Volume 4, Issue 4, April 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

User Search Query Grouping using Association Fusion Graph

Tahira Tabassum*
M. Tech Scholar (Computer Science & Engg. Dept)
Oriental College Of Technology (RGTU)
Bhopal , M.P. ,India

Amit Dubey
A.P. (Computer Science &Engg. Dept)
Oriental College Of Technology(RGTU)
Bhopal , M.P. , India

Abstract— As the size of net will increase alongside range of users, it's noticeably essential for the web site house owners perceive their customers so they will give better service, and conjointly enhance the standard of the web site. To attain this they rely upon the net access log files. Users' accesses are recorded in net logs. Thanks to the tremendous usage of net, the net log files are growing at a quicker rate and therefore the size is turning into huge. The net access log files are well-mined to extract fascinating pattern so the user behaviour is understood. To raise support users in their long-run data quests on the net, search engines keep track of their queries and clicks whereas looking on-line. During this paper, we have a tendency to study the matter of organizing a user's historical queries into groups in an exceedingly dynamic and automatic fashion. Mechanically distinguishing query groups is useful for variety of various program elements and applications, like query suggestions, result ranking, query alterations, sessionization, and collaborative search. In our approach, we have a tendency to transcend approaches that have faith in matter similarity or time thresholds, and that we propose an additional strong approach that leverages search query logs victimization association rule. We study through an experimental study, the performance of various techniques, and showcase their potential, particularly once combined along.

Keywords— Query reformulation, click graph, web mining, pattern analysis, Association Rule graph.

I. INTRODUCTION

In this world of data Technology, accessing info is that the most frequent task. Each day we've got to travel through many reasonably info that we'd like and what we have a tendency to do? simply browse the online and get the desired info on one click. Today, net is taking part in such a significant role in our daily life that it's terribly troublesome to survive while not it. The World Wide Web (WWW) has influenced both users (visitors) and website house owners. The online web site owners able to reach to all the targeted audience nationwide and internationally. They serve their client 24X7. On the opposite facet visitors also availing those facilities [1].

The ability to spot underperforming queries is particularly vital to net search engines. Since those systems got to cowl a broad vary of numerous queries and since ranking algorithms area unit generally trained employing a single sampled information set, there'll be queries on that the ranking algorithms cannot execute effectively.

Primary means that of accessing info on-line continues to be through keyword queries to a search engine. a posh task like travel arrangement must be broken down into variety of codependent steps over a amount of your time. as an example, a user might 1st search on potential destinations, timeline, events, etc. once deciding once and wherever to travel, the user might then hunt for the foremost appropriate arrangements for air tickets, rental cars, lodging, meals, etc. every step needs one or additional queries, and every query leads to one or additional clicks on relevant pages.

Web Usage Mining (WUM), are obtained from server logs, browser logs, proxy logs, or collected from an organization's Database. These information collections vary in terms of the situation of the information supply, the forms of information out there, the section of population from that the information was obtained, and techniques of implementation [1]. WUM could be a division of net Mining, which, consecutive, could be an element of knowledge Mining. The method of mining important and valuable data from large information is named data processing. WUM mines the usage options of the users of net Applications. This obtained information will then be applied in a very varied ways that like, checking of pretend parts etc [2]. WUM is taken into account as a element of the Business Intelligence in a company [3]. It's applied for deciding business approaches via the competent use of net Applications. it's terribly very important for the client Relationship Management (CRM) since it will guarantee client fulfilment until the interface between the client and therefore the organization thinks about [4].

There are several forms of information that may be employed in net Mining.

1. Content: The visible information within the sites or the information that was meant to be provided to the users. This greatly includes text and graphics (images).

2. Structure: The organization of the web site is illustrated by this information. it's partitioned off into 2 classes. Intra-page structure information incorporates the arrangement of many Hyper Text Markup Language (HTML) or Extended Markup Language (XML) tags inside a given page. The key variety of inter-page structure data is that the hyper-links used for web site navigation [13].

3. Usage: information that illustrates the usage patterns of sites, like science addresses, page references and therefore the date and time of accesses and alternative data supported the log format [4].

User profiles, descriptions of user interests, can be used by search engines to provide personalized search results. Many approaches to creating user profiles collect user information through proxy servers (to capture browsing histories) or desktop bots (to capture activities on a personal computer). Both these techniques require participation of the user to install the proxy server or the bot. In this paper, we explore the use of a less-invasive means of gathering user information for personalized search. In particular, we build user profiles based on user search history at the search site itself and study the use of these profiles to provide personalized search results. By implementing a association rules between searched queries by search engine, we were able to collect information about individual user search activities.

This is because users now pursue much broader informational and task-oriented goals such as arranging for future travel, managing their finances, or planning their purchase decisions. How-ever, the primary means of accessing information online is still through keyword queries to a search engine. A complex task such as travel arrangement has to be broken down into a number of co-dependent steps over a period of time. For instance, a user may first search on possible destinations, timeline, events, etc. After deciding when and where to go, the user may then search for the most suitable arrangements for air tickets, rental cars, lodging, meals, etc. Each step requires one or more queries, and each query results in one or more clicks on relevant pages. One important step toward enabling services and features that can help users during their complex search quests online is the capability to identify and group related queries together. Recently, some of the major search engines have introduced a new "Search History" feature, which allows users to track their online searches by recording their queries and clicks.

A user profile that represents the interests of a specific user can be used to supplement information about the search that, currently, is represented only by the query itself. This information could be used to narrow down the number of topics considered when retrieving the results, increasing the likelihood of including the most interesting results from the user's perspective. For the user in our example, if we knew that she had a strong interest in photography but little or none in religion, the photography-related results could be preferentially presented to the user.

One vital step toward sanctioning services and options that may facilitate users throughout their complicated search quests on-line is that the capability to spot and group connected queries along. Recently, a number of the most important search engines have introduced a replacement "Search History" feature that permits users to trace their on-line searches by recording their queries and clicks. for instance, Fig. one illustrates some of a user's history because it is shown by the Bing computer program on Gregorian calendar month of 2010. This history includes a sequence of 4 queries displayed in reverse written record order beside their corresponding clicks. Additionally to viewing their search history, users will manipulate it by manually written material and organizing connected queries and clicks into groups, or by sharing them with their friends. Whereas these options are useful, the manual efforts concerned are often unquiet and can be unreasonable because the search history gets longer over time. In fact, distinguishing groups of connected queries has applications on the far side serving to the users to create sense and keep track of queries and clicks in their search history. 1st and foremost, query grouping permits the computer program to raised perceive a user's session and doubtless tailor that user's search expertise in step with her wants. Once query groups are known, search engines will have a decent illustration of the search context behind the present query exploitation queries and clicks within the corresponding query cluster. this can facilitate to boost the standard of key parts of search engines like query suggestions, result ranking, query alterations, sessionization, and cooperative search. for instance, if a research engine is aware of that a current query "financial statement" belongs to a query cluster, it will boost the rank of the page that has data concerning a way to get a Bank of America statement rather than the Wikipedia article on "financial statement," or the pages associated with monetary statements from alternative banks.

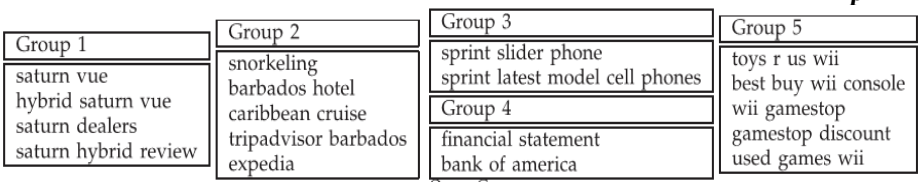
II. PROBLEM FORMULATION

In this paper, we have a tendency to study the matter of organizing a user's search history into a collection of query groups in an automatic and dynamic fashion.

Organizing the query groups at intervals a user's history is difficult for variety of reasons. First, connected queries might not seem near each other, as a groundwork task might span days or perhaps weeks. This is often difficult for the interleaving of queries and clicks from totally different search tasks thanks to users' multitasking [36], gap multiple browser tabs, and often dynamic search topics. For example, in Fig. 1a, the connected queries "hybrid saturn vue" and "Saturn dealers" square measure separated by several unrelated queries. This limits the effectiveness of approaches hoping on time or sequence to spot connected queries. Second, connected queries might not be textually similar. as an example, in Fig. 1b, the connected queries "tripadvisor barbados" and "Caribbean cruise" in group a pair of don't have any words in common. Therefore, relying exclusively on string similarity is additionally deficient. Finally, as users might also manually alter their various query groups, any automatic query grouping has got to respect the manual efforts or edits by the users.

Time	Query	Time	Query
10:51:48	saturn vue	12:59:12	saturn dealers
10:52:24	hybrid saturn vue	13:03:34	saturn hybrid review
10:59:28	snorkeling	16:34:09	bank of america
11:12:04	barbados hotel	17:52:49	caribbean cruise
11:17:23	sprint slider phone	19:22:13	gamestop discount
11:21:02	toys r us wii	19:25:49	used games wii
11:40:27	best buy wii console	19:50:12	tripadvisor barbados
12:32:42	financial statement	20:11:56	expedia
12:22:22	wii gamestop	20:44:01	sprint latest model cell phones

a) Users' search history



b) Query Grouping

Fig. 1. Search history of a real user over the period of one day together with the query groups.

Every query group may be a collection of queries by a similar user that measures query relevant to every alternative around standard information he wants. These query groups square measure dynamically updated because the user problems new queries, and new query groups is also created over time. to raised illustrate our goal, we have a tendency to show in Fig. 1a a collection of queries from the activity of a true user on the Bing programme over the amount of 1 day, at the side of the corresponding query groups in Fig. 2b: the primary query group contains all the queries that square measure associated with Saturn vehicles. The opposite groups, severally, pertain to Barbados vacation, sprint phone, financials, and game console.

Internet usage mining the pattern extraction algorithms square measure applied on the log information when they're processed. Therefore preprocessing is incredibly a lot of vital and should be allotted with correct care. whereas preprocessing the online access log the on top of points ought to be taken into thought in order that it'll turn out an honest set of access logs for pattern extraction [37].

III. RELATED WORK

Our work differs from these prior works in the following aspects. First, the query-log based features in [4], [5] are extracted from co-occurrence statistics of query pairs. In our work, we additionally consider query pairs having common clicked URLs and we exploit both co-occurrence and click information through a combined query fusion graph. Jones and Klinkner [4] will not be able to break ties when an incoming query is considered relevant to two existing query groups. Additionally, our approach does not involve learning and thus does not require manual labeling and retraining as more search data come in; our Markov random walk approach essentially requires maintaining an updated query fusion graph. Finally, our goal is to provide users with useful query groups on-the-fly while respecting existing query groups. On the other hand, search task identification is mostly done at server side with goals such as personalization, query suggestions [5], etc. Some prior work also looked at the problem of how to segment a user's query streams into "sessions." In most cases, this segmentation was based on a "time-out thresh-old" [21], [22], [23], [24], [25], [26], [27]. Some of them, such as [23], [26], looked at the segmentation of a user's browsing activity, and not search activity. Silverstein et al. [27] proposed a time-out threshold value of 5 minutes, while others [21], [22], [24], [25] used various threshold values. As shown in Section 5, time is not a good basis for identifying query groups, as users may be multitasking when searching online [3], thus resulting in interleaved query groups. The notion of using text similarity to identify related queries has been proposed in prior work. He et al. [24] and Ozmutlu and C¸ avdur [28] used the overlap of terms of two queries to detect changes in the topics of the searches. Lau and Horvitz [29] studied the different refinement classes based on the keywords in queries, and attempted to predict these classes using a Bayesian classifier. Radlinski and Joachims [30] identified query sequences (called chains) by employing a classifier that combines a time-out threshold with textual similarity features of the queries, as well as the results returned by those queries. While text similarity may work in some cases, it may fail to capture cases where there is "semantic" similarity between queries (e.g., "ipod" and "apple store") but no textual similarity. In Section 5, we investigate how we can use textual similarity to complement approaches based on search logs to obtain better performance.

A. Query (or Query Group) Relevance

To ensure that every query group contains closely connected and relevant queries and clicks, it's vital to possess an appropriate relevance sim between this query singleton group s_c and an existing query group si2. There are a unit variety of attainable approaches to work out the relevance between s_c and s_i. Below, we tend to define variety of various relevance metrics that we are going to later use as baselines in experiments.

Time. One could assume that s_c and s_i area unit somehow relevant if the queries seem near one another in time within the user's history. In alternative words, we tend to assume that users typically issue terribly similar queries and clicks at intervals a brief amount of your time. During this case, we tend to outline the subsequent time-based relevancy metric sim_{time} that may be utilized in place of sim.

Definition 1 (Time). Sim_{time}(sc,si) is defined as the inverse of the time interval (e.g., in seconds)

$$sim_{time}(s_c, s_i) = \frac{1}{|time(q_c) - time(q_i)|}.$$

The queries qc and qi are the most recent queries in sc and si, respectively. Higher sim time values imply that the queries are temporally closer.

Text. On a unique note, we have a tendency to could assume that 2 query group's are similar if their queries ar textually similar. Matter similarity between 2 sets of words may be measured by metrics like the fraction of overlapping words (Jaccard similarity [32]) or characters (Levenshtein similarity [33]). We will therefore outline the subsequent 2 text-based relevancy metrics which will be utilized in place of sim.

Definition 2 (Jaccard). $sim_{jaccard}$(sc,si) is defined as the fraction of common words between qc and qi as follows:

$$sim_{jaccard}(s_c, s_i) = \frac{|words(q_c) \cap words(q_i)|}{|words(q_c) \cup words(q_i)|}$$

ATSP. This technique is predicated on the principle that 2 queries issued in succession within the search logs area unit closely connected. In [31], the authors gift Answer that initial reorders a sequence of user queries to group similar queries along by determination an instance of the ATSP. Once the queries area unit reordered, query groups area unit generated by determinative "cut points" within the chain of queries, i.e., two ordered queries whose similarity is a smaller amount than a threshold. Note that ATSP must treat the entire set of queries that we have a tendency to associate degree interest} in grouping because it involves an initial rearrangement step.

Definition 3 (ATSP). $sim_{ATSP}(s_c,s_i)$ is defined as the number of times two queries, q_c and q_i, appear in succession in the search logs over the number of times q_c appears. More formally

$$sim_{ATSP}(s_c, s_i) = \frac{freq(q_c, q_i)}{freq(q_c)}$$

In our work we consider both query pairs having common clicked URLs and the query reformulations through a combined query fusion graph.

IV. QUERY SIMILARITIES FOR SEARCH LOGS

We currently develop the machinery to outline the query connection supported internet search logs. Our live of connection is geared toward capturing two necessary properties of relevant queries, namely:

1) Queries that regularly seem along as reformulations and

2) Queries that have evoked the users to click on similar sets of pages.

We have a tendency to begin our discussion by introducing three search behavior graphs that capture the same properties. Following that, we have a tendency to show however we will use these graphs to cipher query connection and the way we will incorporate the clicks following a user's query so as to reinforce our connection metric.

A. Search Behaviour Graphs

We derive three kinds of graphs from the search logs of an advert computer programmed. The query reformulation graph, QRG, represents the connection between a combine of queries that area unit doubtless reformulations of every different. The query click graph, QCG, represents the connection between 2 queries that regularly cause clicks on similar URLs. The query fusion graph, QFG, merges the data within the previous 2 graphs. All 3 graphs area unit outlined over a similar set of vertices V_Q, consisting of queries that seem in a minimum of one among the graphs, however their edges area unit outlined otherwise.

1) Query Reformulation Graph

One way to spot relevant queries is to think about query reformulations that area unit usually found inside the query logs of an exploration engine. If two queries that area unit issued consecutively by several users occur oftentimes enough, they're doubtless to be reformulations of every different. to live the connection between two queries issued by a user, the time-based metric, sim_{time}, makes use of the interval between the timestamps of the queries inside the user's search history. In distinction, our approach is outlined by the applied math frequency with that two queries seem next to every different within the entire query log, over all of the users of the system.

To this end, based on the query logs, we construct the query reformulation graph, QRG=(V_Q,E_{QR}), whose set of edges, EQR, are constructed as follows: for each query pair (q_i,q_j), where qi is issued before q_i within a user's day of activity, we count the number of such occurrences across all users' daily activities in the query logs, denoted $count_r(q_i,q_j)$ .Assuming infrequent query pairs are not good reformulations of each other, we filter out infrequent pairs and include only the query pairs whose counts exceed a threshold τ_r. For each (q_i,q_j), with countr(q_i,q_j)>= τ_r, we add a directed edge from q_i to q_j to E_{QR}. The edge weight, $w_r(q_i,q_j)$, is defined as the normalized count of the query transitions

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

2) Query Click Graph

A different way to capture relevant queries from the search logs is to consider queries that are likely to induce users to click frequently on the same set of URLs. For example, although the queries "ipod" and "apple store" do not share any text or appear temporally close in a user's search history, they are relevant because they are likely to have resulted in clicks about the ipod product. In order to capture such property of relevant queries, we construct a graph called the query click graph, QCG. We first start by considering a bipartite click-through graph, CG=($V_Q \cup V_U$,E_C), used by Fuxman et al. [34]. CG has two distinct sets of nodes corresponding to queries, V_Q, and URLs, V_U, extracted from the click logs. There is an edge (q_i, u_k) ∈ EC,, if query q_i was issued and URL uk was clicked by some users. We weight each edge (q_i, u_k) by the number of time sqi was issued and u_k was clicked, $count_c(q_i, u_k)$. As before, we filter out infrequent pairs using a threshold Tc. In this way, using the CG, we identify pairs of queries that frequently lead to clicks on similar URLs. Next, from CG, we derive our query click graph, QCG= (V_q, E_{qc}), where the vertices are the queries, and a directed edge from q_i to q_j exists if there exists at least one URL, u_k, that both q_i and q_j link to in CG. The weight of edge (q_i,q_j) in QCG, $w_c(q_i,q_j)$, is defined as the weighted asymmetric Jaccard similarity [32] 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 q_j is more related to q_i if more of q_i's clicks fall on the URLs that are also clicked for q_j.

3) Query Association Graph

Search engine performance for a particular query is typically measured using relevance metrics such as precision and recall. Some method found a reasonable correlation between many information retrieval metrics and satisfaction with result rankings. Beyond performance measurement, research on predicting query performance has been conducted to understand differences in the quality of search results provided by search systems for different queries. Such predictions do not require relevance judgments (at least not when the models are being applied, but perhaps during a separate training phase) and can be used to determine when to use additional computational resources or use alternative methods (e.g., specialized ranking algorithms or different interface support) to improve results for difficult queries. While it has been shown that different query representations or retrieval models improves search performance, it is more challenging to accurately predict which methods to use for a particular query.

We find association rules(AR) between two queries if confidence within these two queries is high. So both queries are related to each other as occurrence of users' search log. Using those attributes, we mine association rules to identify similar queries. Among many association rule learning techniques, we selected the apriori algorithm because our system needs to handle a large amount of Web queries and it can effectively work on such datasets.

To this end, based on the query logs, we construct the query association graph, QAG=(V_Q,E_{QA}), whose set of edges, E_{QA}, are constructed as follows: for each query pair (q_i,q_j), where q_i is having high confidence value with association to q_j within a user's day of activity, we count the number of such occurrences across all users' daily activities in the query logs, denoted $count_{as}(q_i,q_j)$. Assuming infrequent query pairs are not good reformulations of each other, we filter out infrequent pairs and include only the query pairs whose counts exceed a threshold value, as. For each (q_i,q_j) with $count_{as}$ (q_i,q_j)>=Tas, we add a directed edge from q_i to q_j to E. The edge weight, $w_{as}(q_i,q_j)$, is defined as the normalized count of the query transitions.

$$w_{as}(q_i, q_j) = \frac{count_{as}(q_i, q_j)}{\sum_{(q_i,q_k)\epsilon\varepsilon_{QR}} count(q_i q_k)}$$

B. Query Fusion Graph

The query reformulation graph QRG, the query click graph QCG and the query association graph QAG capture three important properties of relevant queries, respectively. In order to make more effective use of these properties, we combine the query reformulation information within QRG and the query-click information QCG and query association graph QAG into a single graph, QFG=(V_Q,E_{QF}), that we refer to as the query fusion graph. At a high level, E_{QF} contains the set of edges that exist in E_{QR} , E_{QC} or in E_{QA}. The weight of edge (q_i,q_j) in QFG, $w_f(q_i,q_j)$, is taken to be a linear sum of the edge's weights, $w_r(q_i,q_j)$ in E_{QR} , w_c(qi, qj) in E_{QC}, $w_{as}(q_i,q_j)$ in E_{QA} as follows:

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

The relative contribution of the two weights is controlled by α, and we denote a query fusion graph constructed with a particular value of α, β and γ such as α+β+γ =1 for QFG.

V. EXPERIMENTAL SETUP

In this section, we study the behavior and performance of our algorithms on partitioning a user's query history into one or more groups of related queries. For example, for the sequence of queries "caribbean cruise"; "bank of america"; "expedia"; "financial statement", we would expect two output partitions: first, {"caribbean cruise," "expedia"} pertaining to travel-related queries, and, second, {"bank of america," "financial statement"} pertaining to money-related queries.

A. Dataset

To this finish, we tend to obtained the query reformulation and query click graphs by merging variety of monthly search logs from a billboard computer program. every monthly photo of the query log adds around twenty four % new nodes and edges within the graph compared to the specifically preceding monthly photo, whereas around ninety two % of the mass of the graph is obtained by merging 9 monthly snapshots. to cut back the result of noise and outliers, we tend to cropped the query reformulation graph by keeping solely query pairs that appeared a minimum of double (Tq ¼2), and therefore the query click graph by keeping solely query-click edges that had a minimum of ten clicks (Tc ¼10).

In order to make check cases for our algorithms, we tend to used the search activity (comprising a minimum of 2 queries) of a collection of two hundred users (henceforth known as theRand200data set) from our search log. to get this set, users were picked every which way from our logs, associated 2 human labelers examined their queries and appointed them to either an existing group or a brand new group if the labelers deemed that no connected group was gift. A user's queries were enclosed in theRand200 knowledge set if each labelers were in agreement so as to cut back bias and subjectiveness whereas grouping. The labelers were allowed access to the online so as to see if 2 ostensibly distant queries were truly connected (e.g., "dainik bhaskar " and "star news"). the common variety of groups within the knowledge set was three.84 with thirty % of the users having queries classified in additional than three groups.

B. Performance Metric

To measure the standard of the output groupings, for every user, we tend to begin by computing query pairs within the labeled and output groupings. Two queries kind a try if they belong to an equivalent cluster, with one queries pairing with a special "null" question. To judge the performance of our algorithms against the groupings made by the labelers, we'll use

the Rand Index [35] metric, that could be a ordinarily used live of similarity between 2 partitions. The Rand Index similarity between 2 partitions X,Y of n components every is outlined as $RandIndex(X,Y) = (a+b)/\binom{n}{2}$, wherever a is that the range of pairs that square measure within the same set in X and also the same set in Y, and b is that the range of pairs that square measure in several sets in x and in several sets in Y. Higher RandIndex values indicate higher ability of grouping connected queries along for a given formula. Our formula to realize the most effective performance onRand200 supported the RandIndex metric. We tend to follow an equivalent approach for the baselines that we tend to enforced also. We'll additionally appraise the approaches on further take a look at sets (Lo100, Me100, and Hi100). to achieve a live of usage data for a given user, we glance at the common outdegree of the user's queries (average outdegree), also because the average counts among the outgoing links (average weight) within the query reformulation graph. so as to check the results of usage data on the performance of our algorithms, we tend to created 3 further take a look at sets of a hundred users every. The sets we tend tore additionally manually labeled as we delineated. The primary set, Lo100contains the search activity of a hundred users, with average out-degree < 5. Similarly, Me100 contains user activity for users having out-degree >5 but <10 end Hi100 having out-degree >10.

Based on these information sets, we tend to judge once more the performance of our algorithms and that we show the ends up in rock bottom 3 lines of Table one. As we will see from the table, for QFG, subsets with higher usage info conjointly tend to own higher RandIndex values. Hi100 (RandIndex=0.89) performs higher than Me100(RandIndex=0.947), that successively outperforms Lo100(RandIndex=1.0). ATSP shows the same trend (higher usage shows higher performance) and it outperforms QFG at the Lo100data set. CoR's performance is additional or less similar for the various information sets that is anticipated because it doesn't use the graphs directly. For Jaccard, it's best once the property round the queries among a user's session is comparatively low. We tend to don't observe any important distinction within the performance of the opposite baselines (Time and Levenshtein) in these new information sets.

TABLE I. COMPARATIVE PERFORMANCE (RANDINDEX) OF OUR METHODS

	Time	**Levenshtein**	**Jaccard**	**CoR**	**ATSP**	**QFG**	**Proposed**
Rand200	0.683	0.721	0.750	0.807	0.831	0.860	**0.867**
Lo100	0.620	0.732	0.762	0.794	0.832	0.821	**1.000**
Me100	0.632	0.712	0.748	0.802	0.857	0.868	**0.947**
Hi100	0.654	0.729	0.742	0.809	0.871	0.882	**0.890**

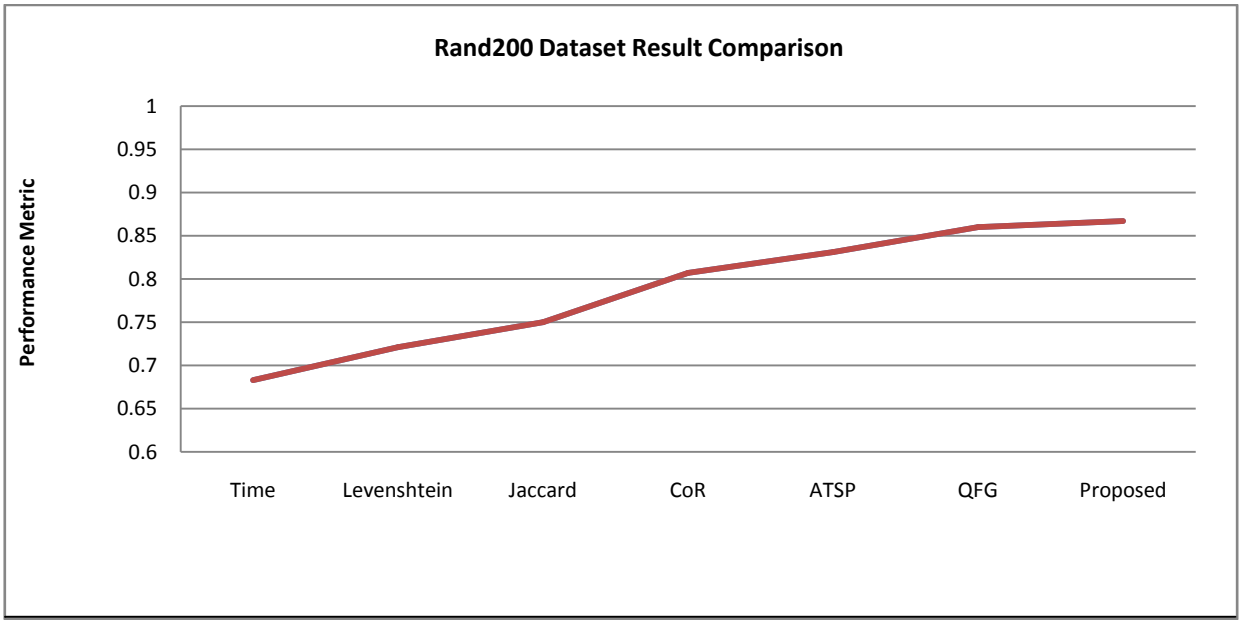


Fig. 2 Performance metric for Rand200 dataset

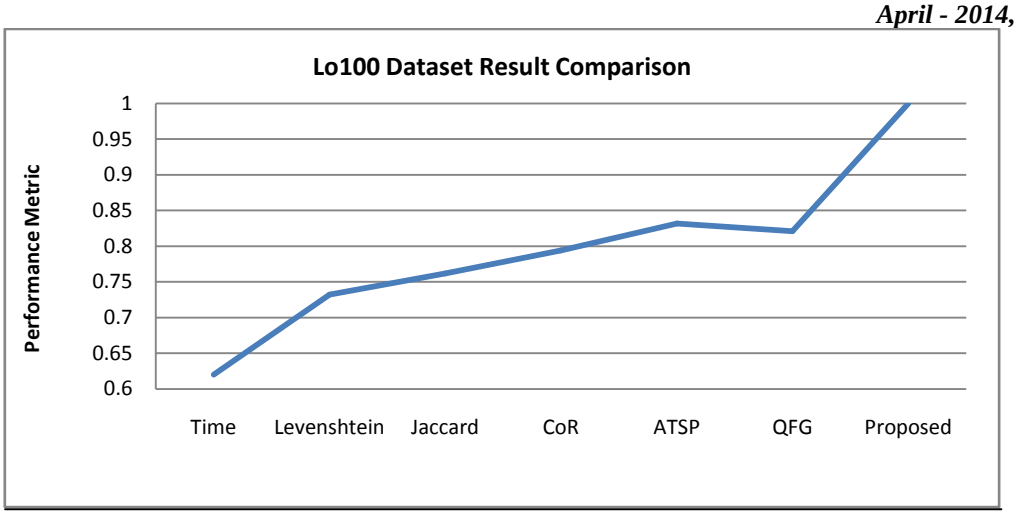


Fig. 3 Performance metric for Lo100 dataset

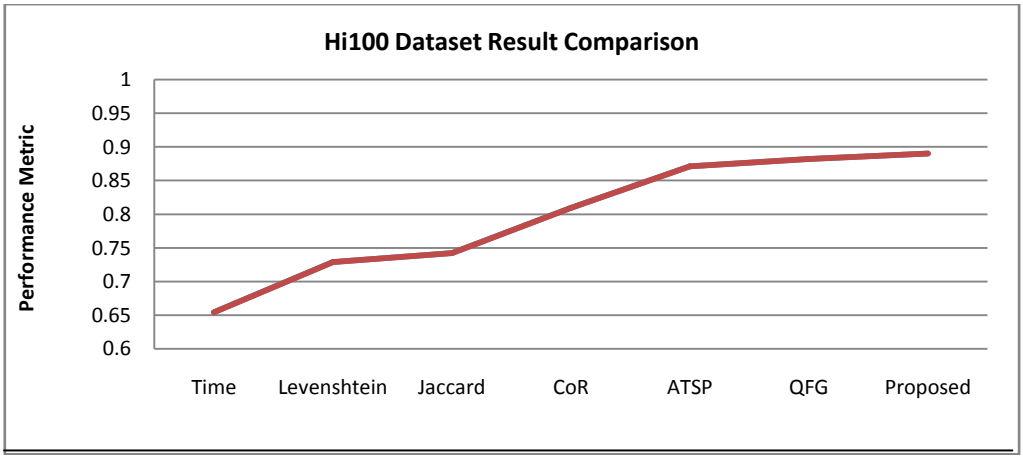


Fig. 4 Performance metric for Hi100 dataset

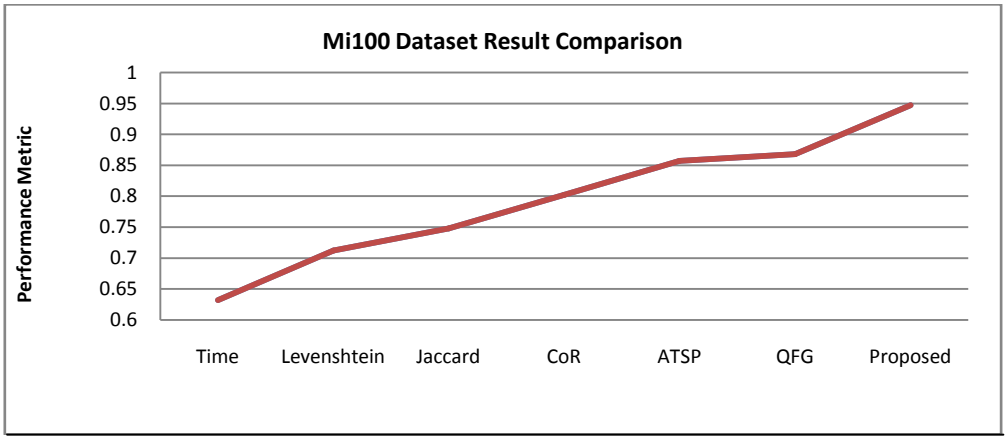


Fig. 5 Performance metric for Me100 dataset

Figures 2 shows result comparison of performance metric for Rand200 dataset with other existing algorithms and it verifies that proposed method outperformed better than others. Proposed method implemented reformation graph, click graph as well as association query information, hence Gives better results. The proposed method also compare with other dataset low100, me100 and hi100 dataset as explain above for all dataset proposed method gives high performance index for grouping queries.

VI. CONCLUSION

We have presented an approach for automatic generation of query grouping of periodic query search patterns from user web usage logs. We built a system that creates user profiles based on implicitly collected information, specifically the queries submitted of user search results. We were able to demonstrate that information readily available to search engines is sufficient to provide significantly improved query grouping. The query reformulation, click graphs and

Association rule graph contain helpful data on user behaviour once looking on-line. During this paper, we tend to show however such data may be used effectively for the task of organizing user search histories into query groups. Additionally, we tend to propose combining the three graphs into a query fusion graph. We tend to show that our approach that's supported probabilistic random walks over the query fusion graph outperforms time-based, keyword similarity-based and association rule approaches. We tend to conjointly realize in combining our technique with keyword similarity-based ways. This proposed method gives better rank index compare to existing methods for query grouping.

REFERENCES

[1] Dr. G. K. Gupta, "Introduction to Data Mining with Case Studies", PHI Publication, 2005.

[2] 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, Page 12-23.

[3] Adel T. Rahmani and B. Hoda Helmi, "EIN-WUM an AIS-based Algorithm for Web Usage Mining", Proceedings of GECCO'08, Atlanta, Georgia, USA, ACM978-1-60558-130-9/08/07, 2008, Pp. 291-292.

[4] Shailey Minocha, Nicola Millard, Lisa Dawson, "Integrating Customer Relationship Management Strategies in (B2C) E-Commerce Environments", IFIP Conference on Human-Computer Interaction- INTERACT, 2003.

[5] C. Ramya, G. Kavitha, K. S. Shreedhara, "Preprocessing: A Prerequisite for Discovering Patterns in Web Usage Mining Process", Computing Research Repository - CORR, vol. abs/1105.0, 2011.

[6] V. Chitraa, Antony Selvdoss Davamani, "A Survey on Preprocessing Methods for Web Usage Data", Computing Research Repository-CORR, Vol. abs/1004.1, 2010. Nizar R. Mabroukeh, Christie I. Ezeife, "A taxonomy of sequential pattern mining algorithms", ACM Computing Surveys - CSUR, Vol. 43, No. 1, 2010, Pp. 1-41.

[7] Francesco Moscato, Nicola Mazzocca, Valeria Vittorini, Giusy Di Lorenzo, Paola Mosca, Massimo Magaldi, "Workflow Pattern Analysis in Web Services", High Performance Computing and Communications - HPCC, 2005, Pp. 395-400.

[8] 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, IEEE, 2012, Page 912-925.

[9] R. Jones and K.L. Klinkner, "Beyond the Session Timeout: Automatic Hierarchical Segmentation of Search Topics in Query Logs," Proc. 17th ACM Conf. Information and Knowledge Management (CIKM), 2008.

[10] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna, "The Query-Flow Graph: Model and Applications," Proc. 17th ACM Conf. Information and Knowledge Management (CIKM), 2008.

[11] P. Anick, "Using Terminological Feedback for Web Search Refinement: A Log-Based Study," Proc. 26th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, 2003.

[12] B.J. Jansen, A. Spink, C. Blakely, and S. Koshman, "Defining a Session on Web Search Engines: Research Articles," J. the Am. Soc. for Information Science and Technology, vol. 58, no. 6, pp. 862-871, 2007.

[13] L.D. Catledge and J.E. Pitkow, "Characterizing Browsing Strategies in the World-Wide Web," Computer Networks and ISDN Systems, vol. 27, no. 6, 1995, pp. 1065-1073.

[14] D. He, A. Goker, and D.J. Harper, "Combining Evidence for Automatic Web Session Identification," Information Processing and Management, vol. 38, no. 5, 2002, pp. 727-742.

[15] R. Jones and F. Diaz, "Temporal Profiles of Queries," ACM Trans. Information Systems, vol. 25, no. 3, 2007, p. 14.

[16] A.L. Montgomery and C. Faloutsos, "Identifying Web Browsing Trends and Patterns," Computer, vol. 34, no. 7, July 2001, pp. 94-95.

[17] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz, "Analysis of a Very Large Web Search Engine Query Log," SIGIR Forum, vol. 33, no. 1, 1999, pp. 6-12.

[18] 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, 2005, pp. 1243-1262.

[19] T. Lau and E. Horvitz, "Patterns of Search: Analyzing and Modeling Web Query Refinement," Proc. Seventh Int'l Conf. User Modeling (UM), 1999.

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

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

[22] 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.

[23] T. Radecki, "Output Ranking Methodology for Document- Clustering-Based Boolean Retrieval Systems," Proc. Eighth Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, 1985, pp. 70-76.

[24] V.R. Lesser, "A Modified Two-Level Search Algorithm Using Request Clustering," Report No. ISR-11 to the Nat'l Science Foundation, Section 7, Dept. of Computer Science, Cornell Univ., 1966.

[25] R. Baeza-Yates, "Graphs from Search Engine Queries," Proc. 33rd Conf. Current Trends in Theory and Practice of Computer Science (SOFSEM), vol. 4362, pp. 1-8, 2007.

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

[27] 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.

[28] 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
[29] D. Beeferman and A. Berger, "Agglomerative Clustering of a Search Engine Query Log," Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), 2000.
[30] 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.
[31] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna, "The Query-Flow Graph: Model and Applications," Proc. 17th ACM Conf. Information and Knowledge Management (CIKM),2008
[32] Lecture Notes in Data Mining,M. Berry, and M. Browne, eds. World Scientific Publishing Company, 2006.
[33] V.I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals,"Soviet Physics Doklady,vol. 10, pp. 707-710, 1966.
[34] 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.
[35] W.M. Rand, "Objective Criteria for the Evaluation of Clustering Methods" J. the Am. Statistical Assoc.,vol. 66, no. 336, pp. 846-850, 1971.
[36] A. Spink, M. Park, B.J. Jansen, and J. Pedersen, "Multitasking during Web Search Sessions,"Information Processing and Manage-ment,vol. 42, no. 1, pp. 264-275, 2006.
[37] 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),2000.

