Query Recommendation for Long Tail Queries - A Review Paper

Anand Prasad Gupta
Department of Computer Science & Engineering
Ajay Kumar Garg Engineering College, Ghaziabad

Sunita Yadav
Department of Computer Science & Engineering
Ajay Kumar Garg Engineering College, Ghaziabad

ABSTRACT
Large volume of queries over large volume of users allows search engine to build methods for generating query suggestion for input query. Query recommendation methods are powerful technique to generate related queries or alternate queries as a query suggestion for original query which is given by user in the search engine first time. In this review paper we discussed about query recommendation methods and comparison between different methods how the query suggestion is given by these methods for the input query by user in search engine. Some query recommendation method do not covers the unseen and rare queries but some of them covers these queries by some additional feature added such as generalize the query token of input query by a suitable place holder from hierarchy tool wordnet3.0 or wekepedia by yago mapping. These generalization techniques are also discussed in this paper.

Keywords
Query suggestion, query log, query session, click through, Web search, query templates, query recommendation.

1. INTRODUCTION
The explosive growth of web information has not only created a crucial challenge for search engine companies to handle large scale data, but also increased the difficulty for a user to manage his information need. It has become increasingly difficult for a user to compose a succinct and precise query to present his search need. Instead of pushing this burden to the users, it is common practice for a search engine to provide some types of query suggestions. The query is defined as whatever a user typed in search engine is called query. In most of the queries few keywords are common and remaining keywords comes under the long tail. These long tail queries are specific queries that may only occur once however, the total number of these individual queries is such that they form the bulk of the total search.
According 80/20 rule, 20% of the total number of search queries will be made up of the most common keywords and the remaining 80% comes under the long-tail.The long tail keywords beneficial for more targeted, refined and specified result as for users need. To recommend these long tail or unseen queries need of query recommendation method arised. These query recommendation techniques provide alternate query as a query suggestion which is related to first query given by the user[1,6].Recommending the most relevant search keywords set to users not only enhances the search engine's hit rate, but also helps the user to find the desired information more quickly [8].According to first and last(first, last) pair methodology for consecutive queries done by particular user in a particular session, the last query is taken as suggestion for first query but the last query must be related to first query.Query-recommendation methods use similarity measures obtained by mining the query terms, the clicked documents, and the user sessions containing the queries.
All the query log based recommendation methods leads to seen queries and do not cover the unseen and rare queries. This is the major drawback of these models. To recover this some query assistant services uses some special techniques to provide suggestion for unseen and rare queries such as query recommendation via template[1], recommendation by term query graph [6] and suggestion by hitting time [2] etc. Some query recommendation methods also works without using the query log. These methods use probabilistic mechanism for generating query suggestions from the corpus in absence of query logs [12].

2. QUERY RECOMMENDATION METHODS
There are many methods to recommend an input search query given by user and give related query suggestion for input query in search engine as discussed below.

2.1 Based on Templates
The long-tail distribution implies that a large fraction of queries are rare. As a result, most query assistance services perform poorly or are not even triggered on long-tail queries.
Idan Szpektor et al. (2009) proposed a method to extend the reach of query assistance techniques (and in particular query recommendation) to long-tail queries by reasoning about rules between query templates rather than individual query transitions, as done in query-flow graph models.
They proposed a technique to address the long-tail problem by leveraging query templates, which are query constructs that abstract and generalize queries [10, 11]. The approach was based on the fact that many individual queries share the same query intent while focusing on different entities. Hence, their related queries also share similar structures. As an example, assume that the queries "Los Angeles hotels", "New York hotels" and "Paris hotels" have been abstracted into the common query template "<city> hotels". In addition, if "Los Angeles restaurants" is a query recommendation for "Los Angeles hotels" and similarly "New York restaurants" is a recommendation for "New York hotels", they extracted the rule:
'<City> hotels → <City> restaurants'
In template based query recommendation method construction of templates recover the problem of query recommendation for rare and unseen queries by generalizing the unknown tokens of input queries by replacing it to suitable placeholder of generalization hierarchy over entities.

2.1.1 Template Construction

Idan Szpektor et al. (2009) stated how templates are constructed with the help of a generalization hierarchy H over entities [1]. The hierarchy H is the WordNet 3.0 hypernymy hierarchy and the Wikipedia category hierarchy, connected together via the yago induced mapping [11].It provided a method to generate an entity hierarchy, e.g. from query logs [10]. For each query q construction the set of templates T (q) as follows: First step to construct template is to normalize q by converting all characters to lower case, collapsing continuous spaces, removing non-printable characters, etc. then, every word n-gram up to length 3 in q is considered as a token for replacement by a placeholder form H. For every n gram k, added T (q) all the templates formed by replacing k with each of its generalizations in H [1].

2.2 Based on clicked document

Clicked document is the data which is clicked by the user in search engine volume space after giving the input to the search engine. The user clicks always on the desired URLs which is related to input query .More clicks on the URL indicates the behavior of user and query suggestion provided by clicked through data according to the behavior of the user clicks.

Baeza-Yates et al. (2004) proposed a measure of query similarity and use it to build methods for query expansion. Their technique is based on a term-weight vector representation of queries, obtained from the aggregation of the term-weight vectors of the URLs clicked after the query [3].

Wen et al. (2001) also presented a clustering method for query recommendation that is centered on the query-click graph [9].

D. Beeferman et al. (2000) proposed a new technique for mining the collection of user transaction with an internet search engine to discover clusters of similar queries and similar URLs [13]. They stated that each transaction record consists of user offered query in search engine and related URL comes into the search engine volume. By viewing the dataset as a bipartite graph, the queries on one side vertices and other side vertices is URLs then applying the agglomerative clustering algorithm to the graph vertices to identify related query and URLs to make query suggestions. The query-click graph is also utilized for finding related documents using random walks, finding related keywords for advertising, query rewriting through co-citation generalization and ranking related queries using the notion of hitting time [2].

Qiaozhu Mei et al. (2008) proposed a novel query suggestion algorithm based on ranking queries with the hitting time on a large scale bipartite graph. The method holds the semantic consistency between the suggested query and the original query and stated that every query is connected with a number of URLs, on which the users clicked when submitting the query to the search engine. The weights on the edges presented how many times the users used this query to access this URL. There is no edge connecting two queries, or two URLs [2].

The labeled query indicates the query for which user want to generate suggestions. Relatively, if for all URLs that used a query to access, and some other people exclusively use another query to access, that query is a good suggestion to the original query.

They used a computation of hitting time assumed that Q0 is the input query and setting h (Qi; 0) = 0 for all queries Qi except for the original query Q0 which has h (Q0; t) = 1 $\forall t \geq 0$, and then iterate the following for a fixed number m of iterations:

$$h(Qi; t) = \sum_{j \neq i} pj, h(Qj, t-1) \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

For a query Qi, the process computed h (Qi; t) was the probability that a random walk arrives to node Qi within t steps or less [2].

Rongwei Cen et al. (2009) proposed an approach to recognize reliable and meaningful user clicks (referred to as Relevant Clicks, RCs) in click-through data [7]. By modeling user click through behavior on search result lists, it is proposed several features to separate RCs from click noises. A learning algorithm is presented to estimate the quality of user clicks. And stated that generally users do not want to provide explicit feedback for search engine therefore, implicit feedback information extracted from click-through data [14] Clicked document is the data which is clicked by the user in search engine volume space after giving the input to the search engine. The user clicks always on the desired URLs which is related to input query .More clicks on the URL indicates the behavior of user and query suggestion provided by clicked through data according to the behavior of the user clicks[7].

To solve click relevance problem that, individual user clicks include bias and cannot be used directly as judgments of absolute relevance, state of- the-art approaches require a large volume of click-through data to extract user feedback information based on user group instead of individual user. They stated that these techniques only deal with hot queries with extensive user interaction data and are not applicable for long-tail queries. For dealing with long tail queries they extracted features based on individual user clicks instead of relying on global statistics oriented features. And they analyzed that When a user types a query into a search engine interface, and clicks some returned results, the results might satisfy user or not. And they defined, a click is a RC (relevance click) if the user is satisfied by the clicked result and they stated that when a user clicks an irrelevant result, he is not satisfied and still need more information. Then the query tends to be refined and resubmitted, or more results will be clicked. For different positions in click sequence, user pays different attention to search results. Before clicking any result, user is likely to pay more attention to result list by comparing the information of each result and user tends to stop when he finally reaches a satisfying document [7, 14].

The mining of clicked documents in search engine volume space discovers clusters of similar queries and similar documents because users clicks always on desired links or document which is related to input query and these clustering and similarity approaches help for query suggestion for input queries.

In hitting time algorithm the suggestion is made by value of hitting time. The query having less hitting time is the suggestion for the similar query which has more hitting time. Hitting time is the time which is taken by the input query to discover desired result.

2.3 Based on Term-Query Graph

Francesco Bonchi et al. (2011) designed a model enabling the generation of query suggestions also for rare and previously unseen queries. The model was based on a graph having two sets of nodes: Term nodes, and Query nodes. The graph induced a Markov chain on which a generic random walker starts from a subset of Term nodes, moves along Query nodes, and restarts (with a given probability) only from the same

initial subset of Term nodes. They stated that computing the stationary distribution of such a Markov chain is equivalent to extracting the so-called Center-piece Sub graph from the graph associated with the Markov chain itself. Given a query, they extracted its terms and set the restart subset to this term set. Therefore, there is no need require to a query to have been previously observed or seen for the recommending model to be able to generate suggestions[6].

After studying the algorithm of term-query graph we find that recommendation based on term query graph improves the recommendation for long tail or rare queries not only dependent on seen queries in query log. It recovers the limitation of many algorithms that based on query log such as query flow graph.

2.4 Based On Query Reformulations

Many effective approaches focus on the analysis of user query sessions. The user sessions containing the queries submitted by user in particular time interval.

Zhiyong Zhang et al. (2006) proposed the method for mining search engine query logs to get fast query recommendations on a large scale industrial-strength search engine. They studied and modeled search engine users' sequential search behavior, and interpreted this consecutive search behavior as client-side query refinement that should form the basis for the search engine's own query refinement process [8]. They combined this method with a traditional content based similarity method to compensate for the high scarcity of real query log data, and more specifically, the shortness of most query sessions [9].And they stated that some users who are not very familiar with a certain domain, then the user uses queries that are used by previous similar searchers who may have gradually refined their query, hence turning into expert searchers, to help these novices in their search. By mining query log of search engine the query recommendation for input query is quite easily done. They stated about two methods have their own advantages and they are complementary to each other. The first method was consecutive query based method, made up clusters that reflect all users' consecutive search behavior as in collaborative filtering [4]. The arcs between consecutive queries in the same session are weighted by a dumping factor d, and the similarity values for non consecutive queries are calculated by multiplying the values of arcs that join them. While the second, content based method, the grouping queries together that have similar composition .By combined they together, for one specific query, get two clusters of related queries, which were sorted according to their cumulative similarity voting factor, when using the two methods [8].

D. Beeferman et al. (2000) used a "content-ignorant" approach and a graph-based iterative clustering method was used to cluster both the URLs and queries [4].

J.-R. Wen et al. (2001) used a method to provide solution for query log clustering by combining content based clustering techniques and cross-reference-based clustering techniques [9].

2.5 Based On Query-Flow Graph

The query flow graph is an aggregated representation of the latent querying behavior contained in a query log.

Boldi et al. (2009) stated that in the query-flow graph a directed edge from query qi to query qj means that the two queries are likely to be part of the same search and stated that in query flow graph the transition between query to query is done[5]. In edge (qi, qj) pair the transition between qi and qj where qj is specialization of qi and may be a query suggestion for qi if and only if qj is related to qi. The query recommendation methods are based on the probability of being at a certain node after performing a random walk over a query graph [15]. They stated that random walk starts in the node corresponding to the input query. At each step, the random walker either remains in the same node with probability 0:9, or follows one of the out-links with probability equal to 0:1; and after that, the links are followed proportionally to w (i, j). And they analyzed that increasing the number of iterations greater than 10 does not improve the results because much of the probability stays close to the original node. They compared two different scoring methods for query suggestion. In the first case the queries to present to the user are chosen based on the personalized Page Rank values obtained by the random walk. And an alternative scoring method ranks the results based on the ratio between the values obtained in the personalization and the Page Rank values obtained by using no personalization (starting at random at any node. They developed attempts to infer the hidden semantics of user interactions with search engines by extracting data from a query log in two steps. First, identify search mission borders by distinguishing query transitions that are reformulations, i.e., queries with a similar information need [17], from query transitions that represent a mission change. And they discussed that built a machine learning model [16] for predicting the probability that two subsequent queries are part of the same search mission. After identifying the search missions, the query reformulations inside each mission can be classified into query reformulation types. In particular, in work the identification of 4 query reformulation types: generalization, specialization, error correction, and parallel move [16].

Query recommendation based on query flow graph is based on query log and only support suggestion for seen queries. It is not effective for unseen queries.

2.6 Based On Probabilistic Mechanism

Sumit Bhatia et al. (2011) stated that major web search engines and most proposed methods that suggest queries depends on search engine query logs to determine possible query suggestions but for customized search systems in the enterprise domain, intranet search, or personalized search such as email or desktop search or for infrequent queries, query logs are either not available or the user base and the number of past user queries is too small to learn appropriate models [12]. They proposed a probabilistic mechanism for generating query suggestions [15] from the corpus with out using query logs and utilize the document corpus to extract a set of candidate phrases. They proposed a method, as a user starts typing a query; phrases that are highly correlated with the partial user query are selected as completions of the partial query and are offered as query suggestions.

This approach is completely different from the other approaches which have query log .here query log is absent and suggestion for input queries is made by probabilistic mechanism based on correlation between phrases and partial input query.

3. CONCLUSION

Query recommendation based on clicked documents, query-click graph, query flow graph, Query Reformulations have one limitation that these do not provide query recommendation for queries that were not seen before. Additionally, the quality of recommendations declines for infrequent queries because these models based on query log only and query log leads the seen query only not recommends

unseen queries. Through an analysis on search behaviors, rare queries are very important, and their effective satisfaction is very challenging for search engines. Therefore, it is even very important to provide good recommendations for long-tail queries. In this paper some query recommendation methods has been discussed that provides recommendations for long tail queries by adding some extra features such that generalize the input query token by a suitable place holder from generalized hierarchy and by calculating hitting time to make suggestion for input query if hitting time is low the recommendation is high. And some other methods such as recommendation by term-query graph also recover these limitations.

4. REFERENCES

[1] Idan Szpektor, Aristides Gionis, Yoelle Maarek (2009). "Improving Recommendation for Long-tail Queries via Templates", International World Wide WebConference, WWW 2009.

[2] Q. Mei, D. Zhou, and K. Church (In CIKM '08). "Query suggestion using hitting time", proceeding of the 17th ACM conference on Information and knowledge management, pages 469–478.

[3] R. A. Baeza-Yates, C. A. Hurtado, and M. Mendoza. (2004). "Query recommendation using query logs in search engines". In EDBT Workshops, pages 588–596.

[4] D. Beeferman and A. Berger (2000). "Agglomerative clustering of a search engine query log", In KDD.

[5] Paolo Boldi, Francesco Bonchi, Carlos Castillo (2009). "Query Suggestions Using Query-Flow Graphs", WSCD '09, Feb 9, 2009. In WSCD '09: Proc. of the workshop on Web Search Click Data, pages 56–63, New York, NY, USA. ACM.

[6] Francesco Bonchi, Raffaele Perego, Fabrizio Silvestri (2011). **"**Recommendations for the Long Tail by Term-Query Graph", WWW 2011.

[7] Rongwei Cen, Yiqun Liu, Min Zhang, Bo Zhou, Liyun Ru, Shaoping Ma (2009). **"**Exploring Relevance for Clicks"**,** CIKM'09.

[8] Z. Zhang and O. Nasraoui (2006). "Mining search engine query logs for query recommendation", In WWW.

[9] J.-R. Wen, J.-Y. Nie and H.-J. Zhang (2001). "Clustering user queries of a search engine", In Proceedings of WWW '01, pages 162–168.

[10] M.Fern´andez-Fern´andez and D.Gayo Avello (2009). "Hierarchical taxonomy extraction by mining topical query sessions", In KDIR.

[11] F. M. Suchanek, G. Kasneci, and G. Weikum (2007). "Yago: A core of semantic knowledge-unifying wordnet and Wikipedia", In WWW.

[12] Sumit Bhatia, Debapriyo Majumdar, Prasenjit Mitra (2011)."Query Suggestions in the Absence of Query Logs"**.** SIGIR'11, July 24–28.

[13] D. Beeferman and A. Berger (2000). "Agglomerative clustering of a search engine query log", In Proceedings of KDD'00, pages 407–416.

[14] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. (2006) "Accurately interpreting click through data as implicit feedback". In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 154–161.

[15] H. Cui, J.-R.Wen, J.-Y. Nie, and W.-Y. Ma (2002). "Probabilistic query expansion using query logs", In WWW '02: Proceedings of the 11th international conference on World Wide Web, pages 325–332.

[16] Boldi P., Bonchi F., Castillo C., and Vigna, S (2008)."Query reformulation models and patterns", Submitted for publication.

[17] Jones, R., and Klinkner, K. L (2008). "Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs", In Conference on Information and Knowledge Management (CIKM) (October), ACM Press.

[18] Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., and Gay.G (2007). "Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search", ACM Trans. Inf. Syst. 25, 2 (Apr. 2007).