Location Aware Keyword Query Suggestion Based on Document Proximity

Shuyao Qi
University of Hong Kong
Email: qisy@connect.hku.hk

Dingming Wu
Shenzhen University
Email: dingming@szu.edu.cn

Nikos Mamoulis
University of Ioannina
Email: nikos@cs.uoi.gr

***Abstract*—Consider a user who has issued a keyword query to a search engine. We study the effective suggestion of alternative keyword queries to the user, which are semantically relevant to the original query and they have as results documents that correspond to objects near the user's location. For this purpose, we propose a weighted keyword-document graph which captures semantic and proximity relevance between queries and documents. Then, we use the graph to suggest queries that are near in terms of graph distance to the original queries. To make our framework scalable, we propose a partition-based approach that greatly outperforms the baseline algorithm.**

I. Introduction

Keyword suggestion (a.k.a query suggestion) has become a key feature 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 engine recommends a set of alternative 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 [6] and query session data [8], or query topic models [3]. 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, such that the suggested queries retrieve documents not only related to the user information needs but also located near the user location. This requirement emerges due to the popularity of spatial keyword search [9].

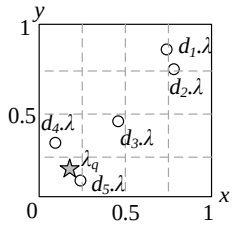
In this paper, we propose the first Location-aware Keyword query Suggestion (LKS) framework. We illustrate the benefit of LKS using a toy example. Consider five geo-documents d_1–d_5 as listed in Figure 1(a). Each document d_i is associated with a location $d_i.\lambda$ as shown in Figure 1(b). Assume that a user issues a keyword query k_q = "*seafood*" at location λ_q, shown in Figure 1(b). Note that the relevant documents d_1–d_3 (containing "*seafood*") are far from λ_q. A location-aware suggestion is "*lobster*", which can retrieve nearby documents d_4 and d_5 that are also relevant to the user's original search intention. Previous keyword query suggestion models (e.g., [4]) ignore the user location and would suggest "*fish*", which again fails to retrieve nearby relevant documents.

II. LKS Framework

Without loss of generality, we assume a set of geo-documents D such that each document $d_i \in D$ has a point location $d_i.\lambda$. Let K be a collection of keyword queries from a query log.

d_1	Fish and Seafood
d_2	Fish Seafood
d_3	Lobster Seafood
d_4	Lobster Restaurant
d_5	Lobster House
k_1	Fish
k_2	Seafood
k_3	Lobster

(a) Documents and Keyword Queries



(b) Locations of Documents

Fig. 1. LKS Example

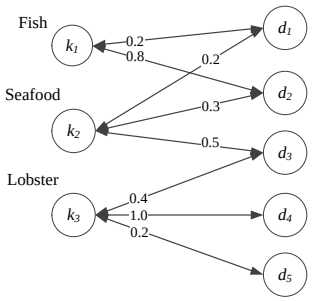


Fig. 2. KD-Graph

LKS, in a preprocessing stage, constructs an *initial* keyword-document graph (KD-graph), which is what a classic keyword suggestion approach that does not consider locations would use [4], [6], [8]. This directed weighted bipartite graph $G = (D, K, E)$ between D and K captures the semantic relevance between queries and documents. If a document d_i is clicked by a user who issued keyword query k_j in the query log, E contains an edge e from k_j to d_i and an edge e' from d_i to k_j. The weights of edges e and e' are the same and equal to the number of clicks on document d_i, given keyword query k_j. Therefore, the direct relevance between a query and a clicked document is captured by the edge weight. Furthermore, the semantic relevance between two queries is captured by their proximity in the graph G, measured by their *random walk with restart* (RWR) distance. As an example, Figure 2 shows the KD-graph (with normalized edge weights) for the documents d_1–d_5 and queries k_1–k_3 of Figure 1(a).

Consider a user-supplied query q with a single word or a phrase k_q. In order to satisfy the location-awareness criterion of query suggestion for q, we propose to *adjust* the edge weights in the KD-graph based on the spatial relationships between the location of the query issuer and the nodes of the KD-graph. Note that this edge adjustment is query-dependent and dynamic. In other words, different adjustment is applied

for each different query independently. Specifically, given q, the weight $w(e)$ of the edge e from a keyword query node k_i to a document node d_j is adjusted by the following function:

$$\tilde{w}(e) = \beta \times w(e) + (1 - \beta) \times (1 - dist(\lambda_q, d_j.\lambda)) \quad (1)$$

where $w(e)$ is the initial weight of e in the KD-graph, $\tilde{w}(e)$ is the adjusted edge weight, $dist(\lambda_q, d_j.\lambda)$ is the (normalized to $[0, 1]$) Euclidean distance between the query issuer's location λ_q and document d_j, and parameter $\beta \in [0, 1]$ is used to balance the importance between the original (i.e., click-based) weight and the distance of d_j to the query location. This keyword-to-document edge weight adjustment increases the weights of the documents that are close to the user's location.

Let $D(k_i)$ be the set of documents connected to a keyword query $k_i \in K$ in the KD-graph. $D(k_i)$ may contain multiple documents and the locations of them form a spatial distribution. We propose to adjust the weights of the edges pointing to k_i by the minimum distance between λ_q and the locations of documents in $D(k_i)$. Such an adjustment favors keyword query nodes which have at least one relevant document close to the query issuer's location λ_q. Specifically, the weight $w(e')$ of the edge e' from a document node d_j to a keyword query node k_i is adjusted as follows:

$$\tilde{w}(e') = \beta \times w(e') + (1 - \beta) \times (1 - mindist(\lambda_q, D(k_i))) \quad (2)$$

where $mindist(\lambda_q, D(k_i))$ is the minimum Euclidean distance between λ_q and any document in $D(k_i)$.

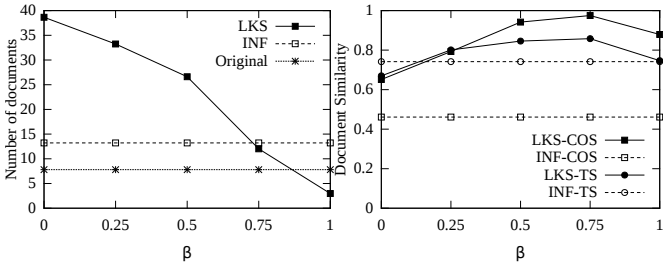
After edge weight adjustment, in order to find the set of keyword queries for recommendation, LKS computes on the KD-graph the RWR score from k_q to all other keyword queries and returns the keyword nodes with the top-m scores as suggestions. This can be done using the popular Bookmark-Coloring Algorithm (BCA) [1], which we adapt to form a baseline algorithm (BA) for LKS. The difference between BCA and BA is that BA only retains ink in keyword nodes, while all incoming ink to document nodes is redistributed.

BA can be slow due to the low rate with which the active ink drops and due to the large number of nodes to prioritize for ink redistribution in large problems. Therefore, we propose a partition-based algorithm (PA) that divides the keyword queries and the documents in the KD-graph into groups according to their spatial distance and textual similarity. The partitioning is done offline on the initial KD-graph. The original KD graph is then transformed to two KD-graphs; one containing only edges from query nodes to document partitions and one containing edges from documents to query partitions. PA delays the ink redistribution into partitions and adopts a lazy distribution mechanism which temporarily buffers ink at each node until the ink exceeds a threshold ϵ, in order to reduce the overall number of iterations. Our experimental results [7] suggest that PA outperforms BA by up to an order of magnitude.

III. Experiments

We conducted extensive experiments to evaluate the effectiveness of LKS and the efficiency of PA. Here we only show a few representative results. Dataset TWEET is based on tweets published inside the New York area. We extracted phrases of length 1 to 10 from the text messages and used them to model the keyword queries; we added an edge to the KD-graph between a query node k_i and a tweet node d_j if d_j contains keywords k_i, using the tf-idf score as its weight.

As a competitor to our LKS framework, we implemented the *influence tag co-occurrence* (INF) method [5]. Given a Flickr photo p published at $p.loc$ and a tag t, INF retrieves k tag recommendations that can be used as alternatives to t, considering both the location and textual information of p. To apply INF in keyword suggestion, we consider the co-occurrence of keywords in the same document and exploit a similar relevance function to the one suggested in [5] for tag recommendation. For a workload of 100 randomly selected keyword queries, Figure 3(a) reports the average number of nearby documents (within 5% of the maximum distance on the map) retrieved by (a) the original query, (b) the queries suggested by INF, and (c) the queries suggested by our LKS framework, when varying parameter β (in Equations 1 and 1). Figure 3(b) compares the similarity of the top-10 retrieved documents by LKS and INF to the top-10 retrieved by the original query. We used two measures; the ranking position similarity (COS) [2] and the textual similarity (TS) of the retrieved documents by the suggestions of LKS and INF to the original query. The plots show that if we balance the importance of semantic relevance and the spatial distance (i.e., $\beta = 0.5$), LKS is superior to INF in terms of both number of nearby documents and relevance of these documents to the original query.



(a) # retrieved nearby documents (b) document similarity

Fig. 3. Effectiveness evaluation

References

[1] P. Berkhin, "Bookmark-coloring algorithm for personalized PageRank computing," *Internet Math*, vol. 3, pp. 41–62, 2006.

[2] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, Oct. 2002.

[3] 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.

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

[5] I. Miliou and A. Vlachou, "Location-aware tag recommendations for Flickr," in *DEXA*, 2014, pp. 97–104.

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

[7] S. Qi, D. Wu, and N. Mamoulis, "Location aware keyword query suggestion based on document proximity," *IEEE Trans. Knowl. Data Eng.*, to appear.

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

[9] D. Wu, M. L. Yiu, and C. S. Jensen, "Moving spatial keyword queries: Formulation, methods, and analysis," *ACM Trans. Database Syst.*, vol. 38, no. 1, 2013.