Query Recommendation for Improving Search Engine Results

Hamada M.Zahera, Gamal F. El Hady, Waiel.F Abd El-Wahed

Abstract— 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. This paper proposes a method for suggesting a list of queries that are related to the user input query. The related queries are based on previously issued queries by the users. The proposed method is based on clustering process in which groups of semantically similar queries are detected. This facility provides some queries which are related to the queries submitted by users in order direct them toward their required information. This method not only discovered the related queries but also rank them according to a similarity measure. Finally the method has been evaluated using real data sets from the search engine query log.

***Index Terms*— Query Recommendation, Clustering, Query log, Search Engines**

I. INTRODUCTION

With the increase of size and popularity of the World Wide Web, many users find it's difficult to get the desired information, although they use most efficient search engines (e.g. Google, yahoo).Actually theses search engines allow users to specify queries simply as lists of keywords, following the approach of traditional information systems [1]. But this list of keywords is not always a good descriptor of the needed information, therefore it was important to achieve user's stratification of search engine results and make it easy to retrieve the required information.

The problem of improving search engine results and obtaining the desired information from this huge amount of web contents has been processed by different ways such as clustering the search engine results in specific topics so the user can find the required results in selected category of search results [2]. Although, the user doesn't use the proper search words or search query while searching so this leads to a problem of getting un-required results and the user have to be familiar with specific terminology in a knowledge domain [3].

This is not always the case of many users; they have only a little background about the information they are searching and unfortunately they didn't get the required results.In order to overcome this problem, it's not enough to use clustering search results method because the problem is not in obtaining the huge results but it's in the keywords used in searching are not strongly related [3].

Query recommendation suggests related queries for search engine users when they are not satisfied with the results of an initial input query, thus assisting users in improving search quality. Conventional approaches to query recommendation have been focused on expanding a query by terms extracted from various information sources such as a thesaurus like WordNet, the top ranked documents and so on [5].

The previous queries stored in query logs can be a source of additional evidence to help future users. A query recommendation system based on large-scale Web access logs and web page archive, and evaluate three query recommendation strategies based on different feature spaces (i.e., noun, URL, and Web community) has been presented [5]. The suggested Method aimed to help search engine users in finding their required results easily and quickly, this method suggests related queries beside the input query while the user searches so he can build a proper search query with the knowledge domain terminology which is important for search engine to get the related results. Also the additional time for improving the results must be unnoticeable by the user.

This paper is organized as the following. The related works is presented in section 2.Section 3 and section 4 describe the details of the experiment methodology and discuss the experimental results respectively. Finally, conclusions and future works are put forward in section 5.

II. RELATED WORK

Yates, R.B [6] has done a survey in to show the different improvements of search engine aspects. J.Wen [7] presented an algorithm for clustering search engine queries according to four notions according to: first, the context of the query; second, common clicked URLs between queries; third, Similar strings between the queries and fourth, the distance of the clicked documents in some pre-defined hierarchy. Befferman and Berger [8] suggested a technique for query clustering based on the third notion. Fonseca [9] presented a new method to discover the related queries which are based on association rules .The

H. M. Zahera is with Faculty of Computers and Information – Computer Science Dep. Menoufiya University, EGYPT, Fax: (048)-223694, email: hamadazahera@gmail.com, Tel: (002) 0108248981.

G. F. El Haddy is with Faculty of Computers and Information – Computer Science Dep. Menoufiya University, EGYPT, Fax: (048)-223694, email: gamal.elemtwali@ci.menofia.edu.eg, Tel: (002) 0107747539

W. F.. Abd El-Wahed is with Faculty of Computers and Information Operation Research and Decision support Dep. Menoufiya University, EGYPT, Fax: (048)-223694, email: wael.abdelwahad@ci.menofia.edu.eg, Tel: (002) 0106897047

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

queries represent items in tradition association rules. The query log file is considered as a collection of transactions which represent a session in which the user submit all related queries in a specific time .The method showed good results ,nevertheless arising of two problems .The first problem is the difficulty of determining which sessions of these queries that are belong to the same search process. The second problem the related queries which are submitted by different users can't be discovered .This is because the support of a rule increases only of its queries are in the same session and they must by submitted by the same user.

M.Hosseini and H.Abolhassni have described a method for recommending associated queries according to clustering process over web queries from search engines query log [4]. Zaiane and Strilets [10] presented a method for recommending queries according to seven aspects of query similarity, Three of them are moderated variations of the first and second notions .In addition our method recommend the related queries to the input query but my search for different issues like the previous information from query log file. There is anther approach to suggest related queries by query expansion. The researchers show that average query terms are near two [11]. So most of the time, queries are ambiguous. One possible solution for this problem is to expand a query with new terms. Query clustering helps to find relevant terms for this expansion which can be applied in two ways:

1- Query expansion in terms of similar queries.

2- Expansion in terms of selected pages of similar queries

III. Description of The Method

In order to compute the similarity between queries, first we built a term-weighted vector for each query. All previous queries stored in the query file are considered with their clicked URLs to recommend the users with the related queries to the input query. We represent the Queries with the clicked URLs in a bipartite graph so that it will be clear which are the URLs clicked with the query submitted by the users [4]. We symbolize the graph by

G (V, E) where V is the set of all vertices of queries (*Q*) and URLs (U), E is the set of edges between *Q* and *U*. The two vertex sets of the graph *Q* and *U* while $Q \cup U = V$ and $Q \cap U = \phi$, each edge connects between the query issued by user and the clicked URLs.

This representation of queries and links as a bipartite graph make it easy to find the similarity between queries as showed in Fig.1.

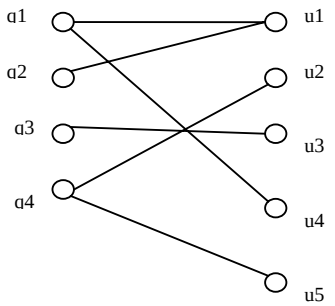

Fig 1. Query – URL representation as a bipartite graph

It's important to set a weight for every edge between *Q* and *U* to show the importance of this edge and distinguish between other edges.

We considered weighted bipartite graph as G (V, E, W) where W is number of clicked on the link *U* when the query *Q* is submitted by the user. Queries along with the clicked URLs extracted from Query log are clustered .This is a preprocessing phase before applying query recommendation algorithm which queries are similar and also to determine which is the most similar cluster to the input query. We compute clusters by k-mean algorithm because of its simple and more appreciate for document clustering [12] compared with other algorithms for document clustering.

A. QUERY RECOMMENDING ALGORITHM

In our research work, we applied an algorithm to recommend related queries to a query submitted by the user.

The clustering process aims to categorize all related queries into groups based on all information in the query log file. When the user submit a query, the algorithm finds the proper group of related queries and ranks them according to it's relevance to the user input query and finally it suggests all previous related queries to the user. The Query Recommendation Algorithm works as the following steps:

1. Queries and Their clicked URLs extracted from the search engine query log file are clustered by k-mean algorithm.
2. while the user submit an input query, the algorithm finds the similar cluster to the input query ;how it's close the centroied of which cluster
3. Query vectorization: in which each query is represented as a vector where j^{th} element represent between the query and URL j. a query vector is shown in (1)

$$\vec{q}_i = [r_1, r_2, r_3, \ldots . r_j] \qquad (1)$$

Where is the relation value between and URL j, it's computed as shown in (2)

$$r_j = \frac{w_{i_j}}{\sum_{k=1}^{n} w_{kj}} \times \log\left(\frac{|L|}{\sum_{k=1}^{m} connect\ (q_i, l_j)} \right) \qquad (2)$$

Where n is the total number of unique queries and m is the total number of distinct URLs. The first part of (2) is the ration between w_{ij} (number of click times) and total number of w_{ij} for all queries with the URL j. The second part is the logarithm of the ration between $|L|$ (total number of distinct URLs) to the number of URLs which *connect* (q_i, l_k) is a Boolean function as shown in (3)

$$connect\ (q_i, l_j) = \begin{cases} 1 & ; w_{ik} = 0 \\ 0 & ; w_{ik} \geq 0 \end{cases} \qquad (3)$$

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

4. Queries similarity: in order to compute similarity between queries, we use Tanimoto coefficient similarity measure to show how two queries are similar together as shown in (4)

$$T(q_i, q_j) = \frac{\vec{q}_i . \vec{q}_j}{\left|\vec{q}_i\right|^2 + \left|\vec{q}_j\right|^2 - \vec{q}_i \cdot \vec{q}_j} \qquad (4)$$

5. Support of the query: this is a measure of how much this query belongs to its cluster. The support of the query is measured as the ratio of the number of clicked URLs for the query |Li| to the total number of the URLs ;for all queries $\sum_{j \in C} |L_i|$ in the cluster C as shown in (5)

$$Sup(q_i) = \frac{|L_i|}{\sum_{j \in C} |L_i|} \qquad (5)$$

6. Finally, the queries in the selected cluster are ranked base on their similarity and their support; the rank score is measured as shown in (6) where the similarity between all queries in the cluster qi and the input query q. The term constants α and β used for normalization [4].

$$Rank(q_i) = \alpha \times T(q_i, q) + \beta \times Sup(q_i) \qquad (6)$$

IV. Experimental Results

In our experiments, a query log of the AOL search engine was used for collecting click through data. A record on this query log represents the visit to a result for a query or the submission of a query (if no result is visited) [13].

Each record store:

- An anonymous ID that allows to group queries from the same user without revealing the AOL user's nickname.
- Query submitted by the user.
- Date and time of the submission of the query
- Rank position of the result visited by the user on each record.

Examples of this record can be found in Table.1 and a full description of the AOL query log is illustrated in [14]

TABLE I. Example of AOL Log Queries

User ID	Query	Time	Visited URLs
123	Computer virus	2006-03-08 12:21:23	http://www.microsoft.com/security/antivirus/whatis.aspx http://www.howstuffworks.com/virus.htm http://www.snopes.com/computer/virus/virus.asp …..
154	Funny videos	2006-05-12 ,02:12:32	http://www.break.com/ http://www.funnyordie.com/ http://www.dailyhaha.com/ …..
217	Albert Einstein	2006-08-17 05:19:05	http://www.westegg.com/einstein/ http://www.albert-einstein.org/

TABLE2: Recommended Queries for "scholarship"

Query	Recommended Query	Similarity by Tanimoto	Similarity by Cosine	Rank Score by Tanimoto	Rank Score by Cosine
Q1	International scholarship	0.958	0.98	0.791	0.808
Q2	Grants and fellowships	0.921	0.989	0.765	0.82
Q3	Fulbright Scholarship	0.651	0.987	0.549	0.818
Q4	Study abroad chances	0.117	0.257	0.122	0.234
Q5	Scholarship programs	0.62	0.794	0.525	0.664
Q6	Full funded scholarship	0.422	0.743	0.366	0.623

In order to evaluate our similarity measure, we compared it with the Cosine similarity which has been used by Mehdi and Hassan [8]. We extracted 10,000 queries from AOL dataset for clustering. After constructing the clusters we select ten queries of the clustered dataset: (1) weather; (2) newspaper; (3) computer Software; (4) airline flights; (5) scholarships; (6) Nobel award; (7) five star hotels; (8) human

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

development books; (9) MIT universities; (10) migrations .All the selected queries are sent to the recommendation algorithm in order to suggest useful queries for the user. In addition to that, the suggest queries are ranked according to rank score which is calculated by Eq.6 with parameters α =0.8 and β =0.2.

Table 2, shows the quality of ranking based on two different similarity measure: Cosine similarity used by Mehdi and Hassan [4] and our similarity measure (Tanimoto coefficient measure).In this example, the algorithm recommend 6 related queries to the input query "scholarship"

In Fig.2, we compared the ranking of results based on cosine similarity measure [4], and Tanimoto coefficient similarity measure to show the efficiency of ranking. Cosine similarity has overestimate measurements of query similarity as we see query 2 "Grants and fellowships" is more similar to user query "Scholarships" rather than query1 "International scholarships". Although query 2 is semantically equal to the user query, but query 1 is much better because it has the most similar string and mean to the user query. Therefore this measure affect on result ranking of search engine by using Tanimoto coefficient in similarity measurement has more precision and accuracy in ranking the search engine results.

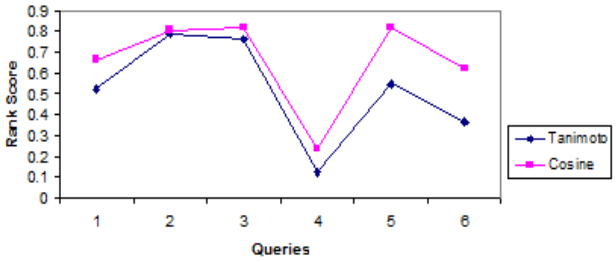

Fig. 2 Ranking efficiency between Tanimoto and Cosine Similarity

V. Conclusion and Future Work

We have presented a method for recommending the related queries to the input based on clustering process over the web queries extracted from a search engine query log. We are doing the experimentations with larger logs more than used and considering more queries to improve the evaluations of our approach. In addition, we are trying to expand the queries using the keywords related to the cluster Also we consider the improvement of Similarity by considering the clicks in query answer to documents that are similar to the input query

As future work, we consider to improve the notion of attention of the suggested queries and to expand other notions of interest for the recommendation algorithm. For example finding the queries which share words but not have common clicked URLs, this might involve the same words but have different meanings if the text of the URLs also is not share. Hence we can know polysomic words.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, chapter 3, pages 75–79. Addison-Wesley, 1999.

[2] Caramia, G. Felici and A. Pezzoli, “Improving search results with data mining in a thematic search engine,” Computer & Operations Research 31, pp. 2387-2404, (2004) Elsevier

[3] R. Baeza-Yates, C. Hurtado and M. Mendoza, “Query Recommendation Using Query Logs in Search Engines,” LNCS 3268, pp. 588-596, (2004), Springer-Verlag Berlin Heidelberg , 2004.

[4] M. Hosseini and H. Abolhassani, “Clustering search engines log for query recommendation,” CSICC, CCIS 6, pp. 380-387, (2008), Springer-Verlag Berlin Heidelberg 2008

[5] L. Li, S. Otsuka, and M. Kitsuregawa “Query Recommendation Using Large-Scale Web Access Logs and Web Page Archive,” LNCS 5181, pp. 134–141, (2008), Springer-Verlag Berlin Heidelberg 2008

[6] R. Yates, “Query usage mining in search engines,” in Scime, A. (ed.) Web Mining: Applications and Techniques. Idea Group (2004)

[7] J. Wen, J. Nie, H. Zhang, “Clustering user queries of a search engine,” in 10th International World Wide Web Conference. W3C, pp. 162–168 (2001)

[8] D. Beeferman, and A. Berger, “Agglomerative clustering of a search engine query log,” in KDD, Boston, MA USA, pp. 407–416 (2000)

[9] B. Fonseca, P. Golgher, E. De Moura, and N. Ziviani, “Using association rules to discovery search engines related queries,” in First Latin American Web Congress (LAWEB 2003), Santiago, Chile (November 2003)

[10] O. Zaiane, A. Strilets, “Finding similar queries to satisfy searches based on query traces,” in Proceedings of the International Workshop on Efficient Web-Based Information Systems (EWIS), Montpellier, France (September 2002) .

[11] D. Jansen, A. Spink, J. Bateman, and T. Saracevic, “Real life information retrieval: a study of user queries on the web,” ACM SIGIR Forum 32(1), 5–17 (1998)

[12] Y. Zhao and G. Karypis, “Comparison of agglomerative and partitional document clustering algorithms,” in SIAM Workshop on Clustering High-dimensional Data and its Applications, 2002.

[13] D. Brenes, and D. Gayo-Avello “Stratified analysis of AOL query log,” Information Sciences 179 (2009) pp.1844–1858, Elsevier .

[14] G. Pass, A. Chowdhury, and C. Torgeson, “A picture of search,” in The First International Conference on Scalable Information Systems, ACM, Hong Kong, 2006, ISBN 1-59593-428-6, pp.1–7.

[15] Caramia, G. Felici and A. Pezzoli, “Improving search results with data mining in a thematic search engine,” Computer & Operations Research 31, pp. 2387-2404, (2004) Elsevier.