International Journal of Computer Science & Mechatronics
A peer reviewed International Journal |Article available at http://ijcsm.in | SJIF: 4.454
©smsamspublications.com | Vol.3.Issue.3.2017

A New Approach of Privacy-Preserving location aware Keyword Query Suggestion (LKS) in Geo-Based Social Applications

Kommu Akshara Jyothi[1], MD.Imran[2]
[1]M.Tech (CSE) Project, [2]Assistant Professor of CSE, Department of CSE
Nimra Institute of Science and Technology, AP. India.
Email: ***akshara.jyothi1@gmail.com***

Abstract: Location-aware keyword queries suggestion (LKS), such that the suggested queries retrieve documents not only related to the user information needs but also located near the user location. Keyword suggestions are the most basic feature of the search engine. Naive users don't know how to express their queries correctly most of the time queries are short and unambiguous; keyword suggestion in web search assists users to access relevant information. Information will be more relevant if location of the users is considered. The location aware keyword (LKS) query suggestion method helps retrieve documents which relates to information provides by user and location where the user is located. However the search results' relevance is known to be correlated with the spatial proximity of the user. In this paper we give brief study all the techniques for the keyword suggestions and discuss about location-aware keyword query suggestion framework and improved Partition Based algorithm

Keywords — **Location privacy, security, location-based social applications, location transformation, efficiency.**

1. INTRODUCTION:

Keyword suggestion (also known as query suggestion) has become one of the most fundamental features of commercial web search engines. After submitting a keyword query, the user may not be satisfied with the results, so the keyword suggestion module of the search engine recommends a z set of m keyword queries that are most likely to refine the user's search in the right direction. Effective keyword suggestion methods are based on click information from query logs and query session data, or query topic models new keyword suggestions can be determined according to their semantic relevance to the original keyword query. However, to our knowledge, none of the existing methods provide location-aware keyword query suggestion (LKS); such that the suggested queries retrieve documents not only related to the user information needs but also located near the user location. This requirement merges due to the popularity of spatial keyword search Google processed a daily average of 4.7 billion queries in 2011, a substantial fraction of which have local intent and target spatial web objects (i.e., points of interest with a web presence having locations as well as text Data mining is the information of domain we are mining like concept hierarchies, to organize attributes onto various levels of abstraction.

A Spatial Keyword query is an approach of searching qualified spatial objects by considering both the query requester's location and user specified keywords. Taking both spatial and Keyword requirements into account, the goal of a spatial keyword query is to efficiently find results

that satisfy all the conditions of a search. Searching is a common activity happening in data mining. This motivated to develop methods to retrieve

Spatial objects. A spatial object consists of objects associated with spatial features. In other words, spatial objects involve spatial data along with longitude and latitude of location. The importance of spatial databases is reflected by the convenience of modeling entities of reality in a geometric manner. For example, locations of restaurants, hotels, hospitals and so on are often represented as points in a map, while larger extents such as parks, lakes, and landscapes often as a combination of rectangles. Many functionalities of a spatial database are useful in various ways in specific contexts.

2. LITURATURE SURVEY

Optimal Rare Query Suggestion with Implicit User Feedback

Query suggestion has been an effective approach to help users narrow down to the information they need. However, most of existing studies focused on only popular/head queries. Since rare queries possess much less information (e.g., clicks) than popular queries in the query logs, it is much more difficult to efficiently suggest relevant queries to a rare query.

In this paper, we propose an optimal rare query suggestion framework by leveraging implicit feedbacks from users in the query logs. Our model resembles the principle of pseudo-relevance feedback which assumes that top-returned results by search engines are relevant. However, we argue that the clicked URLs and skipped URLs contain different levels of information and thus should be treated differently. Hence, our framework optimally combines both the click and skips information from users and uses a random walk model to optimize the query correlation. Our model specifically optimizes two parameters: (1) the restarting (jumping) rate of random walk, and (2) the combination ratio of click and skip information. Unlike the Rocchio algorithm, our learning process does not involve the content of the URLs but simply leverages the click and skip counts in the query URL bipartite graphs. Consequently, our model is capable of scaling up to the need of commercial search engines. Experimental results on one-month query logs from a large commercial search engine with over 40 million rare queries demonstrate the superiority of our framework, with statistical significance, over the traditional random walk models and pseudo-relevance feedback models.

Time-aware Structured Query Suggestion: Most commercial search engines have a query suggestion feature, which is designed to capture various possible search intents behind the user's original query. However, even though different search intents behind a given query may have been popular at different time periods in the past, existing query suggestion methods neither utilize nor present such information.

In this study, we propose Time-aware Structured Query Suggestion (TaSQS) which clusters query suggestions along a timeline so that the user can narrow down his search from a temporal point of view. Moreover, when a suggested query is clicked, TaSQS presents web pages from query-URL bipartite graphs after ranking them according to the click counts within a particular time period. Our experiments using data from a commercial search engine log show that the time-aware clustering and the time-aware document ranking features of TaSQS are both effective.

An Optimization Framework for Query Recommendation

Query recommendation is an integral part of modern search engines. The goal of query recommendation is to facilitate users while searching for information. Query recommendation also allows users to explore concepts related to their information needs. In this paper, we present a formal treatment of the problem of query recommendation. In our framework we model the querying behaviour of users by a probabilistic reformulation graph, or query-flow graph [Boldi et al. CIKM 2008]. A sequence of queries submitted by a user can be seen as a path on this graph. Assigning score values to queries allows us to define suitable utility functions and to consider the expected utility achieved by a reformulation path on the query-flow graph.

Providing recommendations can be seen as adding shortcuts in the query-flow graph that "nudge" the reformulation paths of users, in such a way that users are more likely to follow paths with larger expected utility.

We discuss in detail the most important questions that arise in the proposed framework. In particular, we provide examples of meaningful utility functions to optimize, we discuss how to estimate the effect of recommendations on the reformulation probabilities, we address the complexity of the optimization problems that we consider, we suggest efficient algorithmic solutions, and we validate our models and algorithms with extensive experimentation. Our techniques can be applied to other scenarios where user behaviour can be modelled as a Markov process.

The query-flow graph: model and applications

Query logs record the queries and the actions of the users of search engines, and as such they contain valuable information about the interests, the preferences, and the behaviour of the users, as well as their implicit feedback to search-engine results. Mining the wealth of information available in the query logs has many important applications including query log analysis, user profiling and personalization, advertising, query recommendation, and more. In this paper we introduce the query-flow graph, a graph representation of the interesting knowledge about latent querying behaviour. Intuitively, 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 mission". Any path over the query-flow graph may be seen as a searching behaviour, whose likelihood is given by the strength of the edges along the path.

The query-flow graph is an outcome of query-log mining and, at the same time, a useful tool for it. We propose a methodology that builds such a graph by mining time and textual information as well as aggregating queries from different users. Using this approach we build a real world query-flow graph from a large-scale query log, and we demonstrate its utility in concrete applications, namely, finding logical sessions, and query recommendation. We believe, however, that the usefulness of the query-flow graph goes beyond these two applications.

3. RELATED WORK

Query Recommendation using Query Logs in Search Engines

We propose a method that, given a query submitted to a search engine, suggests a list of related queries. The related queries are based in previously issued queries, and can be issued by the user to the search engine to tune or redirect the search process. The method proposed is based on a query clustering process in which groups of semantically similar queries are identified. The clustering process uses the content of historical preferences of users registered in the query log of the search engine. The method not only discovers the related queries, but also ranks them according to a relevance criterion. Finally, we show with experiments over the query log of a search engine the effectiveness of the method.

A key factor for the popularity of today's Web search engines is the friendly user interfaces they provide. Indeed, search engines allow users to specify queries simply as lists of keywords, following the approach of traditional information retrieval systems. Keywords may refer to broad topics, to technical terminology, or even to proper nouns that can be used to guide the search process to the relevant collection of documents. Despite that this simple interaction mechanism has proved to be successful for searching the Web, a list of keywords is not always a good descriptor of the information needs of users. It is not always easy for users to formulate effective queries to search engines. One reason for this is the ambiguity that arises in many terms of a language.

Queries having ambiguous terms may retrieve documents which are not what users are searching for. On the other hand, users typically submit very short queries to the search engine, and short queries are more likely to be ambiguous. From a study of the log of a popular search engine, Jansen et al, conclude that most queries are short (around 2 terms per query) and imprecise. Users searching for the same information may phrase their queries

differently. Often, users try different queries until they are satisfied with the results.

In order to formulate effective queries, users may need to be familiar with specific terminology in a knowledge domain. This is not always the case: users may have little knowledge about the information they are searching, and worst, they could not even be certain about what to search for. As an example, a tourist seeking for summer rentals ads in Chile may not know that the vast majority of such ads in the Web are for apartments in Vina del Mar, a popular beach in the central part of Chile. In contrast, local users may have the expertise to submit queries with the term Vina del Mar, when they are looking for a location to spend their vacations.

The idea is to use these expert queries to help non-expert users. In order to overcome these problems, some search engines have implemented methods to suggest alternative queries to users3. Their aim is to help the users to specify alternative related queries in their search process. Typically, the list of suggested queries is computed by processing the query log of the search engine, which stores the history of previously submitted queries and the URL's selected in their answers. A central problem that arises in this context is how to model the information needs associated to a query. Some models proposed in previous work represent a query as the set of URL's clicked by users for the query.

This approach have limitations when it comes to identifying similar queries, because two related queries may output different URL's in the first places of their answers, thus inducing clicks in different URL's. In addition, as an empirical study shows, the average number of pages clicked per answer is very low (around 2 clicks per query). Our data shows the same. As in traditional document retrieval, in query recommendation one may expect that the ordering in which the queries are returned to the user plays a central role in the quality of the service, even more important than the set of recommendations itself. As far as we know, a problem not yet addressed is the definition of a notion of interest for the suggested queries.

Context-Aware Query Suggestion by Mining Click-Through and Session Data

Query suggestion plays an important role in improving the usability of search engines. Although some recently proposed methods can make meaningful query suggestions by mining query patterns from search logs, none of them are context-aware – they do not take into account the immediately preceding queries as context in query suggestion. In this paper, we propose a novel context-aware query suggestion approach which is in two steps. In the offline model learning step, to address data sparseness, queries are summarized into concepts by clustering a click-through bipartite. Then, from session data a concept sequence suffix tree is constructed as the query suggestion model. In the online query suggestion step, a user's search context is captured by mapping the query sequence submitted by the user to a sequence of concepts. By looking up the context in the concept sequence suffix tree, our approach suggests queries to the user in a context-aware manner. We test our approach on a large-scale search log of a commercial search engine containing 1.8 billion search queries, 2.6 billion clicks, and 840 million query sessions. The experimental results clearly show that our approach outperforms two baseline methods in both coverage and quality of suggestions.

Random Walks on the Click Graph

Search engines can record which documents were clicked for which query, and use these query-document pairs as 'soft' relevance judgments. However, compared to the true judgments, click logs give noisy and sparse relevance information. We apply a Markov random walk model to a large click log, producing a probabilistic ranking of documents for a given query. A key advantage of the model is its ability to retrieve relevant documents that have not yet been clicked for that query and rank those effectively. We conduct experiments on click logs from image search, comparing our ('backward') random walk model to a different ('forward') random walk, varying parameters such as walk length and self-transition probability. The most effective combination is a long backward walk with high self-transition probability.

4. CONCLUSION & FUTURE ENHANCEMENT

In this paper, proposed an LKS framework providing keyword suggestions that are relevant to the user information needs and at the same time can retrieve relevant documents near the user location.

A baseline algorithm extended from algorithm BCA is introduced to solve the problem. Then, we proposed a partition-based algorithm (PA) which computes the scores of the candidate keyword queries at the partition level and utilizes a lazy mechanism to greatly reduce the computational cost. Empirical studies are conducted to study the effectiveness of our LKS framework and the performance of the proposed algorithms.

The result shows that the framework can offer useful suggestions and that PA outperforms the baseline algorithm significantly. In the future, we plan to further study the effectiveness of the LKS framework by collecting more data and designing a benchmark. In addition, subject to the availability of data, we will adapt and test LKS for the case where the locations of the query issuers are available in the query log. In addition, we believe that PA can also be applied to accelerate RWR on general graphs with dynamic edge weights and we will investigate its general applicability in the future. Moreover, the current version of PA seems to be independent of the partitioning method. It would be interesting to investigate whether alternative partitioning heuristics can further reduce the cost of the algorithm. In our future work we want to prepare synonym suggestion for given keyword query using offline dictionary interface.

REFERENCES:

[1] M. P. Kato, T. Sakai, and K. Tanaka, "When do people use query suggestion? A query suggestion log analysis," Inf. Retr., vol. 16, no. 6, pp. 725–746, 2013.

[2] R. Baeza-Yates, C. Hurtado, and M. Mendoza, "Query recommendation using query logs in search engines," in EDBT, 2004, pp. 588–596.

[3] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in KDD, 2000, pp. 407–416.

[4] 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," in KDD, 2008, pp. 875–883.

[5] N. Craswell and M. Szummer, "Random walks on the click graph," in SIGIR, 2007, pp. 239–246.

[6] Q. Mei, D. Zhou, and K. Church, "Query suggestion using hitting time," in CIKM, 2008, pp. 469–478.

[7] Y. Song and L.-w. He, "Optimal rare query suggestion with implicit user feedback," in WWW, 2010, pp. 901–910.

[8] T. Miyanishi and T. Sakai, "Time-aware structured query suggestion," in SIGIR, 2013, pp. 809–812.

[9] A. Anagnostopoulos, L. Becchetti, C. Castillo, and A. Gionis, "An optimization framework for query recommendation," in WSDM, 2010, pp. 161–170.

[10] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna, "The query-flow graph: Model and applications," in CIKM, 2008, pp. 609–618.

[11] Y. Song, D. Zhou, and L.-w. He, "Query suggestion by constructing term-transition graphs," in WSDM, 2012, pp. 353–362.

[12] L. Li, G. Xu, Z. Yang, P. Dolog, Y. Zhang, and M. Kitsuregawa, "An efficient approach to suggesting topically related web queries using hidden topic model," WWW, pp. 273–297, 2013.

ABOUT AUTHORS:

KOMMU AKSHARA JYOTHI is a student of Nimra institute of science and Technology, Ibrahimpatnam, VIJAYAWADA. She is presently pursuing his M.Tech degree from JNTU, Kakinada.

MD.IMRAN is presently working as Assistant professor in CSE department, Nimra institute of science and Technology, Ibrahimpatnam, Vijayawada.