A novel approach for Query Recommendation Via query logs

Rachna Chaudhary, Nikita Taneja

Abstract— Query recommendation is an essential part of modern search engines. Recently, search engines become more critical for finding information over the World Wide Web where web content growing fast, the user's satisfaction of search engine results is decreased. Query Logs are important information repositories, which record user activities on the search results. The mining of these logs can improve the performance of search engines .The technology for enabling query recommendations is query-log mining, which is used to leverage information concerning how people make the use of search engines, and how they rephrase their queries while they looking for information. The proposed system based on learning from query logs predicts user information needs. To carry out the required task, the approach first mines the query logs. Meanwhile, query similarity between the pair wise queries is to be calculated which is based on query contents and their clicked URLs to perform query clustering. Most favored queries are discovered within every query cluster. The proposed result optimization system also presents a query recommendation scheme towards better information retrieval to enhance the search engine efficiency and effectiveness to a large scale.

Index Terms— Query recommendation, Query Log, Search Engine, Web, and Query Clustering, Query Similarity, Information Retreaival.

1 Introduction

Query recommendations are a module of modern search engines. It is a technique [7] that provides better queries to help users to get the needed documents when the original query submitted by user may be insufficient or imprecise to retrieve those. It serves several purposes: correcting [10] possible spelling mistakes, guiding users through their information-seeking tasks, allowing them to locate information more easily, and helping them to discover additional concepts related to what they are looking for. A key technology for enabling query recommendations is query-log mining, which is used to leverage information about how people use search engines, and how they rephrase their queries when they are looking for information.

With the increase of size [11] and popularity of the World Wide Web, many users find it's difficult to obtain the desired information, even though they use most efficient search engines e.g. Google, yahoo. In spite of the recent Advances in the Web search engine technologies; there are still many situations in which the user is presented with non relevant search results. One of the major reasons for this difficulty [8] is that Web search engines a lot have difficulties in forming a concise and precise representation of the user's information need. Most Web search engine users are not well trained in organizing and formulating their input queries, which the search engine relies on to find the desired search results. On the other hand, users are often not clear about the exact terms that best represent their specific information needs. In the worst case, users are still not clear of what exactly their specific information need is. For example, [6] if the user searches for Madonna in Yahoo! search engine the following related queries are presented: Madonna lyrics, Madonna pictures, Madonna confessions on a dance floor, Madonna biography, and Madonna university. Though, we can imagine, there are a good number of other queries related to Madonna but most likely not having the term Madonna explicitly in their term vectors. Given this problem, the method to retrieve semantically related queries is becoming an increasingly important research topic that attracts considerable attention.

A novel approach for query recommendation is proposed in this paper, which attempts to optimize the search engines result. The approach also recommends the user with a set of similar and most popular user queries so as to make his [8] search more efficient. To carry out the required task, the approach pre-mines the query logs to retrieve the potential clusters of queries and then finds the most popular queries in each cluster.

Apart from Section 1, the rest of the paper is organized as follows. Current research that has been carried out in this area is described Section 2. Afterwards Section 3 presents a novel architecture of proposed work based on pre-mining the query logs Section 4 shows the performance of proposed work with example scenario and the last section concludes the paper.

2 Related Work

The notion of query recommendation has been a subject of interest since many years. A number of researchers have discussed the problem of finding relevant search results from the search engines.

Relevant query recommendation research is mainly based on previous query log of the search engine, which contains the history of submitted query and the user selected URLs. Beeferman and Berger [1] exploited "click through data" in clustering URLs and queries using graph-based iterative clustering technique. Wen et al. [2] used a similar method to cluster queries according to user logs. Both of their algorithms are difficult to deal with in practice due to query log sparseness. That is to say, only a part of popular queries have sufficient log information for mining their common clicked URLs while distance matrices between most queries from real query logs are very sparse. As a result, many queries with semantic similarity might appear orthogonal in such matrices.

Fonseca et al [4] showed a method to discover related queries based on association rules. The query log is viewed as a set of

transactions. However, the fact that similar queries are submitted by different users in most of case, will also lead to sparseness problem. This is because the support of a rule increases only if its queries appear in the same query session, and thus they must be submitted by the same user.
Query expansion [2, 3] is also adopted by search engines to recommend related queries. Its idea is to reformulate the query such that it gets closer to the term weight vector space of the documents the user is looking for. This approach aims at construction of queries rather than recommend previous registered queries in real log
However, a [8] critical look at the available literature indicates that from very beginning, search engines are using some kind of optimization on their search results but they are not much beneficial due to the problems of finding the required information within search results. Hence, a mechanism needs to be introduced gives prime importance to the information needs of users. Query log that keeps record of user queries on the basis of occurrence of query in the query cluster which is formed by clustering similar queries on the basis of keywords and clicked URLs is proposed and optimizes the rank values of returned web pages [8] according to the favored query finder related to his search and returning the desired relevant pages in the top of the search result list.

3 Proposed Work

The proposed optimization system (Fig. 1) lying on learning from [8] historical query logs is proposed to calculate user's information requirements in a better way. The proposed system works as follow. The prime feature of the system is to perform query clustering by finding the query similarity between the two queries, based on user query keywords and clicked URLs. After that, clusters are generated with the help of query clustering tool. This tool is used to cluster user queries using query logs built by search engines which in result produce query clusters. Once [8] query clusters are formed, next step is to find a set of favored queries from each cluster. Favored query are those that occupy a major portion of the whole search request in a cluster. Once favored queries from their query clusters are identified, next step is to optimize the user search by recommending him with most favored query related to his search and returning the desired relevant pages in the [8] top of the search result list.
The proposed architecture of result optimization system (fig 1) consists of the following functional components.
1. Query Log
2. Query Similarity
3. Query Clustering Tool
4. Favored Query finder
5. Query Recommender

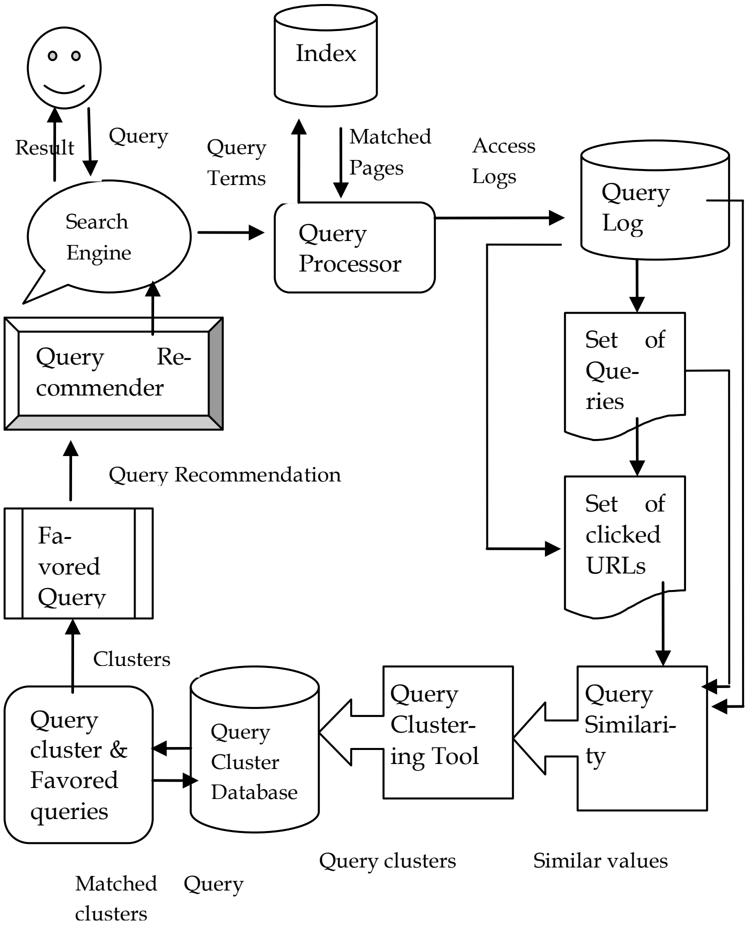

Fig 1: Architecture of result optimization system

When [8] user submits a query on the search engine interface, the query processor component matches the query terms with the index repository of the search engine and returns a list of matched documents in response. On the back end, user browsing behavior including the submitted queries and clicked URLs get stored in the logs and are analyzed continuously by the Query similarity module, the output of which is forwarded to the Query Clustering Tool to generate groups of queries based on their similarities. Query clustering tool produces query clusters and then with the help of favored query finder it extracts most popular queries from each cluster and stores them for future reference and at last query recommender documents are extracted from the favored query finder corresponding to favored query and similar queries and get stored in the interface of search engine which produces final results to user.
The detailed working of these modules is explained in the next subsections.

3.1 Query Logs

Query log [7] has been a popular data source for query recommendation. Query logs are repositories that record all the interactions of users with a search engine for gaining insight

- *Miss Rachna Chaudhary is currently pursuing masters degree program in Computer engineering in Manav Rachna college of engg., Faridabad, Haryana. E-mail: rachuchaudhary04@gmail.com.*
- *Mrs Nikiti Taneja is a Asst. Proffessor in CSE Department in Manav Rachna College of engg. Faridabad , Haryana, E-mail: nikita.mrce@mrei.ac.in*

into how a search engine is used and what the users' interests are. Since they form a complete record of what users searched for in a given time frame. Depending on the specifics of how the data is collected, typical [9] logs of search engines include the following entries: (A) User IDs, (B) Query q issued by the user, (C) URL u selected by the user (D) Rank r of the URL u clicked for the query q and (E) Time t at which the query has been submitted for search A sample query log is shown in Table 1.

TABLE 1: Example Illustration of Query Log

ID	Query	Clicked URL	Rank	Time
Admin	Data Mining	www.dming.com	6	00:01:10
Admin	Data Ware housing	www.dming.com	5	00:01:10
Admin	Data Mining	www.google.com	5	00:01:16
Admin	Data Warehousing	www.datawarehousin om	7	00:01: 16
Admin	Search Engine	www.dming.com	6	00:01: 16
Admin	Web Crawler	www.crawler.com	5	00:01:16

In various studies, researchers and search engine operators have used information from query logs to study about the search process and to get better search engines from early studies of the logs created by users.

Our method considers only [5] queries that appear in the query-log. A single query may be submitted to the search engine several times, and every submission of the query induces a similar query session. A simple notion of query session which consists of a query, along with the URLs clicked is as follow:
Query Session= (Query (Clicked URL))

3.2 Query Similarity

The next step in proposed system is computing the query similarity. It is an important crisis and has a wide range of applications in Information Retrieval in query recommendation. Previous work on query similarity aims to give a single similarity measure without knowing the information that queries are indefinite and generally have several search intents. By introducing search intents into the calculation of query similarity, we can get more exact and also useful similarity measures based on queries and clicked URLs too.

This module is used for finding query similarity using query logs built by search engines and for this it assigns query log entries to query similarity, which produces similar values based on keyword as well as URLs as shown in Fig.2. It works on the following principles.

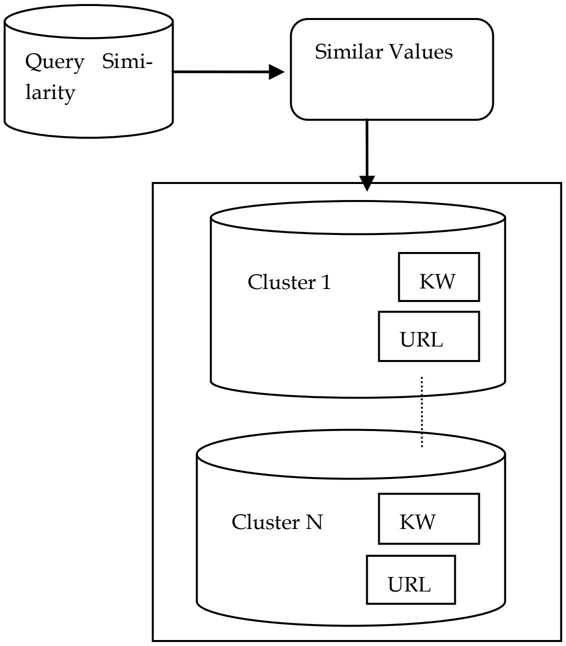

Fig 2: Query Similarity KW (Keyword)

Principle 1 (using query contents): If two queries contain same or similar terms, they are supposed to denote the same or similar information needs. Content based similarity function is defined as follows:

$$Sim\ (p,q)=KW(p,q)/max(kw(p),kw(q)) \qquad (a)$$

Where kW (p) and kW (q) are the sets of keywords in the queries p and q respectively, KW (p, q) is the set of common keywords in two queries.

It is estimated that longer the query, the more reliable it is. However, [2] as most of the user queries are short, this principle alone is not sufficient. Therefore, the second principle is used in combination as a complement.

Principle 2 (using document clicks): Two queries are considered similar if they lead to the selection of same documents (document clicks). Document selections are comparable to user relevance feedback in the traditional IR environment; except that document clicks indicate implicit relevance and not always valid relevance judgments .User feedback based similarity function is defined as follows:

$$Sim\ (p,\ q) =RD\ (p,\ q)/max\ (rd\ (p),\ rd\ (q)) \qquad (b)$$

Where rd (p) and rd (q) are the number of referred documents for two queries p and q respectively, RD (p, q) is the number of document clicks in common.

Combination of Multiple Measures
Both principles have been considered important to determine the similarity of queries and thus, any one of them cannot be ignored; therefore, a combined measure has been defined to

take advantage of both principles as is given below:

Sim (p.q) = α.Sim (p, q) +β.Sim (p, q) (c)

Where α and β are constants with 0<=α (and β) <=1 and α + β=1
There is a question concerning the setting of these parameters and that can be decided by the specialist of concerned domain. In the present implementation, [1] these parameters are taken to be 0.5 each.

3.3 Query Clustering Tool

In support of the clustering process, this tool is used to cluster user queries using query clustering tool built by search engines and for this it assigns query cluster database log entries, which in result produces matched query clusters and favored queries as shown in Fig.3.

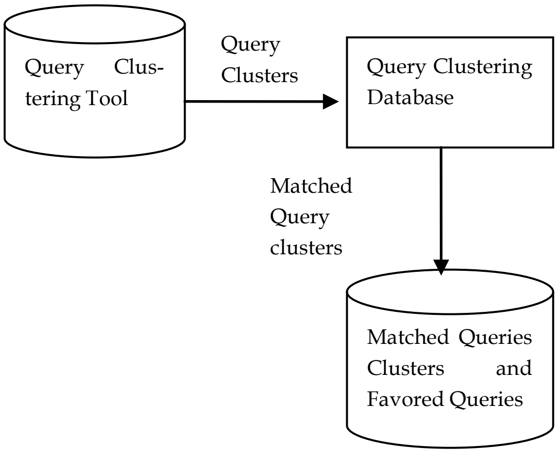
Query Clustering Tool
Query Clusters
Query Clustering Database
Matched Query clusters
Matched Queries Clusters and Favored Queries

FIG 3: QUERY CLUSTERING TOOL

An important component in our proposed work is the concept of clustering queries in user logs. The query clustering is a preprocessing phase and it can be conducted at periodical and regular intervals. Even though the need for query clustering is somewhat new, there have been general studies on document clustering, which are similar to query clustering. However, it is not reasonable to easily apply any document clustering algorithms to queries due to their own characteristics. It is usually observed that queries submitted to the search engines typically are very short, so the clustering algorithm should be suitable for short texts. Additionally query logs are usually very large, the method should be able of handling a large data set in reasonable time and space constraints. Furthermore, due to the fact that the log data changes daily, the method should also be incremental. In view of the above requirements, an adaptive and autonomous clustering algorithm is proposed.
This module is based on the simple perspective: initially, [9] all queries are considered to be unassigned to each cluster. Each query is examined next to all other queries whether classified or unclassified by using (3). If the value of similarity turns out to be greater than the pre-specified threshold value (T), then the queries are grouped into the similar cluster. The similar process is repeated until every query gets classified to any one of the clusters. The method returns overlapped clusters i.e. a particular query may span various clusters. All the queries should be extracted from query logs first and subsequently be stored in the database for the clustering process known as query clustering database. The clustering tool takes O (n2) worst [2] case instance to find all the query clusters, where n is the total number of queries.

3.4 Favored Query Finder

When query [9] clusters are formed, another phase is to find a set of favored queries from each cluster. Query is said to be favored query that occupies the foremost portion of the search requests in a cluster. The process of finding favored queries is shown in fig4 which find the favored queries in one cluster. The method is applied in every the clusters and output is stored in the Query Cluster Database.

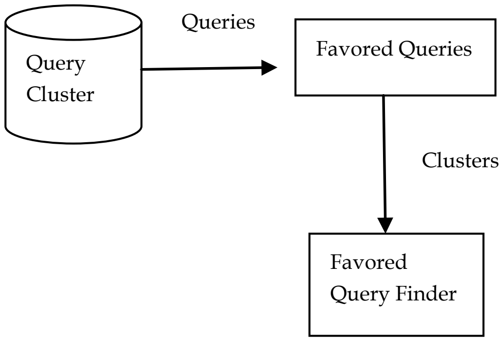

FIG 4: FAVORED QUERY FINDER

3.5 Query Recommender

Query Recommender provides [8] the user with a set of queries which are recommended with the most popular query. The recommended queries are those that are related to the query submitted by the user and therefore these queries are contained in the cluster of that query For example, the recommendations of a query APPLE are:
Apple
Apple India
Apple I Phone
Apple Store
Apple I Pad
The recommended queries are sorted with popular query being highlighted here underlined. When user submits a query, its keywords are matched in Query cluster database and the queries in the matched cluster are outputted by the Query Recommender on the interface of search engine. The user can carry on [8] with the same query otherwise can decide any one of the recommendation.

4 EXPERIMENTAL RESULTS

To validate the proposed approach, we [7] ran several experiments on the query logs of a search engine. A novel approach for result optimization and query recommendation is proposed that attempts to optimize the search engine's results. When combined, they can lead to more satisfactory results.

Query Similarity Calculations
To show the practical evaluation of the proposed architecture, a sample query Log is considered (given in Table 1. Let us consider the 6 queries in the query log. We want to calculate the similarity between the 2 queries (based on query keyword).
Q1=Data Mining
Q2= Data Ware housing
Sim (q1, q2) = 1/5 = 0.2
Sim (q1, q3) = 2/4 = 0.5
Sim (q1, q4) = 1/4 = 0.25
Sim (q1, q5) = 0

Table 2 represents similarities between queries

Query	1.	2.	3.	4.	5	6
1.Data Mining	0.5	0.2	0.5	0.25	0	0
2.Data Ware housing	0.2	0.5	0.2	0.2	0	0
3.Data Mining	0.5	0.2	0.5	0.25	0	0
4.Data Warehousing	0.25	0.2	0.25	0.5	0	0
5.Search Engine	0	0	0	0	0.5	0
6.Web Crawler	0	0	0	0	0	0.5

Now, we want to calculate the similarity between queries (based on clicked URLs).
Q1=Data Mining
Q2= Data Ware Housing
Sim (q1, q2) = 0.2
Sim (q1, q3) = 0.5
Sim (q1, q4) = 0.34
Sim (q1, q5) = 0.68

Table 3 represents the similarities between documents

Query	1.	2.	3.	4.	5.	6.
1.Data Mining	0.5	0.14	0.5	0.3	0.6	0.2
2.Data Ware Housing	0.8	0	0.8	0.16	1	0.1
3.Data Mining	0.5	0.14	0.5	0.3	0.6	0.2
4.Data Warehousing	0.3	0	0.3	1	0.5	0.3
5.Search Engine	0.6	0	0.6	0.4	1	0.33
6.Web Crawler	0.3	0	0.3	0.6	0.6	1

Since, the value of α and β are set to be 0.5. Therefore combined query similarity of first two queries q1=Data Mining, q2= Data Ware housing is to taken by using the formula (c) is as follow:

Sim= (0.5). (0.2)+ (0.5). (0.8) = 0.5

An example of Query Clustering is as shown. For calculating the query similarity based on both principles (a) and (b) or the combined measure (c) can be utilized.

The three cases given below describe the clusters obtained
Using different approaches:

Case 1: If the keyword-based measure is applied formula (a)), the queries are divided into 3 clusters:

Cluster 1: Query 1(Data Mining)
Cluster 2: Query 3 (Data Mining)
Cluster 3: Query 2 and Query 4 and Query 5 and Query 6(Data Ware housing, Data Warehousing, Search Engine, Web Crawler)
Queries 1 and 3 are not clustered together.

Case 2: If we use the measure based on individual documents (formula (b)), we obtain:

Cluster 1: Query 1(Data Mining)
Cluster 2: Query 3(Data Mining)
Cluster 3: Query 2 and Query 4 and Query 5 and Query 6(Data Ware housing, Data Warehousing, Search Engine, Web Crawler)
Now Queries 1 and 3 are not judged to be similar.

Case 3: Now let us use the combined measure (c), where α and β are set to 0.5 and similarity threshold (T) also set to 0.5. The queries are now clustered as:
Cluster 1: Query 1 and Query 3(Data Mining)

By analyzing the results obtained above through different Approaches, it is determined that the mixture of both query content and clicked documents based approach is more suitable for query clustering.

In the final step of [7] query recommendation process, recommended queries are selected from query log according to their query similarity to the new query submit. We here set up two criterions for this selection:
1. The user for recommendation should be within the top-k range i.e. relatively high among all the queries in the log.
2. We observe that some users are shown in the result which is in the range but irrelevant as those out of the range, we therefore set a threshold value to solve this crisis.

5 Conclusion and Future Work

In this paper, Architecture of result optimization system has been proposed based on query log for implementing effective web search. The most significant feature is that the result op-

timization method is based on users' feedback, which determines the relevance between Web pages and user query words. The returned pages with better page ranks are directly mapped to the user feedbacks and dictate higher relevance than pages that exist in the result list but are never accessed by the user. Hence, the time user spends for looking for the required information from search result list can be reduced and the more important Web pages can be presented.
The results obtained from practical evaluation are quite effective in respect to reduced search space and enhanced the use of interactive web search engines. As the future work, we apply a technique to overcome this problem. Conclusion
Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions. Authors are strongly encouraged not to call out multiple figures or tables in the conclusion—these should be referenced in the body of the paper.

ACKNOWLEDGMENT

This paper would not have been possible without my professor Mrs. Nikita Taneja. My deepest gratitude goes to her for the support and guidance. Her patience and encouragement throughout the research paper has motivated me to enthusiastically work and constantly improve. I am honored and thankful for the opportunity to be her student.
It is a great pleasure to thank my guide for being there for me always. Family values that I have gained directed me to the path of constant personal, professional and academic progress. My family has been supporting me in all mentioned aspects. I am very glad that I can share my success with them.
I thank also to my fellow students and friends with whom I shared these precious student days. The friendships that we made now are the ones that last.

REFERENCES

[1] D.Beeferman and A. Berger. Agglomerative Clustering of a Search Engine Query log. In KDD, pages 407-416, Boston, MA USA, 2000.

[2] J.Wen Nie and H.Zhang, Clustering user queries of a Search Engine. In Proceedings at 10th international World Wide Web Conference, pp 162-168, W3C, 2001.

[3] H. Cui, J-R Wen, J. Y.Nie and Lo. Y. Ma. “Query Expansion by Mining User Logs”. IEEE Trans. On Knowledge and Data Engineering, Press July pp-829-839.2003.

[4] B.M Fonseca, P.B Golger, E.S.De Moura and N.Ziviani, “ Using Association rules to discovery Search Engine related queries ”, proc. of first Latin American Web Congress, Santiago, Chile, Nov. 2003.

[5] Baeza- Yates, R. , Hurt ado, C., and Mendoza, M. “Query recommendation using query logs in search engine”. In proc. of int. Workshop on clustering information over the web, Crete, Springer pp 558-596, 2004

[6] Xuedong Shi and Christopher C. Yang. ” Mining related queries from web search engine query logs” an improved association rule mining model”. Wiley Periodicals, Inc. Published Online 3 August 2007 in Wiley Interscience.

[7] Ql Liu, Mingui Jiang, Zhi Chen.” Query Recommendation with the TF-IQF Model and Popularity Factor”. Proceedings of IEEE factor Beijing, China 2008.

[8] Neelam Duhan, A.K Sharma.”Rank Optimization and Query Recommendation in Search Engine using Web Log Mining Technique. Journal of Computing. Vol 2, Issue 12, Dec. 2010

[9] A.K Sharma, Neelam Duhan, Neha Aggarwal, Ranjana Gupta. “Web Search Result optimization by Mining the Search Engine Query Logs”. Proceedings of International Conference on methods and models in Computer Science, Delhi, India, Dec.13-14, 2010.

[10] Aris Anagnostopoules, Luca Becchetti, Carlos Castillo, Aristides Gionis.” An Optimization Framework for Query Recommendation”. WSDM February 4-6, New York, USA, 2010.

[11] Hamada M.Zahera, GI-Wahed.”Gamal F. El. Hady, Waiel, F.Abd EI-Wahed.” Query Recommendation for improving Search Engine Results”. Proceedings of the world congress on engineering and Computer Science Vol.1, October 20-22. San Franciso, USA. 2010.