard to a couple of query issued in succession among the search logs unit of measurement strongly associated. A. Fuxman, P. Tsaparas, K. Achan,and R. Agrawal presented a technique to primarily reorganize a series of user web log queries to cluster similar queries on by measuring an illustration of the association rules.

Sim_AS (s_c,s_i) is defind as the number of times two queries q_c and q_i appear in succession in the search log over the no.of times q_c appears.We calculate the Query Association Similarity as follows:

$$Sim_{AS}(S_c, S_i) = \frac{(freq(q_c, q_i))}{(freq(q_c))}$$

B. Final Query Similarity

The Final Query Similarity (FQS) calculated by reformation similarity, query click similarity, text similarity and query association similarity as follows:

$$w_f(q_i, q_j) = \alpha \times w_r(q_i, q_j) + \beta \times w_c(q_i, q_j) + \gamma \times w_{ts}(q_i, q_j) + \delta \times w_{as}(q_i, q_j)$$

Here the α, β, γ and δ are four constants and we take it's value is equal to 0.25. The relative contribution of the four weights is controlled by α, β, γ and δ. We denote a query final similarity constructed with a particular value of α, β, γ and δ such as α+β+γ+δ=1 for FQS.

C. Proposed Algorithm

Fig 2 colaborates our proposed work flowchart as well as overall proposed algorithm in sequential manner and given as follows:

Algorithm for Search Engine Query Clustering Using SOM

Input: The queries Dataset containing the current query id, time, rank and clicked urls.

A set of existing query groups S={s1,s2,……..sm}

A similarity threshold 0<th<1

Output: The query group s that best matches S, or a new one if necessary.

Step1: Set the initial parameters

Let the users, S={s1,s2,s3,…….sn}

Current query and clicks, {qc,clk}

Weight of query reformation graph=Wr

Weight of query click graph=Wc

Weight of text similarity graph=Wtxt

Weight of query association graph=Was

α=0.25,β=0.25,γ=0.25,δ=0.25

Step2: Load query log dataset.

Step3: Feature extraction of time, query click, URLs etc.

Simtime(sc,si) =1/time(qc)-time(qi)

Step4: Creation of query reformation graph.

Weight is calculated as

$$w_r(q_i, q_j) - \frac{cou}{\sum_{(q_i,q_k)\in s_i}}$$

Step5: Creation of query click graph by click url

Weight is calculated as

$$w_c(q_i, q_j) = \frac{\sum_{u_k} min\,(count_c(q_i, u_k), count_c(q_j, u_k))}{\sum_{u_k} count_c(q_j, u_k)}$$

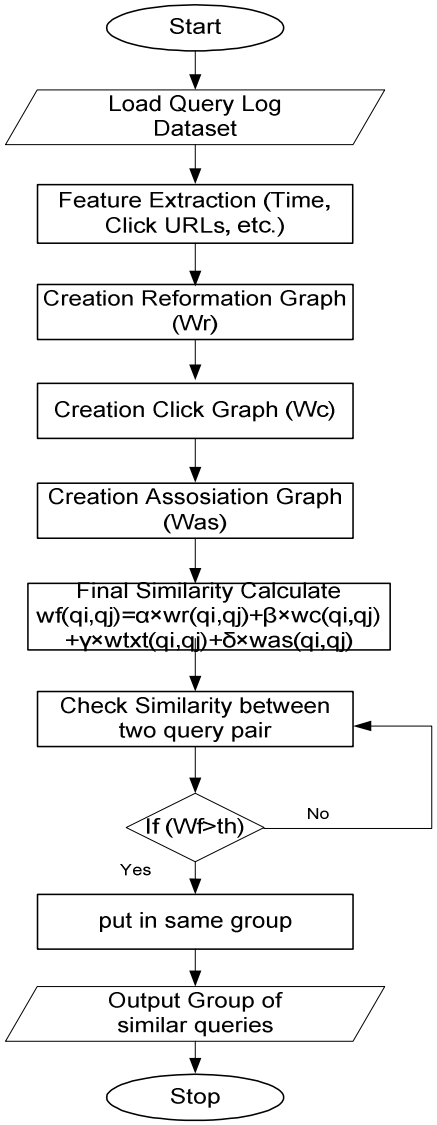

Fig 2. Flow diagram of Proposed Work

Step6: Creation of text similarity graph

Weight is calculated as

$$W_{txt}(q_i, q_j) = \frac{|words(q_c) \cap words(q_j)|}{|words(q_c) \cup words(q_j)|}$$

Step7: Calculate the weight for association similarity

$$W_{as}(S_c, S_i) = \frac{(freq(q_c, q_i))}{(freq(q_c))}$$

Step8: Calculate the weight for final similarity

$$w_f(q_i, q_j) = \alpha \times w_r(q_i, q_j) + \beta \times w_c(q_i, q_j) + \gamma \times w_{txt}(q_i, q_j) + \delta \times w_{as}(q_i, q_j)$$

Step9: This similarity matrix passed to SOM tool for k-means clustering. Clustering method results output similar query groups as one cluster.

Step10: Get the final groups accuracy and comparison.

IV. EXPERIMENTAL RESULTS

To compute the effectiveness of the proposed clustering method, with respect to real world user, here queries are clustered by measuring query pairs similarity and then labeled and generates resultant clustering according to SOM clustering. Any two queries may belongs to same cluster if they have similarity more than predefined threshold value, at the starting level all query have zero similarity with compare to any other query. To calculate the importance of proposed method alongside the clusters are produced by using the labelers assign according to similarity, We'll use the Rand Index [9] metric, that would be a commonly used live of similarity between two partitions. The Rand Index similarity between two partitions X,Y of n elements each is made public as , where a is that the vary of pairs that area inside constant set in X and conjointly stable collection within Y, other hand b with the intention of the vary of pairs that area in many sets in x and in many collections within Y. Greater RandIndex points denote better potential of grouping for similar queries while a precise mathematical explanation. The proposed method to improve the primary effective performance on Rand200 supported the RandIndex metric. This work tends to follow a similar approach for the baselines that tend to enforced conjointly. We'll to boot appraise the approaches on any take a glance at sets (Lo100, Me100, and Hi100). To confirm practical implementation of user log record checks the outdegree of user query output of simple counts among the outgoing links (average weight) inside the query reformulation graph. Therefore on check the results of usage information on the performance of our algorithms, to created three any take a glance at sets of 100 users each. The sets we have a tendency to tend molding to boot manually labeled as we have a tendency to painted. The first query collection dataset Lo100 having the web search activity of 100 users, for normal out degree less than 5. Similarly, Me100 contains user activity for users having out-degree greater than 5 but less than 10, at last Hi100 dataset holding out-degree more than 10.

As seen from table-1 our proposed method gives the better results as compared to the previous existing methods.It gives the highest value on the dataset Lo100 and the accuracy is about 97%. It gives the second higher value on the dataset Hi100 and the accuracy is about 91%.Similarly third highest value on Rand200(90%) and last one is the Me100(88%). So we concludes from table-1 that our method gives the more accurate and better results as compared to previous one.The results shows the effectiveness of our method.

Table 1. Comparative Performance of our Method

	Jaccard	CoR	ATSP	QFG	**Proposed**
Rand200	0.750	0.807	0.831	0.860	**0.903**
Lo100	0.762	0.794	0.832	0.821	**0.976**
Me100	0.748	0.802	0.857	0.868	**0.880**
Hi100	0.742	0.809	0.871	0.882	**0.912**

Figures 3 demonstrates outcome evaluation of performance metric for our dataset with base algorithms and it confirms that proposed scheme produced more accuracy than previous method. Proposed scheme takes advantages of Query Reformation Graph (QRG), Query click graph (QCG), Text Similarity graph (TSG) and Query Association Graph (QAG) between queries, which helps to generates better query clustering. The proposed scheme moreover measure up to with dataset Rand100, low100, me100 and hi100 dataset make a case for on top of for all dataset, proposed scheme gives high performance index for clustering queries as compared to other previous methods.

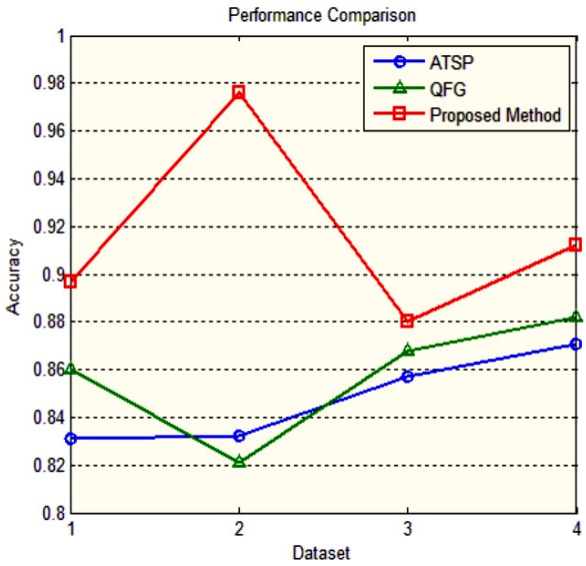

Fig 3. Comparative Performance (RandIndex) of Our Methods

V. CONCLUSION

We have given associate approach for automatic generation of query grouping of periodic query search patterns from user internet usage logs that are semantically enriched with data and resource topics. Over time, the mental object will capture each shopper internet access behavior and search log history of the net resources on the user. we have a tendency to designed a system that makes user profiles supported implicitly collected data, particularly the queries input given by user for search results. The projected framework mechanically generates association rules to spot underperforming queries. so as to create effective rules, initial generate topical attributes recognized from query text and formulate association rules that discover frequent patterns of the attributes for distinctive discontent queries. Then, apply a call tree learning to spot discriminative

keywords of discontent queries and mix the keywords through the association similarity strength between queries. The results are generated by simulations verified the efficiency of proposed system within the task of discontent query clustering compared to already presented query recommendation calculation. In view of previous work on representation user satisfaction, the benefit of specified methodology is distinct query time and proposed system will give proof of discontent to look engines before users abandon searches. Future work plans to plot effective strategies to enhance the instances of search discontent known by this methodology.

REFERENCES

[1] Ageev, M., Guo, Q., Lagun, D., and Agichtein, E. "Find it if you can: a game for modeling different types of web search success using interaction data". SIGIR, pp-**345 –354 , 2011**.

[2] Feild, H., Allan, J., and Jones, R. Predicting searcher frustration. SIGIR,pp- **34–41, 2010**.

[3] Guo, Q., White, R.W., Zhang, Y., Anderson, B., and Dumais, S.T. Why searchers switch: understanding and pre-dicting engine switching rationales. SIGIR, pp-**335–344, 2011**.

[4] Hassan, A., Song, Y., and He, L. A task level user satisfaction model and its application on improving relevance estimation. CIKM,pp- **125–134,2011**.

[5] Spink, M. Park, B.J. Jansen, and J. Pedersen,"Multitasking during Web Search Sessions" Information Processing and Management, vol. 42, no. 1, pp. **264-275, 2006**.

[6] Fuxman, P. Tsaparas, K. Achan, and R. Agrawal, "Using the Wisdom of the Crowds for Keyword Generation" Proc. the 17th Int'l Conf. World Wide Web (WWW '08), **2008.**

[7] Heasoo Hwang, Hady W. Lauw, Lise Getoor, and Alexandros Ntoulas, "Organizing User Search Histories", IEEE Transactions On Knowledge And Data Engineering, Vol. 24, NO. 5, Page **912-925.2012.**

[8] A. Fuxman, P. Tsaparas, K. Achan, and R. Agrawal, "Using the Wisdom of the Crowds for Keyword Generation" Proc. the 17th Int'l Conf. World Wide Web (WWW '08),**2008**.

[9] W.M. Rand, "Objective Criteria for the Evaluation of Clustering Methods" J. the Am. Statistical Assoc.,vol. 66, no. 336, pp. **846-850, 1971**.

[10] Jaideep Srivastava, Robert Cooley, Mukund Deshpande Pang-Ning Tan, "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data", SIGKDD Explorations, Vol. 1, No. 2, **2000.**

[11] Spink, M. Park, B.J. Jansen, and J. Pedersen, "Multitasking during Web Search sessions," Information Processing and Management, vol. 42, no. 1, pp. **264-275, 2006**.

[12] H.C. Ozmutlu and F. C¸ avdur, "Application of Automatic Topic Identification on Excite Web Search Engine Data Logs," Information Processing and Management, vol. 41, no. 5**,** pp. **1243-1262, 2005**

[13] F. Radlinski and T. Joachims, "Query Chains: Learning to Rank from Implicit Feedback," Proc. ACM Conf. Knowledge Discovery and Data Mining (KDD), **2005.**

[14] J. Yi and F. Maghoul, "Query Clustering Using Click-through Graph," Proc. the 18th Int'l Conf. World Wide Web (WWW '09), **2009.**

[15] E. Sadikov, J. Madhavan, L. Wang, and A. Halevy, "Clustering Query Refinements by User Intent," Proc. the 19th Int'l Conf. World Wide Web (WWW '10**), 2010.**

[16] R. Baeza-Yates and A. Tiberi, "Extracting Semantic Relations from Query Logs," Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), **2007.**

[17] M. Spiliopoulou, C. Pohle, and L.C. F aulstich. Improving the effectiveness of a website with web usage mining. In Advances in Web Usage Analysis and User Profiling, Berlin, Springer, pp. **141-62, 2000**

[18] K. Collins-Thompson and J. Callan, "Query Expansion Using Random Walk Models," Proc. 14th ACM Int'l Conf. Information and Knowledge Management (CIKM), **2005.**

[19] N. Craswell and M. Szummer, "Random Walks on the Click Graph," Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '07)**, 2007**.

[20] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang, "Query Clustering Using User Logs," ACM Trans. in Information Systems, vol. 20, no. 1, pp. **59-81,2002.**

[21] Tahira Tabassum, Amit Dubey, "User Search Query Grouping using Association Fusion Graph",International Journal of Advanced Research in Computer Science and Software Engineering, Volume4, ,Issue4, Page **259-267**, April **2014.**