Time Dependent Approach for Query and URL Recommendations Using Search Engine Query Logs

R.Umagandhi, A.V.Senthilkumar

Abstract - **Search engine retrieves the significant and essential information from the web, based on the query keyword given by the user. Query log file is a repository contains every query request and its navigation in the search engine and maintained either in the system desktop or in the proxy server. This paper has proposed an algorithm for query and URL recommendation which is based on the user's search histories and a click through data. A new framework is constructed here based on the navigation time. First, the proposed algorithm identifies frequently accessed Queries and URLs from the log file using the frequent pattern generation algorithms. Next hub and authority weights are calculated for the frequent items. The similar queries are clustered; it uses the temporal characteristics of historical click-through data. The intuition is to reveal that more accurate semantic similarity of queries can be obtained by considering the timestamps of the log data. The cluster generated in this approach is used to provide query and URL recommendations to the user. Finally the method has been evaluated using real data set from the search engine query log.**

Index Terms - **Query, URL, Modified Hits, Prefix span and Up down Directed Acyclic Graph.**

I. INTRODUCTION

The tremendous growth of World Wide Web has paved the way for getting the required information from the web. Search engines are used to retrieve the result from the web in terms of web snippets for the query given by the user. The retrieved result may not be relevant all the time. At times irrelevant and redundant results are also retrieved by the search engine because of the short and ambiguous query keywords [1]. The user scans the search result from the top to the bottom according to Joachim's [2] and then decides whether the web snippet is either relevant or irrelevant. A study done by C. Silverstein [3] on Alta Vista Query Log has shown that more than 85% of the queries contain less than three terms and the average length of the query is 2.35 terms. So the shorter length query does not provide any meaningful, relevant and needed information to the users. Table I shows some of the examples for ambiguous query keywords. [4] Reported that up to 23.6% of web search queries are ambiguous, this causes poor retrieval results.

TABLE I
AMBIGUOUS QUERY KEYWORDS

Query	Search Topics
Java	Programming Language, Bike, Country
Apple	Company, Fruit, System

When the user is not satisfied with the result given by the search engine for the initial input query, Query recommendation technique is used as it provides suggestions to the user to frame relevant and meaningful queries in future to retrieve the relevant results. The recommendation made by the search engine depends on the real intent of the user, and the user's intent is analysed from the search histories. For example, consider a user who submits the query term 'apple' in the search process, but he review the result only for 'apple iPod' and not for the 'apple fruit'. Here the user's interest is on apple iPod only. The query recommendation system provides suggestions on the iPod when the same query 'apple' is triggered by the same user next time. Here the recommendation is given by considering the user's past navigations.

Consider another example, user U_1 wants to get the information on 'Android applications'. Not keeping in mind of its keyword he enters the query keyword as 'mobile applications'. The top documents do not have the information on 'Android applications'. After a long searching process U_1 gets the result and clicks the correct *URL*. But another user U_2 enters the correct query on Android applications by using the correct keywords and clicks the same *URL* which is clicked by U_1. Our algorithm generates the cluster which contains the users who have the similar intents (that is U_1 and U_2), the query keywords and the URLs are clicked for the queries. The cluster is used to provide the query recommendations to the first user U_1 by using the keywords of U_2. Here the query recommendation is a collaborative technique, which is based on the intent of more than one user.

Here the major contributions of the work are summarized as follows:

- First, the frequently accessed queries and URLs are identified from the log file by considering the prefix and suffix patterns of the Queries and Clicked URLs using the existing frequent pattern generation algorithms namely Prefix Span [5] and Up down Directed Acyclic Graph [6].
- t-measure, Hub and authority weight is calculated for the frequently accessed queries and URLs.

R.Umagandhi, Associate Professor, is with the Department of Computer Technology, Kongunadu Arts and Science College, Coimbatore, Affiliated to Bharathiar University, Tamil Nadu - 641 029, India.
(email: umakongunadu@gmail.com)

Dr.A.V.Senthilkumar Director, is with the Department of Computer Applications, Hindustan College of Arts and Science, Coimbatore, Tamil Nadu - 641 028, India. (email: avsenthilkumar@yahoo.com)

(Advance online publication: 19 August 2013)

- Frequently accessed queries and URLs are clustered using the Agglomerative clustering algorithm. This cluster contains a set of similar queries.
- Finally the queries are recommended to the user to frame the meaningful queries in future. Here the query recommendation is either content based or collaborative based. Content based approach gives the recommendation based on the search histories and navigational behaviour of the user. Whereas the recommendation from Collaborative approach is based on the preferences of multiple users.

The rest of this paper is organised as follows: Section II reviews the related work. Section III defines some commonly used terms in the proposed work. Pre-processing of the log file and the architecture of the proposed work are explained in section IV. Generation of query clusters by considering the prefix patterns is discussed in Section V. Query cluster generation by considering the prefix and suffix patterns is explained in Section VI. In Section VII, Experiments and results are discussed. Section VIII concludes the paper.

II. RELATED WORKS

Web is getting expanded day by day. Many search engines are used to retrieve the information from the web. In such situations, it is the responsibility of service provider to provide suitable, significant and quality information to the user against their query submitted to the search engine. Query Recommendation is an information retrieval technology to recommend identical or related queries for a particular query [7]. Search engine provides the query recommendation in two ways. Firstly the suggestion is given while the query is typed by the user. This technique is used to frame the queries at hitting time [8]. Secondly the lists of queries are recommended at the end of the search result. The proposed work provides the query recommendation to frame the meaningful and reliable queries in future.

Search engines retrieve the result in terms of web snippets for the query given by the user and its navigational information is stored in query log. Much research has been done in query expansion, Query suggestions and Query recommendations [9][10][11]. The similar queries in the log entries are clustered based on similarity measure. [12][13] Recommend the query using similarity based query cluster.

Query recommendations are often based on clustering methods with the inconvenience that queries falling in the same cluster are some time more ambiguous and less helpful than the original query [14]. The frequently used queries and URLs in the log file are identified using Prefix Span [5] or Up down Directed Acyclic Graph [6]. The similar queries and URLs are clustered; the cluster recommends the queries. Hub and authority weight is calculated for each unique URL [15]. The total weight value is considered for generating the query cluster.

A good query recommendation system should observe the following properties [16]

- Relevance: Recommended queries should be semantically relevant to the user search query.
- Redundancy Free: The recommendation should not contain redundant queries that repeat similar search intents.
- Diversity: The recommendation should cover search intents of the different interpretations of the keywords given in the input query.
- Ranking: Highly relevant queries should be ranked first ahead of the less relevant ones in the recommendation list.
- Efficiency: Query recommendation provides online help. Therefore, the recommendation algorithms should achieve fast response times.

The query recommendation technique provided in this approach satisfies the properties defined in [16].

In this paper, association rules are generated between the frequent queries and frequent URLs without any pre assigned weights. [17][18] Discussed the weighted association rule mining where the items have some predefined weights. Ke Sun et al. [19] have introduced the new measure of items w-support for mining the association rules without any pre assigned weights.

III. GENERAL TERMS

Item: In this context an item is a query or clicked URL. When the queries are recommended, query becomes an item and when the URLs are recommended, URL becomes an item.

Candidate: Set of URLs accessed in a day or set of Queries given in a particular day.

Support: An item set X has support s in T if $s\%$ of the transactions in T contains X. support of an URL is calculated by number of times the URL accessed being divided by the total number of distinct URLs in the data set.

For example, consider the query log of AOL search engine. From the first 200 log entries, 148 unique URLs and 113 unique queries are retrieved. The URL http://www.google.com appears 5 times and its support value is 3%. The query 'lotto' appears 12 times and its support value is 10.6%.

Frequent Item: An item I is frequent if its support is higher than the user specified minimum support threshold.

Association rule: Consider $I = \{i_1, i_2 \ldots. i_n\}$ is a set of items and $T = \{t_1, t_2 \ldots.. t_n\}$ is a set of transactions where each transaction t_i consists of a subset of items in I. An association rule is of the form:

$$X \rightarrow Y, \quad X \in I, Y \in I, X \cap Y = \varnothing$$

Consider $Q = \{q_1, q_2 \ldots.. q_n\}$ set of queries. The association rule is $X \rightarrow Y, \quad X \in Q, Y \in Q, X \cap Y = \varnothing$

Confidence: Confidence is an interestingness measure of an association rule. The rule $X \rightarrow Y$ holds in T with confidence c if $c\%$ of transactions in T that contain X also contain Y.

$$\text{Confidence } (X \rightarrow Y) = \text{Support } (XUY) / \text{ Support}(X)$$

Lift: Lift is a simple correlation measure that compares the rule of confidence with the expected rule of confidence.

$$\text{Lift}(X \rightarrow Y) = \text{Confidence } (X \rightarrow Y) / \text{Support } (Y)$$

Association Rules from Query log file: We have generated the associations between the queries and the clicked URLs which are stored in the query log file. Consider the following traversal path of the user U_1 for the input query Q_1.

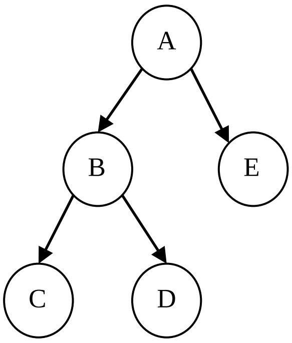


Fig. 1. Traversal path for Q_1 by U_1

Here the user clicks the document B and E from the document A. The referring URL for the documents C and D is B. The adjacency matrix representation for the traversal path given in Fig. 1 is

$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 1 & 0 & 0 & 1 \\ B & 0 & 0 & 1 & 1 & 0 \\ C & 0 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 \end{array}$$

The association rules generated for the above traversal path is

A→ B, A→ E, B→ C, B → D, AB→ C, AB → D

The Association rules generated from the query log may be based on the weights such as in degree, out degree, number of clicks, time spent on the web pages, and etc [11][12]. The rules are also generated without any pre assigned weights [13]. The proposed work generates the association rules between the queries and in between the URLs without considering any pre-assigned weights.

Hub: The hub identifies the URLs clicked for the query Q. In Fig. 2, the URLs A, B and C are accessed for the query Q.

Authority: The authority identifies the URLs pointed for the query Q. In Fig. 3, A, B and C are the URLs which have resources for the query Q. For example,

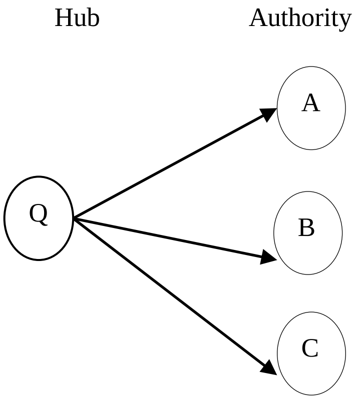


Fig. 2. Multiple Authorities

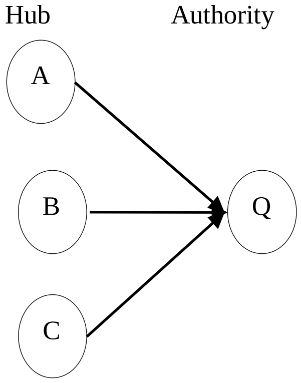


Fig. 3. Multiple Hubs

From Fig.2,

Hub (Q) = Number of out links from Q
= Authority (A) +Authority (B) +Authority(C)
= 3

From Fig. 3,

Authority (Q) = Number of in links to Q
= Hub (A) +Hub (B) + Hub(C) = 3

IV. QUERY LOG FILE

A. Architecture for Time Dependent Recommendations

Fig. 4 describes the architecture for time dependent recommendations. The user submits the query to the search engine interface. The user’s 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 not relevant 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, here the message “NoClick” is assigned to the attribute ClickURL. The pre-processed log entries are stored in the query log file.

The frequently occurred queries and URLs are identified and the associations among the frequent items are generated. The hub and authority weight for the frequent URLs are calculated. Next, the query cluster is generated based on the time stamp of the query. Consider a situation, either the query *Q* is issued several times or the user clicks different URLs or the same URL is clicked for different queries. For these cases the hitting time is considered and the weight is assigned for each time interval. The time weight of the recently triggered query is high when compared with the older query.

Based on the hub and authority weights of URLs and time weight, the queries are clustered. When the user supplies the same query next time, this cluster recommends the query.

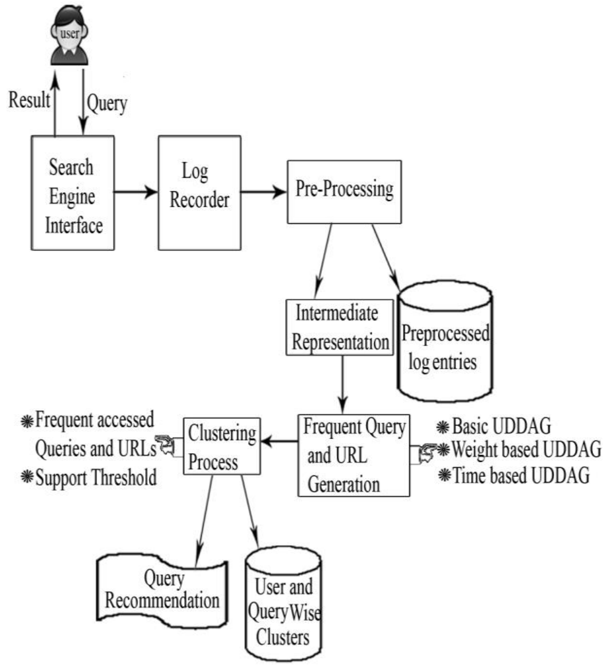


Fig. 4. Architecture for Time Dependent Recommendations

B. Pre-processing of Query Log file

In order to give the suggestions to frame the future queries, the search histories are analysed. To evaluate this work, we consider the AOL data set from 2006 -03 - 01 to 2006 – 05 - 31 (zola. di. unipi. it / smalltext/datasets.html). The data set contains 1975811 records and 19131507 words in 174 MB, based on our system's memory and its speed we consider a maximum of first 200 pre-processed records. The search histories are organized under the attributes

< AnonID Query QueryTime ItemRank ClickURL>

Table II shows the attributes and its description used in the data set.

TABLE II
AOL SEARCH ENGINE'S ATTRIBUTE AND ITS DESCRIPTION

Attribute	Description
AnonID	an anonymous user identifier
Query	the query issued by the user
QueryTime	The date and time on which the query was triggered by the user
ItemRank	If the user clicked on the search result, the rank of the item on which they clicked is listed.
ClickURL	If the user clicked on the search result, the domain portion of the URL in the web snippets is listed.

Table III shows the sample log entries in AOL search engine's data set.

The user 1038 in the last row is either obtained the information from the web snippets itself or does not satisfy with the result; hence the user does not click any URL. The query log entries are pre-processed by using the following steps.

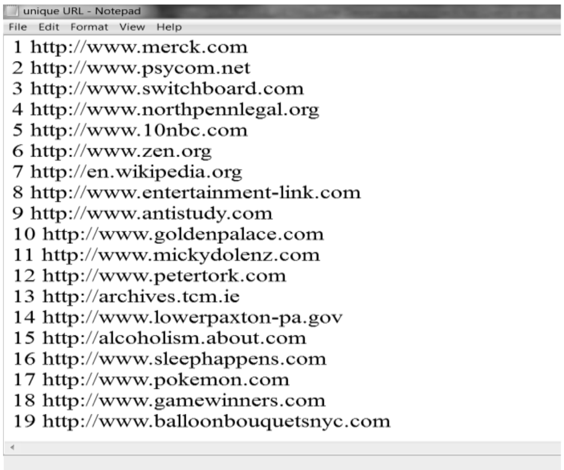
- If ItemRank and ClickURL attributes are empty then assign the message "Norank" and "NoClick" for the attributes in the data set, because for those entries, the user does not click any URL.

TABLE III
AOL - SAMPLE LOG ENTRIES

Anon ID	Query	QueryTime	Item Rank	ClickURL
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
309	whec tv in rochester ny	2006-05-11 14:54:43	1	http://www.10nbc.com
366	intravenous	2006-03-01 17:16:19	3	http://en.wikipedia.org
647	rabbit hole the broad way play	2006-03-01 22:15:33	2	http://www.entertainment-link.com
1038	tow truck	2006-03-01 23:17:31	No Click	NoRank

- If the Query keyword is empty then the record is removed from the data set. The query log file is cleaned by using the algorithm given in [20] [21].
- The bad queries are removed from the query log. Bad queries are non-interpretable query. For example a query contains the IP address is considered as a bad query.
- Non-alpha-numerical characters from query strings except for '.' (dot) and redundant space characters are removed. Singular and plural words are treated as same. All the uppercase letters are converted into lowercase.
- Stop words (www.link-assistant.com/seo-stop-words. html) from the query keywords are removed.
- Queries that appear only once in the search log are also considered because these queries are rare queries.
- Unique Queries and URLs are retrieved from the query log. Users and their sessions are identified.

From the first 200 pre-processed log entries, 148 unique URLs and 113 unique queries are retrieved. A numeric identifier is assigned for the retrieved unique URL which is shown in Fig. 5.



1 http://www.merck.com
2 http://www.psycom.net
3 http://www.switchboard.com
4 http://www.northpennlegal.org
5 http://www.10nbc.com
6 http://www.zen.org
7 http://en.wikipedia.org
8 http://www.entertainment-link.com
9 http://www.antistudy.com
10 http://www.goldenpalace.com
11 http://www.mickydolenz.com
12 http://www.petertork.com
13 http://archives.tcm.ie
14 http://www.lowerpaxton-pa.gov
15 http://alcoholism.about.com
16 http://www.sleephappens.com
17 http://www.pokemon.com
18 http://www.gamewinners.com
19 http://www.balloonbouquetsnyc.com

Fig. 5. Unique URLs

Unique queries are retrieved from the log file and its size is 1.99 KB (2,038 bytes). After pre-processing, its size is reduced as 1.71 KB (1,760 bytes). Table IV shows some of the original query terms and its pre-processed form.

C. Time Wise URL Cluster

Time Independent query recommendation model [23] defines that the recommendation assumes all the navigations are treated as equal irrespective of the time stamps. Recent query preferences to react in a better way of the current trends than the older preferences do.

TABLE IV
ORIGINAL AND PRE-PROCESSED QUERY TERMS

Original Query Term	Pre-processed Query Term
psychiatric disorders	psychiatric disorders
Cyclothymia	cyclothymia
grooming in harrisburg pa	grooming harrisburg pa
subsidized housing in harrisburg pa	subsidized housing harrisburg pa
whec tv in rochester ny	whec tv rochester ny
pen pals for KIDS	pen pals kids
rabbit hole the broadway play	rabbit hole broadway play
CLIFF NOTES	cliff notes
friendship community center	friendship community center
rehabs in harrisburg pa	rehabs harrisburg pa

User interests are varying from time to time, that is the user has different needs at different time periods. For example, during the weekdays the user is interested about the ‘apple computers’ while in the weekends the user intent is on apple fruit or recipes. Hence we have to assign the weight for the time periods.

The algorithm CLUSTTIME is used to cluster the query log entries day wise, that is queries and URLs are grouped based on the date and time on which the navigation is occurred.

Definition: Time Schema

Time Schema *T*=(*R*, *C*) where *R* is set of access logs with time attribute and *C* is the Constraint. For example, Consider 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.

Definition: Time Cluster

Consider the Time Cluster D = (URL, T), where URL is the identifier assigned to each unique URL which is triggered at the particular time period T. D = (URLi, T) where i = 1, 2...144.

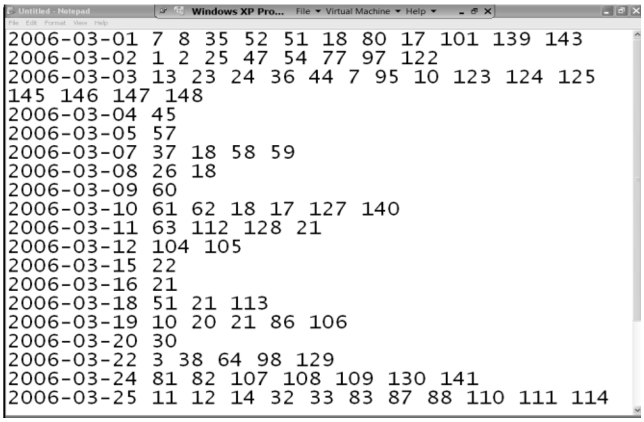
```
Algorithm CLUSTTIME
Input: Query log entries and Unique URLs Identifier
Output: Cluster of URLs day wise
begin
     For month (QueryTime) from 3 to 5 do
      For day (QueryTime) from 1 to 31 do
        Search the query log entry for the given month and day
               If found (log entry) then
                   Search the URLs id from URLLIST and Group
                   the URLs
end
```

Fig. 6 shows that 54 cluster of URLs generated by using the algorithm CLUSTTIME.

Hash Tables are used to calculate the hub and authority weights of the URLs in day wise URL clusters, which are given in Fig. 7. If the URL cluster of the 3rd month is represented by a similarity matrix then the size of the matrix is 31*144 because 144 unique URLs are identified from the first 200 pre-processed entries of the data set. If the hash table representation is used [22] to represent the URL cluster of 3rd month then the size of the table is r*c where

r = number of days the search engine is accessed in the 3rd month and

c = maximum number of URLs accessed in any day of the 3rd month.



```
2006-03-01 7 8 35 52 51 18 80 17 101 139 143
2006-03-02 1 2 25 47 54 77 97 122
2006-03-03 13 23 24 36 44 7 95 10 123 124 125
145 146 147 148
2006-03-04 45
2006-03-05 57
2006-03-07 37 18 58 59
2006-03-08 26 18
2006-03-09 60
2006-03-10 61 62 18 17 127 140
2006-03-11 63 112 128 21
2006-03-12 104 105
2006-03-15 22
2006-03-16 21
2006-03-18 51 21 113
2006-03-19 10 20 21 86 106
2006-03-20 30
2006-03-22 3 38 64 98 129
2006-03-24 81 82 107 108 109 130 141
2006-03-25 11 12 14 32 33 83 87 88 110 111 114
```

Fig. 6. Day wise URL cluster

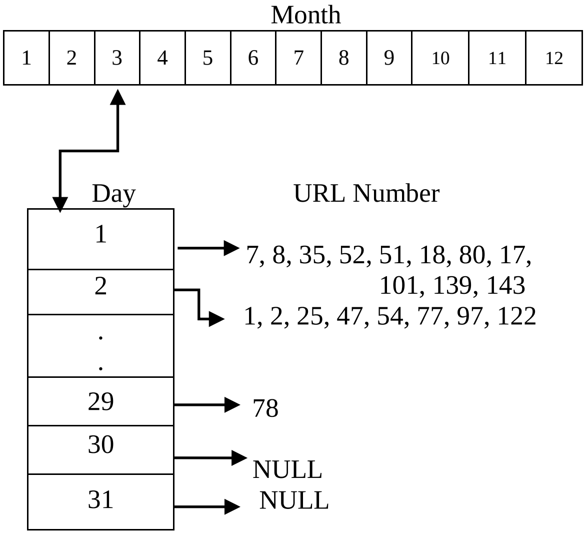


Fig. 7. Hash Table Representation

A new measurement is introduced here to assign the weights for the time period called it as t-measure. The algorithm CLUSTTIME gives the cluster of URLs day wise. If the URL u_1 is accessed in two different day’s t_1 and t_2 (t_1 occurs earlier than t_2) then the t-measure assigned to u_1 at t_1 is lesser than the t-measure assigned at t_2.

$$t - measure(u_{\mathrm{i}}) = Cluster\ number(u_i)\ / \sum_{i=1}^{n} \mathrm{i}$$

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

TABLE V
URLS ACCESSED ON 4 DIFFERENT DAYS

Date	URLs
2006-03-01	7,52, 51, 18, 80, 17
2006-03-14	8,10
2006-04-02	7,9
2006-05-01	7

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

t-measure (URL 7 at cluster 1) = 1 / (1+2+3+4) = 0.1
t-measure (URL 7 at cluster 3) = 3 / (1+2+3+4) = 0.3
t-measure (URL 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 URLs at first cluster is 0.1, second cluster is 0.2, third cluster is 0.3 and the last cluster is 0.4.

$$\text{Total (t-measure)} = \sum_{i=1}^{n} t - measure(cluster\ i)$$

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

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

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

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

where n=number of clusters. Total t-measure for the clusters in Table V is 0.1+0.2+0.3+0.4=1.

V. IDENTIFYING FREQUENT URLS USING PREFIX PATTERNS

The generation of association among all the unique URLs and queries are very tedious and ineffective process. Hence the frequently accessed URLs are obtained by considering the prefix patterns generation procedure. The basic prefix span identifies 13 one URL sets and 1 two URL sets that are frequently accessed. By using these URLs, totally 115 rules are generated. The URLs which satisfy the minimum support of 2 are considered as frequent URLs. Algorithm PrefixSpanBasic generates the frequent item set.

```
Algorithm PrefixSpanBasic
Input: URL Cluster, support threshold
Output: Frequently accessed URLs and Association rules
begin
Step 1: From URL Cluster, generates the URL patterns
Step 2: Find the count for each URL pattern
Step 3: If support (pattern) < threshold then delete the
        pattern otherwise generate the association rule for
        that pattern
Step 4: if support (rule) >= threshold then
        URLs used in the rule are considered as frequent
        URLs.
end
```

Next calculate the hub and authority weight for the unique URLs. The URLs which satisfy the minimum authority 1 are considered for recommendation. Totally 6 URLs are identified and 13 rules are generated.

```
Algorithm MHitsPrefixspan
Input: URL Cluster, support threshold
Output: Frequently accessed URLs with hub weight
begin
Step 1: Identify the frequent URLs and their corresponding
        queries using PrefixSpanBasic Algorithm
Step 2: Calculate the hub and authority weight for each URL
        using Hits Algorithm [15]
Step 3: if authority weight (URL) > =1 and support (URL)
        >= threshold then generate the association rule and
        URLs used in the rule are considered as frequent
        URLs.
end
```

Now t-measure is calculated for the frequently accessed URLs along with hub and authority weight generated from MHitsPrefixspan. The support, authority weight and t-measure value for the frequent URLs are given in Table VI.

TABLE VI
FREQUENT URLS WITH SUPPORT

PrefixSpanBasic		PrefixSpan with Hub and Authority		PrefixSpan with t-measure	
Frequent URL	Support	Frequent URL	Authority weight	Frequent URL	Time Weight
7	2.0	7	2.0	7	2.001
51	4.0	51	3.540	51	3.567
18	5.0	18	4.493	18	4.499
17	4.0	17	3.567	17	3.572
21	9.0	21	7.905	21	7.923
87	3.0	87	2.310	87	2.349
25	2.0				
22	3.0				
38	2.0				
10	2.0				
41	2.0				
40	2.0				
68	2.0				
18, 17	2.0				

The frequently accessed URLs are identified by considering the prefix patterns of the URL and it generates 13 association rules. Table VII shows the rules and their confidence value. For example, http://en.wikipedia.org =>> http:// www. txlottery. org is an association rule and its confidence value is 55%. The user 366 clicked the URL http://en.wikipedia.org on 2006-03-01 for the query 'intravenous'. The rule gives the recommendation to the user 366 as instead of selecting the URL http://en.wikipedia.org he may select the URL http:// www.txlottery.org which was selected by the user 309. Third rule in Table VII provides the recommendation to the users 309 and 366 by using the URLs clicked by 366 and 309 respectively.

VI. IDENTIFYING FREQUENT URLS USING PREFIX AND SUFFIX PATTERNS

An approach which is used to identify the frequent items which is based on Up Down Directed Acyclic Graph (UDDAG) and it is proposed as a fast pattern growth algorithm in [4]. UDDAG is a novel data structure, which supports bidirectional pattern growth from both ends of detected patterns. Prefix Span identifies the pattern in one direction. The performance of both the algorithms is discussed in section VII. The process of identifying the frequent patterns by using the prefix and suffix patterns are given in the algorithm MUDDAG.

```
Algorithm MUDDAG
Input: Unique URLs, URL Cluster and support threshold
Output: Frequent URLs and association rule
begin
Step1: Identify the frequent URLs and their corresponding
queries using UDDAG
Step 2: Calculate the hub and authority weight for each URL
using Hits Algorithm [15]
Step 3: Calculate the time weight for each URL using t-
measure
end
```

The URL clusters are scanned and the bidirectional patterns for the URLs are generated. For example, Consider the URL 7 which is selected by the user 366 and its bidirectional pattern is 13 7 10 23 24 36 44. Here the prefix pattern of the URL 7 is 13 and the suffix pattern is 10 23 24 36 and 44. Association among the prefix and suffix patterns of the URL 7 are

13 7 10 -> 23 7 10   13 7 10 -> 24 7 10   13 7 10 -> 36 7 10
13 7 10 -> 44 7 10   23 7 10 -> 24 7 10   23 7 10 -> 36 7 10
23 7 10 -> 44 7 10   24 7 10 -> 36 7 10   24 7 10 -> 44 7 10
36 7 10 -> 44 7 10

The rules in Table VIII give the recommendation to the user 366 by using the URLs clicked by the users 1038 and 706 on 2006-03-03.

VII. EXPERIMENTAL RESULTS

The algorithms were implemented in JDK .6.0_24. All the experiments are performed in Intel Core i3 processor 2.53 GHz with Windows 7 Home Premium (64-bit) and 4 GB RAM. For the evaluation of time dependent query and URL recommendations, the experimental data are prepared from AOL search engine query log. The log entries from 1-3-2006 to 31-5-2006 are considered (zola.di.unipi.it /smalltext/ datasets. html). The data set contains 1975811 records and 19131507 words in 174 MB, based on our system’s memory and speed the maximum of first 200 pre-processed records are considered. Totally 113 distinct queries are issued by 8 users i.e., 56.5% queries are unique. This analysis shows that 43.5% of the queries are redundant and the users’ intents are same at some point. 90% of the query keywords contain less than 3 terms and the average query length is 2.195. Table IX shows the statistics of our experimental data and Fig.8 depicts the analysis of query length.

TABLE VII
RULES WITH CONFIDENCE

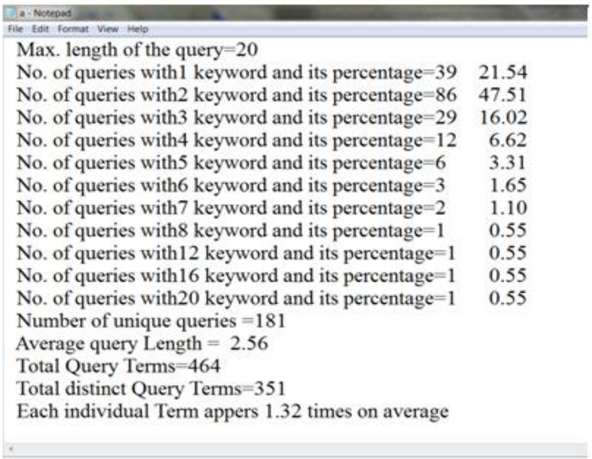
Association Rules	Confidence % (Min. Conf=20%)		
	Prefix Span Basic	PrefixSpan with Hub and Authority	PrefixSpan with t-measure
http://en.wikipedia.org =>>http://www.google.com	30.0	27.702	27.825
http://en.wikipedia.org =>>http://www.gamewinners.com	35.0	32.466	32.484
http://en.wikipedia.org =>>http://www.txlottery.org	55.0	49.527	49.591
http://www.google.com =>>http://www.gamewinners.com	22.5	22.690	22.613
http://www.google.com =>>http://www.pokemon.com	20.0	20.076	20.015
http://www.google.com =>>http://www.txlottery.org	32.5	32.328	32.210
http://www.gamewinners.com =>>http://www.txlottery.org	28.0	27.593	27.608
http://www.pokemon.com =>>http://www.gamewinners.com	22.5	22.594	22.594
http://www.pokemon.com =>>http://www.txlottery.org	22.5	32.159	32.176
http://www.monkees.net =>>http://www.google.com	23.33	25.321	25.185
http://www.monkees.net =>>http://www.gamewinners.com	26.66	29.444	29.155
http://www.monkees.net =>>http://www.pokemon.com	23.33	25.438	25.209
http://www.monkees.net =>>http://www.txlottery.org	40.0	44.210	43.728

TABLE VIII
RULES FOR THE ITEMS 13, 23, 24, 36 AND 44 WITH 7, 10 AND ITS CONFIDENCE

Association Rules	Confidence % (Min. Conf=10%)		
	UDDAG Basic	UDDAG with Hub and authority	UDDAG with t-measure
http://archives.tcm.ie http://en.wikipedia.org http://www.goldenpalace.com =>>http://www.bbc.co.uk http://en.wikipedia.org http://www.goldenpalace.com	11.578	12.450	11.579
http://archives.tcm.ie http://en.wikipedia.org http://www.goldenpalace.com =>>http://www.lutonfc.com http://en.wikipedia.org http://www.goldenpalace.com	10.0	10.871	10.000
http://archives.tcm.ie http://en.wikipedia.org http://www.goldenpalace.com =>>http://www.answers.com http://en.wikipedia.org http://www.goldenpalace.com	13.157	14.029	13.158
http://archives.tcm.ie http://en.wikipedia.org http://www.goldenpalace.com =>>http://www.uefa.com http://en.wikipedia.org http://www.goldenpalace.com	10.0	10.871	10.000
http://www.bbc.co.uk http://en.wikipedia.org http://www.goldenpalace.com =>>http://www.lutonfc.com http://en.wikipedia.org http://www.goldenpalace.com	11.578	12.450	11.579
http://www.bbc.co.uk http://en.wikipedia.org http://www.goldenpalace.com =>>http://www.answers.com http://en.wikipedia.org http://www.goldenpalace.com	11.363	12.235	11.364
http://www.bbc.co.uk http://en.wikipedia.org http://www.goldenpalace.com =>>http://www.uefa.com http://en.wikipedia.org http://www.goldenpalace.com	11.578	12.450	11.579
http://www.lutonfc.com http://en.wikipedia.org http://www.goldenpalace.com =>>http://www.answers.com http://en.wikipedia.org http://www.goldenpalace.com	13.157	14.029	13.158
http://www.lutonfc.com http://en.wikipedia.org http://www.goldenpalace.com =>>http://www.uefa.com http://en.wikipedia.org http://www.goldenpalace.com	10.0	10.871	10.000
http://www.answers.com http://en.wikipedia.org http://www.goldenpalace.com =>>http://www.uefa.com http://en.wikipedia.org http://www.goldenpalace.com	13.157	14.029	13.158

TABLE IX
STATISTICS OF THE EXPERIEMNTAL DATA

Number of Records	200
Number of distinct Users	8
Number of unique queries	113
Number of unique URLs	148
Number of unique query terms	216
Average number of queries per user	14.125
Average number of URLs per user	18.5
Average number of Query terms per user	27
Average length of the query	2.195



```
Max. length of the query=20
No. of queries with1 keyword and its percentage=39   21.54
No. of queries with2 keyword and its percentage=86   47.51
No. of queries with3 keyword and its percentage=29   16.02
No. of queries with4 keyword and its percentage=12    6.62
No. of queries with5 keyword and its percentage=6     3.31
No. of queries with6 keyword and its percentage=3     1.65
No. of queries with7 keyword and its percentage=2     1.10
No. of queries with8 keyword and its percentage=1     0.55
No. of queries with12 keyword and its percentage=1    0.55
No. of queries with16 keyword and its percentage=1    0.55
No. of queries with20 keyword and its percentage=1    0.55
Number of unique queries =181
Average query Length =  2.56
Total Query Terms=464
Total distinct Query Terms=351
Each individual Term appers 1.32 times on average
```

Fig. 8. Analysis of Query Length

Table X depicts number of URLs accessed in the data set month wise. For example, in March 2006, 104 URLs are accessed in 23 days.

TABLE X
MONTH WISE REPORT FOR AOL

Month	Number of days accessed	Number of URLs accessed
2006, March	23	104
2006, April	12	21
2006, May	19	50

The frequently accessed URLs and queries are identified by considering the prefix pattern of the URL. For example, The URL http://en.wikipedia.org (numbered 7 in URL list) was clicked by the user 1038 on 2006-03-03 for the query term ‘shane mcfaul’. The same URL was selected by the user 366 on 2006-03-01 for the query ‘intravenous’. This prefix pattern is considered and the support value of URL 7 is 2. The URL numbers and their support values are listed below;

7&2.0, 51&4.0, 18&5.0, 17&4.0,
25&2.0, 10&2.0, 21&9.0, 22&3.0,
38&2.0, 87&3.0, 41&2.0, 40&2.0,
68&2.0, 18 17&2.0

The frequently occurred queries and URLs are generated and clustered first (Minimum support is 2). Next the combined similarity measure [23] in terms of query and clicked URL similarity is calculated for the frequent items in the cluster. Fig. 9 shows the confidence of rules generated by considering the prefix patterns, prefix patterns with authority weight and prefix patterns with t-measure. The confidence is increased to 6.05% and 5.36% by consider the authority weight and t-measure respectively for the frequent items generated by using the algorithm PrefixSpanBasic.

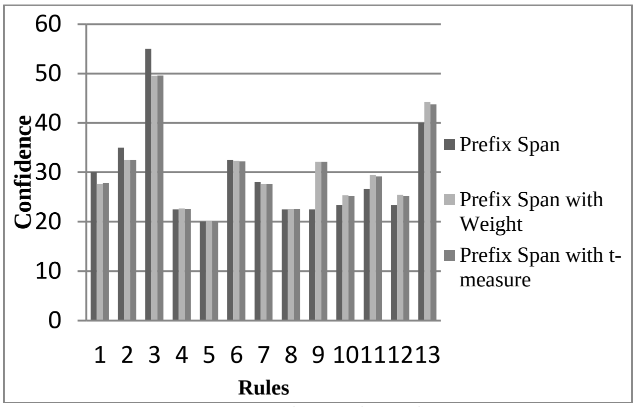


Fig. 9. Confidence for prefix patterns

Table XI shows the precision and recall values for the frequent URLs generated by using PrefixSpanBasic and MHitsPrefixspan.

TABLE XI
PREFIX PATTERNS - PRECISION AND RECALL

PrefixSpanBasic			PrefixSpan with Authority and t-measure		
Frequent URL	Precision	Recall	Frequent URL	Precision	Recall
7	0.143	0.071	7	0.143	0.167
51	0.039	0.143	51	0.039	0.333
18	0.167	0.214	18	0.167	0.5
17	0.235	0.286	17	0.235	0.667
21	0.333	0.5	21	0.238	0.833
87	0.115	0.714	87	0.069	1
25	0.2	0.357			
22	0.364	0.571			
38	0.237	0.643			
10	0.6	0.429			
41	0.268	0.786			
40	0.3	0.857			
68	0.191	0.929			
18, 17	0.402	0.5			

Fig. 10 depicts the precision and recall measures for the frequent URLs which are generated by using PrefixSpanBasic and MHitsPrefixspan.

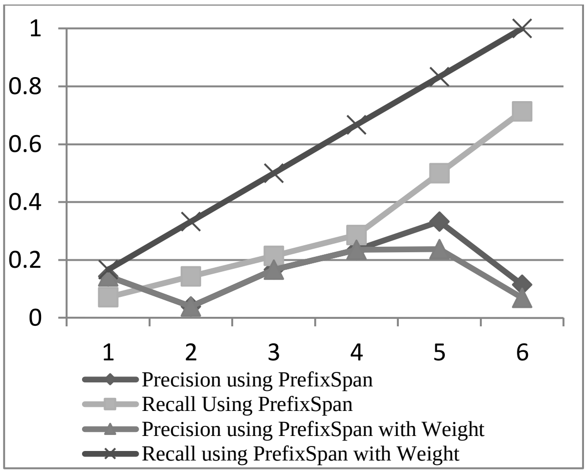


Fig.10. Precision and recall – PrefixSpanBasic & PrefixSpan with weight

Next, f-measure is calculated for the frequent items using precision and recall values and Fig. 11 displays the f-measure value for the frequent URLs 7, 51, 18, 17, 21 and 87 using the techniques PrefixSpanBasic and PrefixSpan with weight. The measurement is high when the authority weight or t-measure is included.

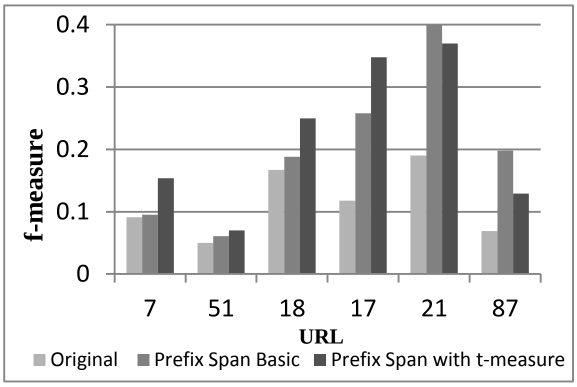


Fig.11. f-measure of PrefixSpanBasic and PrefixSpanBasic with weight

Frequent queries and URLs generated by considering the prefix and suffix pattern provides the following URL combinations and their support value for the URL 7.

13 7 10&19.0 23 7 10&22.0 24 7 10&19.0
36 7 10&25.0 44 7 10&19.0

Here the URLs 13, 23, 24, 36, 44, 7, 95, 10, 123, 124, 125, 145, 146, 147 and 148 are clustered based on time because all were accessed on 2006-03-03. The hub and authority weights are calculated using the hits method [10]. The t-measure is calculated for this cluster which is 0.002. The pattern 13, 7 and 10 occurs 19 times. For the above URL combinations, 10 rules are generated. Fig. 12 shows the confidence value of the rules generated by considering the prefix and suffix patterns, patterns with authority weight and with t-measure. The average confidence is increased by 8.7% when the hub and authority weight is considered for the frequent items generated by using basic UDDAG. Fig. 13 depicts the lift measure for the rules.

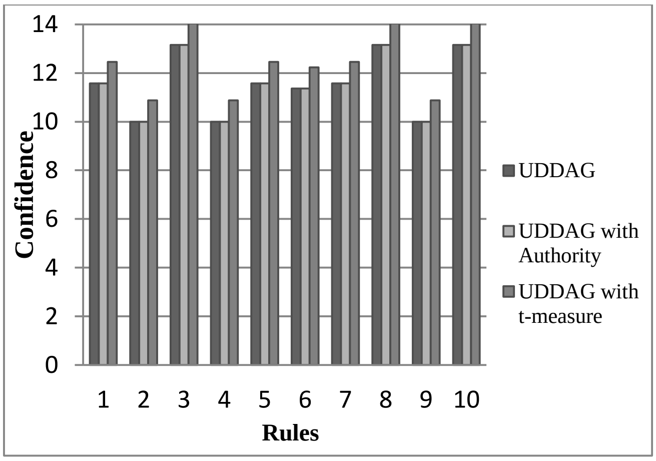


Fig. 12. Confidence for prefix and suffix patterns

Confidence and Lift measures in Fig. 12 and Fig.13 shows that the measurement is high when t-measure is considered along with the prefix and suffix patterns. Table XII shows the precision and recall values for the frequent items generated for the URL 7 using the algorithm MUDDAG. Fig. 14 depicts the precision and recall measures calculated for the frequent URLs retrieved using MUDDAG.

If the user issues the queries ‘shane mcfaul’, ‘intravenous’ and ‘on line casino’ then our system recommends the query ‘liam george’ for the user because there exists a rule between the URLs 13 7 10 and 24 7 10.

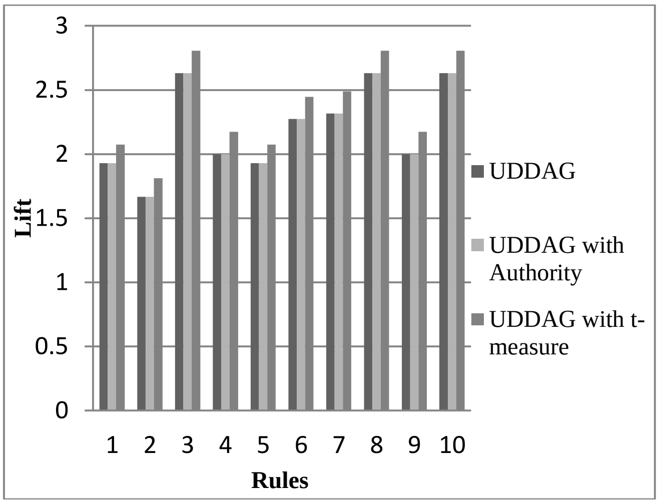


Fig. 13. Lift measure for prefix and suffix patterns

TABLE XII
PREFIX AND SUFFIX PATTERNS - PRECISION AND RECALL

Frequent URL	Precision	Recall
13	0.077	0.143
7	0.286	0.286
10	0.3	0.429
23	0.174	0.571
24	0.208	0.714
36	0.167	0.857
44	0.159	1

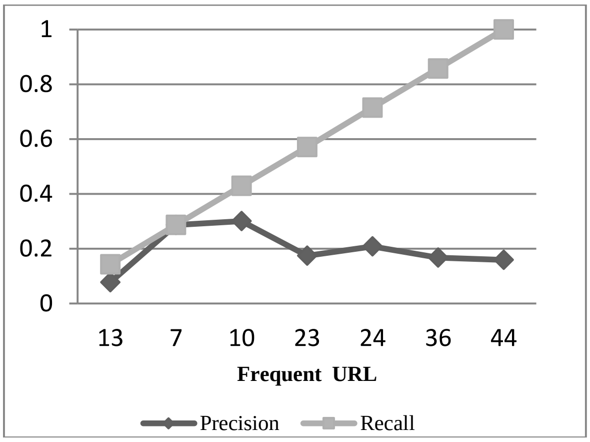


Fig.14. Precision and Recall – MUDDAG

13	1038	shane mcfaul	2006-03-03 17:52:43	2	http://archives.tcm.ie
7	36	intravenous	2006-03-01 17:16:19	3	http://en.wikipedia.org
10	706	on line casino	2006-03-19 14:29:21	1	http://www.goldenpalace.com
23	1038	shane mcfaul	2006-03-03 17:27:32	11	http://www.bbc.co.uk
24	1038	liam george	2006-03-03 17:57:51	1	http://www.lutonfc.com
44	1038	shane mcfaul	2006-03-03 17:30:30	14	http://www.uefa.com
26	227	harrisburg pa hotels	2006-03-08 23:50:49	2	http://www.harrisburgpahotels.worldweb.com

The proposed work provides the query recommendation which is either content based or collaborative based. Content based approach gives the recommendation based on the search histories and navigational behaviour of the user. For example, consider the query ‘1999’ given by the user 1038 on 2006-03-21. Our approach recommends the following queries and URLs to the user 1038 by considering only the search behaviour of prefix patterns of the user 1038.

Recommended Queries:

1999 hyundai accent, 1999 hyundai accent air bag

Recommended URLs:

http://www.internetautoguide.com

http://www.usedpartslive.com, http://www.2carpros.com

While the prefix and suffix patterns are considered, the following recommendations are given to the user 1038.

Recommended Queries:

1999 hyundai accent, 1999 hyundai accent air bag

1999 hyundai accent cascover door

Recommended URLs:

http://www.internetautoguide.com

http://www.usedpartslive.com, http://www.2carpros.com

http://www.autobytel.com, http://www.carsearch.com

Next category of recommendation is Collaborative approach which is based on the preferences of other users. For example, Consider the user 1038 gives the query ‘map’ on 2006-05-06 at 15:32:13. When the prefix patterns are considered, the recommendations are given from the queries of the users 309, 808 and 647.

Recommended Queries:

Maps, map of iraq.com. mapquest com

Recommended URLs:

http://www.mapquest.com, http://www.comcast.net

VIII. CONCLUSION

The proposed algorithm generates the frequently accessed queries and URLs in its first phase by considering the prefix and suffix patterns. In the second phase, the authority weight and the time dependent weight t-measure are calculated for frequently accessed URLs retrieved from the first phase. Based on the authority and time weight, the association rules are generated. Combined similarity measure is calculated for the frequent patterns [23]. In the proposed method, the URLs are recommended to the user by using the URLs clicked by the experts those who have the same intent of the user. Here the queries are also recommended to the user to frame the meaningful and relevant future queries. The ranking of recommended queries and the ontology based concept representation will be concentrated in future.

REFERENCES

[1] Wen Ji-Rong, Jian-Yun Nie, and Hong-Jiang Zhang. "Clustering user queries of a search engine." In Proceedings of the 10th international conference on World Wide Web, pp. 162-168. ACM, 2001.

[2] Joachims Thorsten. "Optimizing search engines using clickthrough data." Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2002.

[3] Silverstein Craig, M. Henzinger, H. Marais and M. Moricz. “Analysis of a very large AltaVista query log. Technical Report” 1998-014, Systems Research Center, Compaq Computer Corporation, 1998.

[4] Sanderson M, “Ambiguous queries: test collections need more sense”. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 499-506). ACM, 2008, July.

[5] Pei Jian, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. "Mining sequential patterns by pattern-growth: The prefixspan approach." Knowledge and Data Engineering, IEEE Transactions on 16, no. 11 (2004): 1424-1440.

[6] Chen Jinlin. "An UpDown Directed Acyclic Graph Approach for Sequential Pattern Mining." Knowledge and Data Engineering, IEEE Transactions on 22.7 (2010): 913-928.

[7] Kang Yangyang, Yu Hong, Li Yu, Jianmin Yao, and Qiaoming Zhu. "Divided Pretreatment to Targets and Intentions for Query Recommendation." In Natural Language Processing and Chinese Computing, pp. 199-212. Springer Berlin Heidelberg, 2012.

[8] Mei Qiaozhu, Dengyong Zhou, and Kenneth Church. "Query suggestion using hitting time." In Proceedings of the 17th ACM conference on Information and knowledge management, pp. 469-478. ACM, 2008.

[9] Limam Lyes, David Coquil, Harald Kosch, and Lionel Brunie. "Extracting user interests from search query logs: A clustering approach." In Database and Expert Systems Applications (DEXA), 2010 Workshop on, pp. 5-9. IEEE, 2010.

[10] Zahera Hamada M., Gamal F. El-Hady, and W. F. El-Wahed. "Query Recommendation for Improving Search Engine Results." International Journal of Information Retrieval Research (IJIRR) 1.1 (2011): 45-52.

[11] Baeza-Yates, Ricardo, Carlos Hurtado, and Marcelo Mendoza. "Query recommendation using query logs in search engines." Current Trends in Database Technology-EDBT 2004 Workshops. Springer Berlin Heidelberg, 2005.

[12] Fu Lin, Dion Hoe-Lian Goh, and Schubert Shou-Boon Foo. "The effect of similarity measures on the quality of query clusters." Journal of information science 30.5 (2004): 396-407.

[13] Neelam Dunhan, A.K.Sharma, “Rank Optimization and Query Recommendation in Search Engines using Web Log Mining Techniques”, Journal of Computing, Vol.2, Issue 12, December 2010.

[14] Bodon Ferenc. "A survey on frequent item set mining." Budapest University of Technology and Economics, Tech. Rep 2006.

[15] Kleinberg Jon M. "Authoritative sources in a hyperlinked environment." Journal of the ACM (JACM) 46.5 (1999): 604-632.

[16] Li Ruirui, Ben Kao, Bin Bi, Reynold Cheng, and Eric Lo. "DQR: a probabilistic approach to diversified query recommendation." In Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 16-25. ACM, 2012.

[17] Wang Wei, Jiong Yang, and Philip S. Yu. "Efficient mining of weighted association rules (WAR)." In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 270-274. ACM, 2000.

[18] Tao Feng, Fionn Murtagh, and Mohsen Farid. "Weighted Association Rule Mining using weighted support and significance framework." In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 661-666. ACM, 2003.

[19] Sun Ken, and Fengshan Bai. "Mining weighted association rules without pre-assigned weights." Knowledge and Data Engineering, IEEE Transactions on 20.4 (2008): 489-495.

[20] R.Umagandhi, Dr.A.V.Senthilkumar, “An Efficient Method to Identify Users and Sessions from Web Logs”, IJARCS, Vol. 3, No. 2, March-April 2012.

[21] Sudhakar, P., G. Poonkuzhali, and R. Kishore Kumar. "Content Based Ranking for Search Engines." Proceedings of the International Multi Conference of Engineers and Computer Scientists. Vol. 1. 2012.

[22] R.Umagandhi, Dr.A.V.Senthilkumar, “Approaches to find URL click count from Search Engine Query Logs”, International Journal of Computer Information Systems, Vol. 4, 2012.

[23] R.Umagandhi, Dr.A.V.Senthilkumar, “Time Independent Query Recommendations from Search Engine Query Logs”, Proceedings of the International Conference on Software Engineering and Mobile Applications, Dec 2012.