ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Keyword Query Suggestion Based on Document Proximity

M.Padmaja
M.Tech Student,
Deaprtment of CSE,
KIET Engineering College.

T.Rajendra Prasad MCA, M.Tech
Assistant Professor,
Deaprtment of CSE,
KIET Engineering College.

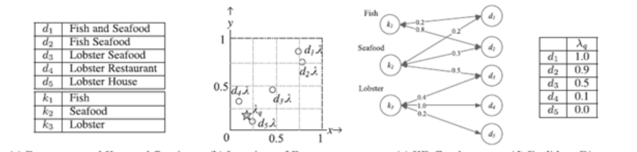
Abstract:
In this paper, we design a location-aware keyword query suggestion framework. We propose a weighted keyword-document graph, which captures both the semantic relevance between keyword queries and the spatial distance between the resulting documents and the user location. The graph is browsed in a random-walk-with-restart fashion, to select the keyword queries with the highest scores as suggestions. To make our framework scalable, we propose a partition-based approach that outperforms the baseline algorithm by up to an order of magnitude. The appropriateness of our framework and the performance of the algorithms are evaluated using real data. Keyword suggestion in web search helps users to access relevant information without having to know how to precisely express their queries. Existing keyword suggestion techniques do not consider the locations of the users and the query results; i.e., the spatial proximity of a user to the retrieved results is not taken as a factor in the recommendation. However, the relevance of search results in many applications (e.g., location-based services) is known to be correlated with their spatial proximity to the query issuer.

Index Terms:
Query suggestion, spatial databases.

1 INTRODUCTION:

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 emerges due to the popularity of spatial keyword search [12], [13], [14], [15], [16].

Google processed a daily average of 4.7 billion queries in 2011,1 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 descriptions) or geo-documents(i.e., documents associated with geo-locations). Furthermore, 53 percent of Bing's mobile searches in 2011 have a local intent.2 In this paper, we propose the first Location-aware Keyword query Suggestion framework. We illustrate the benefit of LKS using a toy example. Consider five geo-documents d1-d5 as listed in Fig. 1a. Each document di is associated with a location di:_ as shown in Fig. 1b. Assume that a user issues a keyword query kq ¼ "seafood" at location _q, shown in Fig. 1b. Note that the relevant documents d1–d3 (containing "seafood") are far from _q. A location-aware suggestion is "lobster", which can retrieve nearby documents d4 and d5 that are also relevant to the user's original search intention. Previous keyword query suggestion models (e.g., [5]) ignore the user location and would suggest "fish", which again fails to retrieve nearby relevant documents. Note that LKS has a different goal and therefore differs from other location-aware recommendation methods (e.g., auto-completion/instant searches [17], [18], tag recommendation [19]). Section 5 provides a detailed discussion about the differences between LKS and these models, while in Section 4 we experimentally show that an adaptation of the method in [19] is less effective than LKS.



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

	λ_q
d_1	1.0
d_2	0.9
d_3	0.5
d_4	0.1
d_5	0.0

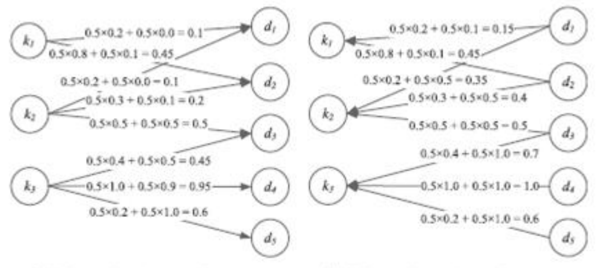
(a) Documents and Keyword Queries (b) Locations of Documents (c) KD-Graph (d) Euclidean Distances

Fig. 1. LKS example.

The first challenge of our LKS framework is how to effectively measure keyword query similarity while capturing the spatial distance factor. In accordance to previous query suggestion approaches [2], [3], [4], [5], [6], [7], [9], [10], LKS constructs and uses a keyword-document bipartite graph (KD-graph for short), which connects the keyword queries with their relevant documents as shown in Fig. 1c. Different to all previous approaches which ignore locations, LKS adjusts the weights on edges in the KD-graph to capture not only the semantic relevance between keyword queries, but also the spatial distance between the document locations and the query issuer's location q.

2 LKS FRAMEWORK:

Consider a user-supplied query q with initial input kq; kq can be a single word or a phrase. Assuming that the query issuer is at location _q, two intuitive criteria for selecting good suggestions are: (i) the suggested keyword queries (words or phrases) should satisfy the user's information needs based on kq and (ii) the suggested queries can retrieve relevant documents spatially close to _q. The proposed LKS framework captures these two criteria. relevant documents spatially close to _q. The proposed LKS framework captures these two criteria.


$0.5\times0.2+0.5\times0.0=0.1$
$0.5\times0.8+0.5\times0.1=0.45$
$0.5\times0.2+0.5\times0.0=0.1$
$0.5\times0.3+0.5\times0.1=0.2$
$0.5\times0.5+0.5\times0.5=0.5$
$0.5\times0.4+0.5\times0.5=0.45$
$0.5\times1.0+0.5\times0.9=0.95$
$0.5\times0.2+0.5\times1.0=0.6$
$0.5\times0.2+0.5\times0.1=0.15$
$0.5\times0.8+0.5\times0.1=0.45$
$0.5\times0.2+0.5\times0.5=0.35$
$0.5\times0.3+0.5\times0.5=0.4$
$0.5\times0.5+0.5\times0.5=0.5$
$0.5\times0.4+0.5\times1.0=0.7$
$0.5\times1.0+0.5\times1.0=1.0$
$0.5\times0.2+0.5\times1.0=0.6$

(a) Keyword queries → Documents (b) Keyword queries ← Documents

Fig. 2. Location-aware edge weight adjustment.

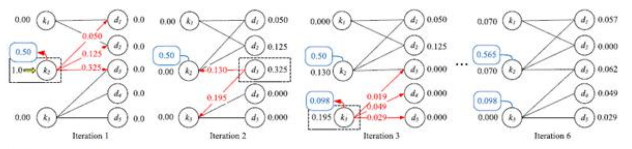


Fig. 3. Illustration of Algorithm BA.

3 ALGORITHMS:

In this section, we introduce a baseline algorithm (BA) for location-aware suggestions (Section 3.1). Then, we propose our efficient partition-based algorithm (Section 3.2).

3.1 Baseline Algorithm (BA):

We extend the popular Bookmark-Coloring Algorithm [25] to compute the RWR-based top-m query suggestions as a baseline algorithm. BCA models RWR as a bookmark coloring process. Starting with one unit of active ink injected into node kq, BA processes the nodes in the graph in descending order of their active ink. Different from typical personalized PageRank problems [27], [28] where the graph is homogeneous, our KD-graph Gq has two types of nodes: keyword query nodes and document nodes. As opposed to BCA, BA only ranks keyword query nodes; a keyword query node retains a portion of its active ink and distributes 1^a portion to its neighbor nodes based on its outgoing adjusted edge weights, while a document node distributes all its active ink to its neighbor nodes.

Algorithm 1. Baseline Algorithm (BA)

```
Input: $G(D,K,E)$, $q=(k_q,\lambda_q)$, $m$, $\epsilon$
Output: $C$
PriorityQueue $Q \leftarrow \emptyset$, $C \leftarrow \emptyset$;
Add $k_q$ to $Q$ with $k_q.aink \leftarrow 1$;
$AINK \leftarrow 1$;
while $Q \neq \emptyset$ and $Q.top.aink \geq \epsilon$ do
    Deheap the first entry $top$ from $Q$;
    $tm$ = the top-$m$ entry from $C$;
    $tm'$ = the top-$(m+1)$ entry from $C$;
    if $tm.rink > tm'.rink + AINK$ then
        break
    $distratio = 1$;
    if $top$ is a keyword query node then
        $distratio = 1 - \alpha$;
        $top.rink \leftarrow top.rink + top.aink \times \alpha$;
        $AINK \leftarrow AINK - top.aink \times \alpha$;
        if there exist a copy $t$ of $top$ in $C$ then
            Remove $t$ from $C$;
            $top.rink \leftarrow top.rink + t.rink$;
        Add $top$ to $C$;
    for each node $v$ connected to $top$ in $G$ do
        $v.aink \leftarrow top.aink \times distratio \times \tilde{w}(top,v)$;
        if there exists a copy $v'$ of $v$ in $Q$ then
            Remove $v'$ from $Q$; $v.aink \leftarrow v.aink + v'.aink$;
        Add $v$ to $Q$;
return the top-$m$ entries (excluding $k_q$) in $C$;
```

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

3.2 Partition-Based Algorithm:

Algorithm BA can be slow for several reasons. First, at each iteration, only one node is processed; thus, the active ink drops slowly and the termination conditions are met after too many iterations. Second, given the large number of iterations, the overhead of maintaining queue Q is significant. Finally, the nodes distribute their active ink to all their neighbors, even if some of them only receive a small amount of ink. To improve the performance of BA, in this section, we propose a partition-based algorithm that divides the keyword queries and the documents in the KD-graph G into groups. Let PK ¼ fPKi g be the partitions of the keyword queries and PD ¼ fPDi g be the document partitions. Algorithm PA follows the basic routine of algorithm BA, but with the following differences:

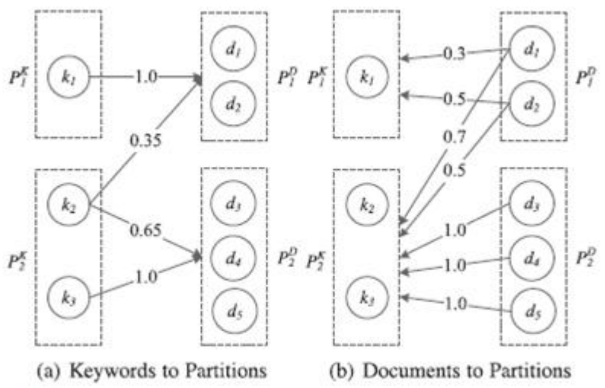


Fig. 4. Node-partition graphs.

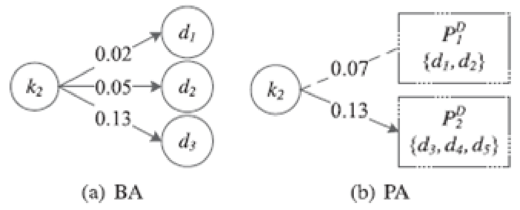


Fig. 5. Ink distribution methods.

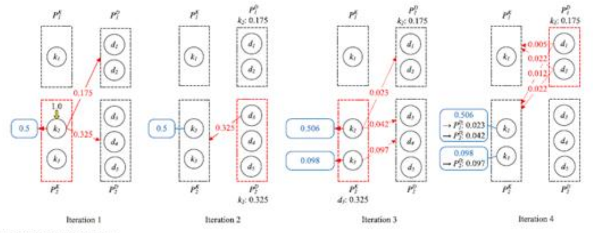


Fig. 6. Illustration of Algorithm PA.

Algorithm 2. PA

```
   Input: $G(D, K, E), G^{KP}, G^{DP}, q = (k_q, \lambda_q), m, \epsilon$
   Output: $C$
 1 PriorityQueue $Q \leftarrow \emptyset, C \leftarrow \emptyset$ ;
 2 Add partition $P \ni k_q$ to $Q$ with $P.aink \leftarrow 1$;
 3 $AINK \leftarrow 1$;
 4 while $Q \neq \emptyset$ and $Q.top.aink_{v_i} \geq \epsilon$ do
 5     Deheap the top entry $P_i$ from $Q$;
 6     $tm$ = the top-$m$ entry from $C$;
 7     $tm'$ = the top-$(m+1)$ entry from $C$;
 8     if $tm.rink > tm'.rink + AINK$ then
 9         break;
10     Spread the active ink to nodes in $P_i$;
11     for each node $v$ in partition $P_i$ do
12         $distratio = 1$ ;
13         if $v$ is a keyword query node then
14             $distratio = 1 - \alpha$ ;
15             $v.rink \leftarrow v.rink + v.aink \times \alpha$;
16             $AINK \leftarrow AINK - v.aink \times \alpha$;
17             if there exist a copy $t$ of $v$ in $C$ then
18                 Remove $t$ from $C$;
19                 $v.rink \leftarrow v.rink + t.rink$;
20             Add $v$ to $C$;
21             Get partition set $\mathcal{P}$ connected from $v$ in $G^{KP}$;
22         else
23             Get partition set $\mathcal{P}$ connected from $v$ in $G^{DP}$;
24         for each partition $P_i$ in $\mathcal{P}$ do
25             $ink \leftarrow v.aink \times distratio \times \tilde{w}(v, P_i)$;
26             if $ink + v.acc.P_i \geq \epsilon$ then
27                 $P_i.aink \leftarrow ink + v.acc.P_i$;
28                 if there exist a copy $P_i'$ of $P_i$ in $Q$ then
29                     Remove $P_i'$ from $Q$;
                       $P_i.aink \leftarrow P_i.aink + P_i'.aink$;
30                 Add $P_i$ to $Q$;
31             else
32                 Accumulate $ink$ at node $v$ for $P_i$ ($v.acc.P_i$);
33 return the top-$m$ entries (excluding $k_q$) in $C$;
```

TABLE 1
Parameters

Description	Parameter	Values	Default Value
Number of partitions	N	1,4,16,64,256	16
RWR probability	α	0.2,0.35,0.5, 0.65,0.8	0.5
Edge weight adjustment param.	β	0,0.25,0.5, 0.75,1	0.5
Threshold ($\times 10^{-5}$)	ϵ	0.1,0.5,1, 5,10	1
Number of documents (M)	$\|D\|$	0.5, 1, 1.5, 2, 2.5	1.5

CONCLUSION:

In this paper, we 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 [25] is introduced to solve the problem.

Then, we proposed a partition-based algorithm 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. Finally, we believe that PA can also be applied to accelerate RWR on general graphs with dynamic edge weights; we will investigate this potential in the future.

REFERENCES:

[1] R. Baeza-Yates, C. Hurtado, and M. Mendoza, “Query recommendation using query logs in search engines,” in Proc. Int. Conf. Current Trends Database Technol., 2004, pp. 588–596.

[2] D. Beeferman and A. Berger, “Agglomerative clustering of a search engine query log,” in Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2000, pp. 407–416.

[3] 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 Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2008, pp. 875–883.

[4] N. Craswell and M. Szummer, “Random walks on the click graph,” in Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2007, pp. 239–246.

[5] Q. Mei, D. Zhou, and K. Church, “Query suggestion using hitting time,” in Proc. 17th ACM Conf. Inf. Knowl. Manage., 2008, pp. 469–478.

[6] Y. Song and L.-W. He, “Optimal rare query suggestion with implicit user feedback,” in Proc. 19th Int. Conf. World Wide Web, 2010, pp. 901–910.

[7] T. Miyanishi and T. Sakai, “Time-aware structured query suggestion,” in Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2013, pp. 809–812.

[8] A. Anagnostopoulos, L. Becchetti, C. Castillo, and A. Gionis, “An optimization framework for query recommendation,” in Proc. ACM Int. Conf. Web Search Data Mining, 2010, pp. 161–170.

[9] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna, “The query-flow graph: Model and applications,” in Proc. 17th ACM Conf. Inf. Knowl. Manage., 2008, pp. 609–618.

[10] Y. Song, D. Zhou, and L.-w. He, “Query suggestion by constructing term-transition graphs,” in Proc. 5th ACM Int. Conf. Web Search Data Mining, 2012, pp. 353–362.

[11] 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,” World Wide Web, vol. 16, pp. 273–297, 2013.

[12] 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, pp. 7:1–7:47, 2013.

[13] D. Wu, G. Cong, and C. S. Jensen, “A framework for efficient spatial web object retrieval,” VLDB J., vol. 21, no. 6, pp. 797–822, 2012.

[14] J. Fan, G. Li, L. Zhou, S. Chen, and J. Hu, “SEAL: Spatio-textual similarity search,” Proc. VLDB Endowment, vol. 5, no. 9, pp. 824– 835, 2012.

[15] P. Bouros, S. Ge, and N. Mamoulis, “Spatio-textual similarity joins,” Proc. VLDB Endowment, vol. 6, no. 1, pp. 1–12, 2012.

[16] Y. Lu, J. Lu, G. Cong, W. Wu, and C. Shahabi, “Efficient algorithms and cost models for reverse spatial-keyword k-nearest neighbor search,” ACM Trans. Database Syst., vol. 39, no. 2, pp. 13:1–13:46, 2014.

[17] S. Basu Roy and K. Chakrabarti, “Location-aware type ahead search on spatial databases: Semantics and efficiency,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 361–372.

[18] R. Zhong, J. Fan, G. Li, K.-L. Tan, and L. Zhou, “Location-aware instant search,” in Proc. 21st ACM Conf. Inf. Knowl. Manage., 2012, pp. 385–394.

[19] I. Miliou and A. Vlachou, “Location-aware tag recommendations for flickr,” in Proc. 25th Int. Conf. Database Expert Syst. Appl., 2014, pp. 97–104.

[20] H. Tong, C. Faloutsos, and J.-Y. Pan, “Fast random walk with restart and its applications,” in Proc. 6th Int. Conf. Data Mining, 2006, pp. 613–622.

[21] Y. Fujiwara, M. Nakatsuji, M. Onizuka, and M. Kitsuregawa, “Fast and exact top-k search for random walk with restart,” Proc. VLDB Endowment, vol. 5, no. 5, pp. 442–453, Jan. 2012.

[22] D. Fogaras, B. R_acz, K. Csalog_any, and T. Sarl_os, “Towards scaling fully personalized PageRank: Algorithms, lower bounds, and experiments,” Internet Math., vol. 2, no. 3, pp. 333–358, 2005.

[23] B. Bahmani, A. Chowdhury, and A. Goel, “Fast incremental and personalized PageRank,” Proc. VLDB Endowment, vol. 4, no. 3, pp. 173–184, Dec. 2010.

[24] K. Avrachenkov, N. Litvak, D. Nemirovsky, E. Smirnova, and M. Sokol, “Quick detection of top-k personalized Page Rank lists,” in Proc. 8th Int. Workshop Algorithms Models Web Graph, 2011, vol. 6732, pp. 50–61.