An Implementation of an Enhanced Web Graph Search Engine based on User Profiles and Clickthrough Patterns

Rushikesh M. Shete
Assistant Professor
Department of Computer Science & Engineering
DMIETR, Sawangi (Meghe), Wardha
RTMNU, Nagpur

Dhiraj D. Shirbhate
Assistant Professor
Department of Computer Science & Engineering
JDIET, Yawatmal
SGBAU, Amravati, India

ABSTRACT
As the exponential explosion of various contents generated on the Web Recommendation techniques have become increasingly indispensable. Innumerable different kinds of recommendations are made on the Web every day, including movies, music, images, books recommendations, query suggestions, tags recommendations, etc. In this paper, aim is to providing a general framework on user profiles & Clickthrough patterns. Firstly proposing a method which propagates similarities between different nodes i.e. from user profiles and generates recommendations from Clickthrough data. The proposed framework can be utilized in many recommendation tasks on the World Wide Web, including query suggestions, tag recommendations, expert finding, image recommendations etc. The experimental analysis on large data sets will show the promising future of our work.

General Terms
GRms: name given to the system. DRek: Previous implemented work.

Keywords
Recommendations, Query Suggestions, Clickthrough data, User Profiles

1. INTRODUCTION
A key factor for the popularity of today's Web search engines is the friendly user interfaces they provide. With the diverse and explosive growth of Web information, how to organize and utilize the information effectively and efficiently has become more and more critical [1]. This is especially important for Web 2.0 related applications since user-generated information is more freestyle and less structured, which increases the difficulties in mining useful information from these data sources. In order to satisfy the information needs of Web users and improve the user experience in many Web applications, Recommender Systems, have been well studied and widely deployed in industry. In recent research, focus is on how to utilize web as knowledge for decision making.

Web mining is the technique of data mining. In this work the web graphs mining is used. The directed links between pages of the World Wide Web are described by the web graph. A graph, in general, consists of several vertices, some pairs connected by edges. In a directed graph, edges are directed lines or arcs. The web graph is a directed graph, whose vertices correspond to the pages of the WWW, and a directed edge connects page X to page Y if there exists a hyperlink on page X, referring to page Y. The degree distribution of the web graph strongly differs from the degree distribution of the classical random graph model. The web graph is an example of a scale-free network. The web graph is used for computing the Page Rank of the WWW pages. Recommender systems are a subclass of information filtering system that seek to predict the 'rating' or 'preference' that a user would give to an item, such as music, books, or movies or social element (e.g. people or groups) they had not yet considered, using a a model built from the characteristics of an item or the user's social environment [4], [6]. Typically, recommender systems are based on Collaborative Filtering which is a technique that automatically predicts the interest of an active user by collecting rating information from other similar users or items. The underlying assumption of collaborative filtering is that the active user will prefer those items which other similar users prefer. Based on this simple but effective intuition, collaborative filtering has been widely employed in some large, well-known commercial systems, including product recommendation or movie recommendation etc. Typical collaborative filtering algorithms require a user-item rating matrix which contains user-specific rating preferences to infer users' characteristics [7].

2. RELATED WORK
Recommendation on the Web is a general term representing a specific type of information filtering technique that attempts to present information items (queries, movies, images, books, Web pages, etc.) that are likely of interest to the users. In this section, we review several work related to recommendation, including collaborative filtering, query suggestion techniques, image recommendation methods, and Clickthrough data analysis.

2.1 Collaborative Filtering
Neighborhood-based and model-based are two types of collaborative filtering [5]. The most analyzed examples of neighborhood-based collaborative filtering include user-based approaches and item-based approaches. User-based approaches predict the ratings of active users based on the ratings of their similar users, and item-based approaches predict the ratings of active users based on the computed information of items similar to those chosen by the active user. Recently, several matrix factorization methods have been proposed for collaborative filtering. These methods all focus on fitting the user-item rating matrix using low-rank approximations, and use it to make further predictions [7].

2.2 Query Suggestion

In order to recommend relevant queries to Web users, a valuable technique, query suggestion, has been employed by some prominent commercial search engines, such as Yahoo!, Live Search, Ask, and Google.

The goal of query suggestion is similar to that of query expansion, query substitution, and query refinement which all focus on understanding users' search intentions and improving the queries submitted by users. Query suggestion is closely related to query expansion or query substitution, which extends the original query with new search terms to narrow down the scope of the search [4]. But different from query expansion, query suggestion aims to suggest full queries that have been formulated by previous users so that query integrity and coherence are preserved in the suggested queries. Query refinement is another closely related notion, since the objective of query refinement is interactively recommending new queries related to a particular query.

2.3 Clickthrough Data Analyses

In the field of clickthrough data analysis is commonly used for optimizing Web search results or rankings. Web search logs are utilized to effectively organize the clusters of search results by 1) learning "interesting aspects" of a topic and 2) generating more meaningful cluster labels[5],[6]. A ranking function is learned from the implicit feedback extracted from search engine clickthrough data to provide personalized search results for users. Besides ranking, clickthrough data is also well studied in the query clustering problem. Query clustering is a process used to discover frequently asked questions or most popular topics on a search engine. This process is crucial for search engines based on question answering. A typical relationship can be learning from clickthrough data is that "BMW" is a child of "car." The method proposed can extract attributes such as "capital city" and "President" for the class "Country," or "cost," "manufacturer" and "side effects" for the class "Drug." The method initially relies on a small set of linguistically motivated extraction patterns applied to each entry from the query logs, and then employs a series of Web-based precision-enhancement filters to refine and rank the candidate attributes [3].

3. ANALYSIS OF PROBLEM

Typical collaborative filtering algorithms require a user-item rating matrix which contains user-specific rating preferences to infer users' characteristics. However, in most of the cases, rating data are always unavailable since information on the Web is more diverse and less structured. If a general graph recommendation algorithm is designed, many recommendation problems on the Web can be solved. For recommendations on the Web several challenges are to be faced while designing framework that need to be addressed.

3.1 Long Query Web Searches

The first case is it is not easy to recommend latent semantically relevant results to users. Take Query Suggestion as an example; there are several outstanding issues that can potentially degrade the quality of the recommendations, which merit investigation. The first one is the ambiguity which commonly exists in the natural language. Queries containing ambiguous terms may confuse the algorithms which do not satisfy the information needs of users. Another consideration, as reported is that users tend to submit short queries consisting of only one or two terms under most circumstances, and short queries are more likely to be ambiguous.

3.2 Personalization in Web Searches:

The second case is the personalization feature. Personalization is needed for many scenarios where different users have different information needs. This problem is associated with presentation of information and type of information. Since it depends on user's interest while interacting with the web.

3.3 Time to Show Recommendations

The third case is that it is time consuming and inefficient to design different recommendation algorithms for different recommendation tasks. Actually, most of these recommendation problems have some common features, where a general framework is needed to unify the recommendation tasks on the Web.

4. OBJECTIVES

In this implemented work, aim is to solve the problems analyzed above; a general framework is designed for the recommendations on the Web. This framework is built upon the user profiles and he Clickthrough data patterns, and has several objectives.

1. It is a general method, which can be utilized to many recommendation tasks on the Web.
2. It provides latent semantically relevant results to the original information need.
3. It provides a long query to the user within short time.
4. It provides the specific recommendations to the user.

5. IMPLEMENTED FRAMEWORK

5.1 System architecture

Query Suggestion is a technique widely employed by commercial search engines to provide related queries to users' information need. In this section, we demonstrate how our method can benefit the query suggestion, and how to mine latent semantically similar queries based on the users' information need. Clickthrough data record the activities of Web users, which reflect their interests and the latent semantic relationships between users and queries, as well as queries and clicked Web documents as shown in figure 1 below.

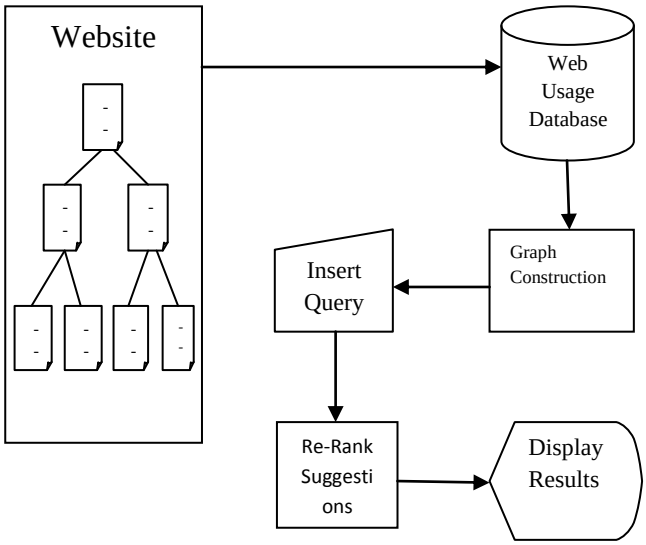

Fig-1: System Architecture

The framework is composed of the following main components:

A web graph formation: Neo4J is used for the graph database. From existing database web graph is formed with URLs & links as nodes of graph. All the URLs & links are ranked.

Re-rank method: When query is fired, suggestions for the query are re-ranked. Highest value of rank gives the gives quality information to user.

Display Results: After re-ranking, suggestions are displayed in descending order for the fired query.

Recommendations: When user again wants to search for different query he / she will get the recommendations from the framework.

The empirical analysis on several large scale data sets (AOL Clickthrough data and Flicker image tags data) shows that our proposed framework is effective and efficient for generating high-quality recommendations. Flowchart in Fig. 2 will show the execution of the process of this work.

5.2 Query Suggestions using Re-rank method:

Query Suggestion is a technique widely employed by commercial search engines to provide related queries to users' information need. In this section, how this method can benefit the query suggestion is demonstrated, and how to mine latent semantically similar queries based on the users' information need.

When user enters query to the search engine, it suggests the query as per the requirement. When query is suggested by the search engine user can select the query and can surf it. Query suggestion utilizes the query logs from user profiles. From user profiles required information is sorted from previous related queries. Query can be sometimes different from the user's expectations. Query suggestion is necessary because from clicked data from previous user it does not give critical information. Effective query suggestion need the users query intent and then suggests query. It may help user to retrieve useful information. The aim of query suggestion is to use past information from previous user profiles.

Data Collection

A query suggestion graph is constructed based on the clickthrough data of user profiles. This data set is gathered from currently active user's & from past user's user profiles. Clickthrough data record the activities of Web users, which reflect their interests and the latent semantic relationships between users and queries as well as queries and clicked Web documents. Each line of clickthrough data contains the following information: a user ID (u), a query (q) issued by the user, a URL (l) on which the user clicked, the rank (r) of that URL, and the time (t) at which the query was submitted for search. Thus, the clickthrough data can be represented by a set of quintuples < u, q, l, r, t >. From a statistical point of view, the query word set corresponding to a number of Web pages contains human knowledge on how the pages are related to their issued queries. Thus, in this work, the relationships of queries and Web pages are utilized for the construction of the bipartite graph containing two types of vertices < q, l, r >. The information regarding user ID and calendar time is ignored.

This data set is the raw data recorded by the search engine, and contains a lot of noise which will potentially affect the effectiveness of our query suggestion algorithm. Hence, a similar method employed in [14] to clean up the raw data is conducted. The data is filtered by only keeping those frequent, well formatted, English queries (queries which only contain characters "a," "b," . . . , "z," and space). After cleaning and removing duplicates, unique queries and unique URLs are obtained in data collection.

Hereafter, all the URLs & links are given a rank. In this work pair wise relationship for ranking query is used. For the calculation of rank each word in a query is matched with the words of URLs & links. Each time when going through each node of existing database, query word is matched with the word of URLs & links score is incremented by one. Highest the rank data quality is high and vice versa. Ranking method becomes easy for user to search for required information.

Re-ranking

Following is pseudo code for query suggestion with re-ranking. It is known as re-ranking because it ranks the already ranked query with different parameters. Web graph is formed by using links and URL. Graph consists of set of query and set of URL's. When travelled through the graph first root node is visited and the entered query by user is compared with the matching data of the node. If it matches to the node value that link is suggested as suggestion to the user.

The bipartite graph is formed by using query and URL as nodes and the connecting links to these nodes are weighted by rank value.

Pseudo code of Query Suggestion:

1. Initialize the list of URLs & links.
2. Get the complete existing database from user profiles.
3. Create bipartite graph.
    a. Collect a set Q of queries.
    b. Collect a set U of URLs.
    c. For each of the n unique query, create node q_i.
    d. For each of the m unique URL, create node u_j.
    e. If query q_i appeared with URL u_j, then place an edge between the q_i and node u_j.
4. Assign rank k to each URL & link of database by incrementing it by 1 i.e. k=k+1 for every matching word.
5. For query q search web graph.
6. Match each word of the query with title, contains & URLs of graph database by comparing word node by node.

    If (q = link || URL)

    Then j=k+1; // here k is the actual rank of URL or link calculated in step 4
7. Output the top Q queries with descending order of re-rank i.e. j as the suggestions.

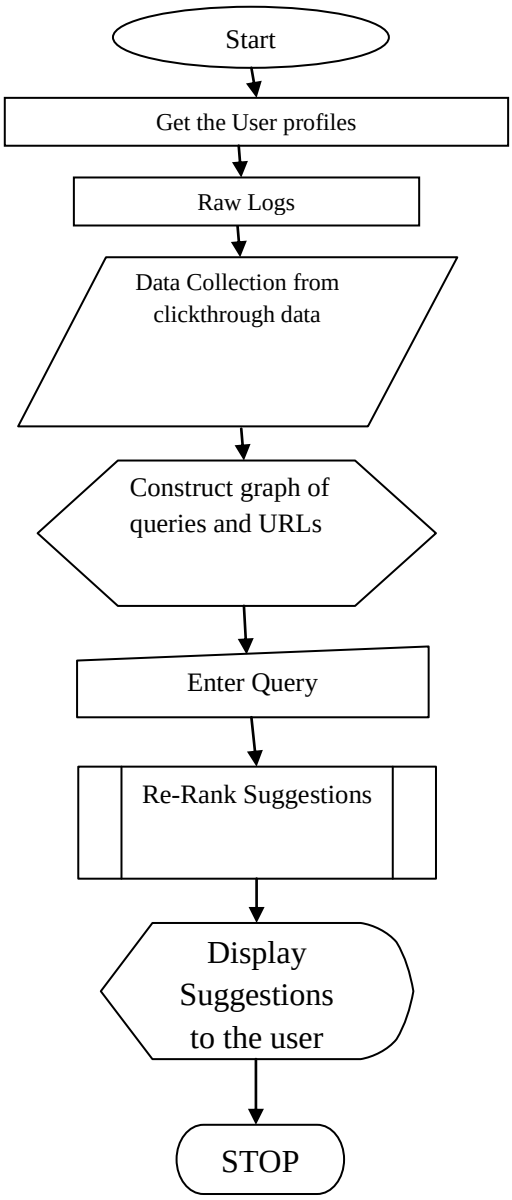

Fig-2: Flowchart Showing Basic Steps

5.3Results for Query suggestion

The framework is tested to find query suggestions using different test queries. The results with 5 query suggestions are presented in table 1 below. As in [1], [5] short queries give ambiguous & incorrect suggestions, in GRms short queries are giving fruitful information also suggestions are unambiguous. Here in table 1, along with the suggestions rank is also shown. This rank represents the score i.e. quality of information in suggested link. Higher ranked suggestion has prime knowledge about users query.

As in [3], rank improvement gives caliber query suggestions. In our method we are using re-rank method, which calculates rank or score by comparing query word with link suggested, information present in that link and in URL. Based on these three parameters suggestions are re-ranked. For re-ranking clickthrough data is used. Raw logs are taken from web and compared with the query inserted by the user. If matching words are found web graph is formed. For this graph database is used.

GRms system is also solving the problem of less number of suggestions for query. In table 2, number of suggestions for various query fired. If user is getting plenty of suggestions he / she will be able to collect valuable information from the suggestions.

In this framework, recommendations are also shown to the user. If user is unable to write expected query, recommendations are suggested for writing query. When user will type any letter, that letter is compared with the graph database and if letters are matched user will get recommendations for the query. This personalizes user's requirement.

5.4 Clickthrough Data

Clickthrough information serves to perceive the examples of information that whether it is data, picture and so forth this methodology are mining the information from web. In web graph all the information or data is put away as hubs in the form of nodes. Every node will continue to the obliged data of the client. In web graph every node is the connection to the embedded inquiry. Inquiry or query can be contrasted and the connection at the hub, on the off chance that it discovers the oblige data it might be proposed to the client with the rank. [3], [4].

Clickthrough data includes different steps, data collection & data cleaning. From user profiles raw logs are maintained. These raw logs contain all the information related to user entered query. From this raw log only quality data is extracted. This extracted information contains all the information related to the users wishing information. After cleaning data all the data is collected and database is created. This step may reduce the size of data to the great extent.

After data collection each link to the required information will be provided the rank. This rank can be a re-rank. Re-ranking is done on the basis of priority, which will be decided by the entered query. If the link in the graph is having quality information the rank of the node will be the highest. If the information is not up to the mark, rank will be lowest.

Clickthrough data in search engines can be thought of as triplets (q, r, c) consisting of the query q, the ranking r presented to the user, and the set c of links the user clicked on. In this method, user asks the query, receive the suggestions with the rank. The next step is, user will select appropriate suggestions according to the requirement.

Clearly, users do not click on links at random, but make a (somewhat) informed choice. While clickthrough data is typically noisy and clicks are not "perfect" relevance judgments, the clicks are likely to convey some information. The key question is: how can this information be extracted or recorded?

Clickthrough data can be recorded with little overhead and without compromising the functionality and usefulness of the search engine. In particular, compared to explicit user feedback, it does not add any overhead for the user. The query q and the returned ranking r can easily be recorded whenever the resulting ranking is displayed to the user. For recording the clicks, a simple proxy system can keep a log file.

Table 1: Results of Query Fired

Testing Queries	**5 Suggestions with rank**									
Taj Mahal	Taj Mahal, Uttar Pradesh, archaeological survey of India	154	Taj Mahal Unesco world heritage centre	139	sand sculpture of Taj Mahal draws crowds	137	local business results for Taj Mahal	134	the Taj Mahal	76
I-Phone mobile	Apple mobile phones in India price list (updated April 2014)	299	iPhone – mobile	224	apple iphone - mobile phone prices	207	all apple phones - GSM arena	171	iPhone 5 - t-mobile	160

6. EXPERIMENTAL RESULTS AND DISCUSSION

6.1 Results for Query suggestion

Implementing system is not only the valuable work of the developer. It becomes successful when optimized results are obtained. Most important part of the framework is query suggestions. This was the main focus of at the time of implementation.

The framework is tested to find query suggestions using different test queries. The results with 5 query suggestions are presented in Table 1 above. As in [1], [5] short queries give ambiguous & incorrect suggestions, in this system short queries are giving fruitful information also suggestions are unambiguous. Here in Table 1, along with the suggestions rank is also shown for 5 different queries. This rank represents the score i.e. quality of information in suggested link. Higher ranked suggestion has prime knowledge about user's query.

As in [4], rank improvement gives caliber query suggestions. In this method re-ranking method is used, which calculates rank or score by comparing query word with link suggested, information present in that link and in URL. Based on these three parameters suggestions are re-ranked. For re-ranking clickthrough data is used from user profiles. Re-ranking method helps user to select proper suggestion from the list of output. This ultimately saves time of user at surfing time.

This system is also solving the problem of less number of suggestions for query. In Table 2, number of suggestions for various query fired are given. If user is getting plenty of suggestions he / she will be able to collect valuable information from the suggestions.

In this framework, recommendations are also shown to the user. If user is unable to write expected query, recommendations are suggested for writing query. This is personalization feature, which predicts user interest and shows recommendations to the user. When user will type any letter, that letter is compared with the graph database and if letters are matched user will get recommendations for the query. This personalizes user's requirement.

Query suggestions results are compared with the DRek algorithm proposed in [5], and with commercial search engine L*ive Search* in table 2. Numbers of query suggested by implemented algorithm are more as compared with other works. As shown in table 2 below.

Table 2: No. of suggestions for query inserted

Query fired by user	**GRms**	**DRek**	**Live Search**
Taj mahal	46	23	18
I-phone mobile	58	30	20
Pizza	52	28	17
Car	54	31	22
Laptop	63	35	20
Camera	48	26	14
Apple	43	28	17
Medicine	50	34	23
Bike	46	29	18
Flowers	42	21	15

From table 2 it can b said that no. of suggestions shown by GRms method is higher than compared with other systems. All these suggestions are re-ranked. It gives vision to user for which query he / she should go. The data values in the table are based on Clickthrough data. These values are observed on particular date & time. These values may vary at different instance of time. Graph below will show the detailed comparison.

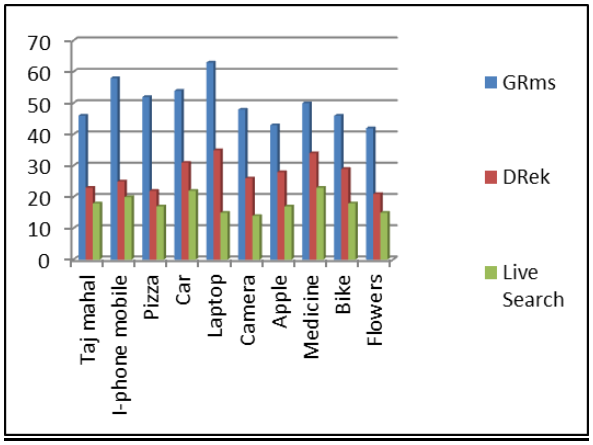

Fig 3: Comparison of No. recommendations returned by GRms and from other systems.

6.2 Time required for display of suggestions

One of the analysis problems is time. Time is also important factor in recommendation for query. Generally users input short queries to the search engine, which gives ambiguous results. In GRms system unambiguous results are obtained for short queries. When user fire short queries it gets the results in specific time i.e. seconds or minuets. In this system time required for displaying suggestions for short query, long query & for very large query is shown in table 2.

Table 3 is drawn from different query inserted to the system GRms. Firstly, short query for example: taj mahal was given as input, then for query size having 15-20 letters was given as input and lastly query of 25 and above letters was inserted. From the input given, time required for showing suggestions was measured as shown in table below. This process was repeated for 10 times with different queries.

Table 3: Time required time for different query size.

Query size	GRms	DRek
Short (10 -12 letters)	5 seconds	12 seconds
Long (15-25 Letters)	6 seconds	15 seconds
Very Long (25-40 letters)	10 seconds	More than 20 seconds

Note: Above data is based on 10 times testing.

From the above table it can be easily understood that GRms is taking less time compared to DRek system.

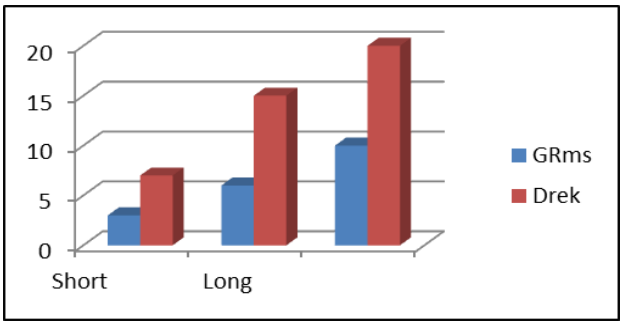

Fig 4: Comparison of time required to display recommendations of different query size.

6.3 Comparison of Results with other work

In this section the comparison of this work is explained with existing work. For this various terms are considered.

Term1: Ambiguity

Recommendations are ambiguous, which do not satisfy information needs in previous works compare to this work gives unambiguous recommendations as web graphs are mined to search the query fired by user. This gives required information to the user.

Term2: Query

In previous work, Short queries submitted didn't give the specific recommendations to the user as in [2], D Beeferman et al proposed query clustering based on distance notion which generates sparse query logs. In this work, for any size of query submitted, it gives caliber suggestions to the user. Also re-ranking method assign score to the suggestion, which helps user to select suggestion for searching.

Term3: Recommendations

In [1], recommendations are formed by using global analysis, which gives worse results to the user. From this framework personalization feature is used to show recommendations, which works according to the user's interest.

7. CONCLUSIONS

In this work, a novel framework for recommendations can be generated on large scale Web graphs using user profiles and Clickthrough data. From the comparative tables 2 & 3 and graphs shown in figure 3 & 4 it is observed that the time taken for query evaluation is less in GRms compared to DRek system by 11.6%. Recommendations are unambiguous as web graphs are mined to search the query fired by user. This gives required information to the user. Hence, in system the experimental analysis on several large scale Web data sources shows the promising future of this approach by developing new algorithms. This model in general can be applied to more complicated graphs.

Some limitations are still there; in future this can be removed or updated. For future more prominent work can be done on Query writing. So that the much corrected suggestions will be obtained.

8. REFERENCES

[1] B.J. Jansen, A. Spink, J. Bateman, and T. Saracevic, "Real Life Information Retrieval: A Study of User Queries on the Web," ACM SIGIR Forum, vol. 32, no. 1, pp. 5-17, 1998.

[2] D. Beeferman and A. Berger, "Agglomerative Clustering of a Search Engine Query Log," KDD '00: Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 407-416, 2000.

[3] Ch. Nagini, M. Shrinivasa Rao, R. V. Krishnaiah, "A General Framework for Recommendations on the Web", IJARCSSE Volume 3, Issue 5, May 2013.

[4] D. Shen, M. Qin, W. Chen, Q. Yang, and Z. Chen, "Mining Web Query Hierarchies from Clickthrough Data," AAAI '07: Proc. 22nd Nat'l Conf. Artificial Intelligence, pp. 341-346, 2007.

[5] E. Agichtein, E. Brill, and S. Dumais, "Improving Web Search Ranking by Incorporating User Behavior Information," SIGIR '07: Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 19-26, 2006.

[6] Hao Ma, Irwin King, Michael Rung-Tsong Lyu, "Mining Web Graphs for Recommendations" IEEE Transactions on Knowledge Data Engineering, vol 24 June 2012.

[7] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li, "Context-Aware Query Suggestion by Mining Click-Through and Session Data," KDD '08: Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 875-883, 2008.

[8] J.-T. Sun, D. Shen, H.-J. Zeng, Q. Yang, Y. Lu, and Z. Chen, "Web- Page Summarization Using Clickthrough Data," SIGIR '05: Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 194-201, 2005.

[9] R. Baeza-Yates, C. Hurtado, and M. Mendoza. *Query Recommendation Using Query Logs in Search Engines*, volume 3268/2004 of *Lecture Notes in Computer Science*, pages 588–596. Springer Berlin / Heidelberg, November 2004.

[10] R. Jones, B. Rey, O. Madani, and W. Greiner. "Generating query substitutions". In Proc. of WWW, pages 387–396, Edinburgh, Scotland, 2006.

[11] Raymond Kosala, Hendrik Blockeel, "Web Mining Research: A Survey", ACM SIKDD Explorations, Volume 2, Issue 1, July 2000.

[12] Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In *CIKM '08: Proceeding of the 17th ACMconference on Information and knowledge management*, pages 469–478, 2008.

[13] Rushikesh M. Shete, Prof. V. S. Gulhane, "An Enhanced Web Graph Search Engine Based on User Profiles and Clickthrough Patterns", IJERT, ISSN: 2278-0181, Vol. 2 Issue 12, December – 2013

[14] X. Wang and C. Zhai. "Learn from web search logs to organize search results". In Proc. of SIGIR, pages 87–94, Amsterdam, The Netherlands, 2007.