K-SPIN: Efficiently Processing Spatial Keyword Queries on Road Networks
(Extended Abstract)

Tenindra Abeywickrama
Grab-NUS AI Laboratory
National University of Singapore
Singapore
tenindra@nus.edu.sg

Muhammad Aamir Cheema
Faculty of Information Technology
Monash University
Melbourne, Australia
aamir.cheema@monash.edu

Arijit Khan
School of Computer Science and Engineering
Nanyang Technological University
Singapore
arijit.khan@ntu.edu.sg

***Abstract*—Given the prevalence and volume of local search queries, today's search engines are required to find results by both spatial proximity and textual relevance at high query throughput. Existing techniques to answer such *spatial keyword queries* employ a keyword aggregation strategy that suffers from certain drawbacks when applied to road networks. Instead, we propose the K-SPIN framework, which uses an alternative *keyword separation* strategy that is more suitable on road networks. While this strategy was previously thought to entail prohibitive pre-processing costs, we further propose novel techniques to make our framework viable and even light-weight. Thorough experimentation shows that K-SPIN outperforms the state-of-the-art by up to two orders of magnitude on a wide range of settings and real-world datasets.**

***Index Terms*—Road networks, points of interest search, spatio-textual queries, network Voronoi diagrams**

I. Introduction

FINDING the nearest relevant points of interest (POIs) to a user is an important type of query in map-based services [1]. A *spatial keyword* query retrieves POIs that are close to a user's location (e.g., in terms of travel time) with textual descriptions that are relevant to query keywords provided by the user. Boolean kNN (BkNN) queries retrieve the closest POIs that satisfy certain keyword constraints. Constraints may be disjunctive (POIs contain *any* query keyword) or conjunctive (contain *all* query keywords). On the other hand, top-k spatial keyword queries return k POIs with the best scores, where the score of a POI combines the POI's proximity to the query location and the relevance of the POI's textual description to the query keywords.

A popular search engine like Google experiences ≈2500 search queries with a location component every second on average [2]. Measuring proximity of POIs using road network distance is more accurate and supports various metrics, for example, travel-time via the road network is more accurate than Euclidean distance (i.e., "as-the-crow-flies"). However, existing techniques on road networks use an inefficient keyword aggregation strategy that severely hinders their ability to meet such high throughput requirements.

Keyword aggregation is a technique used extensively in Euclidean spatial keyword techniques [1] that involves summarizing keyword occurrences over geographical regions. Spatial keyword queries are then answered by searching the most promising regions first while pruning regions that cannot contain results. The drawback of this approach is the generation of many false-positives when regions appear promising but in reality, contain no results. Whenever a candidate POI is encountered, its distance from the query must be computed to confirm if it is close and relevant enough. Computing distance in Euclidean space is a quick arithmetic operation, but in road networks, it is a complex graph operation and far more expensive. Consequently, the penalty paid for incurring false-positives in road networks is significantly higher than in Euclidean space. Ultimately, this makes keyword aggregation far less effective for road networks.

The problems encountered cannot be solved in straightforwards ways due to the permanent loss of discriminating information that results from aggregation. Detailed examples of how costly false-positives occur are provided in the full version of the paper [3]. To overcome these challenges we present the **K**eyword **Sep**arated **In**dexing (K-SPIN) framework employing an alternative *keyword separation* strategy, which creates a separate index for each keyword. While this approach may initially seem to entail prohibitive pre-processing costs, our techniques make it not only viable but also light-weight.

II. Solution Overview

The modules that compose the K-SPIN framework are shown in Figure 1. Here we briefly describe each module and how they interact to efficiently answer spatial keyword queries.

1. Lower Bounding Module. This module computes a lower-bound network distance between any two vertices using selected heuristics. For example, a lower-bound can be obtained using landmarks as in the ALT [4] index. ALT pre-computes network distances between some chosen landmark vertices and all vertices in the graph then uses the triangle inequality to obtain a lower-bound network distance between any two vertices. Moreover, multiple heuristics can be incorporated to obtain the tightest lower-bound network distance overall.

2. Network Distance Module. This module computes the exact network distance between any two given vertices in a graph. *Any* Road Network Index technique can be used to compute network distance, e.g., extremely fast 2-hop labels [5]. The system administrator may choose a technique based on its efficiency and/or index size or may simply choose the

2375-026X/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDE48307.2020.00237

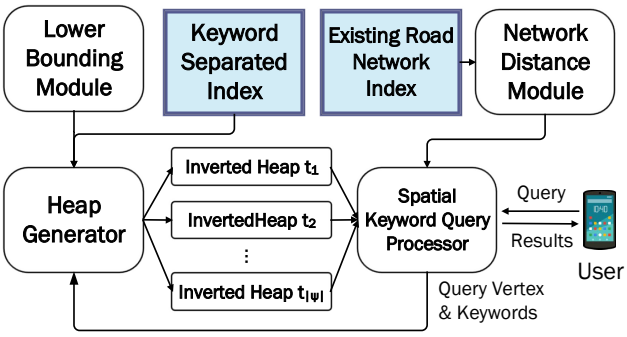

Fig. 1. Keyword Separated Indexing (K-SPIN) Framework

techniques already being used to answer other queries. This module is the bottleneck as network distance computations are the most expensive operation performed for a POI.

3. Heap Generator. The Heap Generator is responsible for creating and maintaining the *on-demand inverted heaps*. An on-demand inverted heap for a particular keyword t satisfies the following property at any point in time (i.e., when the heap is first created and whenever a heap element is extracted).

Property 1: Given the current top object o in inverted heap $\mathcal{H}$ for keyword t and its lower-bound distance $LB(q, o)$ from query vertex q; any object o_t containing t, not yet extracted from $\mathcal{H}$, has network distance $d(q, o_t) \geq LB(q, o)$.

Property 1 allows our query algorithms to access objects associated with a particular keyword t in order of their lower-bound network distances from q computed using the *Lower Bounding Module*. To efficiently create and maintain an inverted heap, the Heap Generator utilizes a *Keyword Separated Index* (KSI) that indexes $inv(t)$ for each keyword t in corpus W where $inv(t)$ is the set of all objects associated with t. Property 1 allows the heap to be populated lazily, i.e., objects are added incrementally such that the property is met.

Network Voronoi Diagrams (NVDs) are the first choice data structure to accurately generate candidate objects [6]. However, employing them as our KSI leads to impractical pre-processing costs in both space and time. Using several key observations (detailed in the full paper [3]), we propose an approximate NVD for use as our KSI with significantly reduced pre-processing time and space by up to an order of magnitude. Moreover, this comes at a small theoretically bounded cost to query efficiency and still returns exact results.

4. Query Processor. The Query Processor contains algorithms to answer various spatial keyword queries. Algorithms use on-demand inverted heaps to retrieve relevant candidate objects. The challenge lies in deciding which heap to use and how to filter poor candidates using an effective lower-bound score. Hence the efficiency of the Query Processor is critical in avoiding the false-positive problems of existing methods mentioned earlier. The Query Processor uses the *Network Distance Module* to compute the network distances between the query vertex and the filtered candidate objects.

We propose query algorithms to answer both BkNN and top-k spatial keyword queries. Our techniques carefully exploit Property 1 to generate fewer false-positive candidates. In particular, we propose the idea of a pseudo-lower bound score for our top-k query algorithm that infers details about unseen objects in inverted heaps to retrieve more promising candidates and terminate sooner. Furthermore, we prove that even though the pseudo-lower bound is not a real lower-bound, our algorithm still retrieves correct results [3].

Technique	Index Size (in GB)	Queries/second	
		Top-k	BkNN
K-SPIN [Our Method] + CH [7]	0.6 + 0.6	865	1021
K-SPIN [Our Method] + PHL [5]	0.6 + 15.8	3942	9869
Spatial Keyword G-tree [8]	2.7	266	178
ROAD [9]	4.5	83	✗

TABLE I
COMPARISON OF INDEX SIZE AND THROUGHPUT (# OF QUERIES PROCESSED PER SECOND) ON US ROAD NETWORK DATASET

III. Analysis & Conclusions

We conduct an extensive experimental investigation of K-SPIN using a variety of settings, parameters, and variables with real-world road network and POI datasets. Table I depicts spatial keyword query performance and index size for our techniques and competing methods on the US dataset with default settings [3]. As shown, our techniques support significantly higher throughput, even when using space-efficient but slower road network indexes such as Contraction Hierarchies (CH) [7]. The significant improvement of K-SPIN over competing methods shows that keyword separation is a more effective alternative to keyword aggregation on road networks.

Acknowledgment

We sincerely thank Hanan Samet for his insightful comments. The research of Muhammad Aamir Cheema is supported by ARC DP180103411 and FT180100140. Arijit Khan is supported by MOE Tier-1 RG83/16 and NTU M4081678. Tenindra Abeywickrama was supported by an Australian Government RTP Scholarship.

References

[1] L. Chen, G. Cong, C. S. Jensen, and D. Wu, "Spatial Keyword Query Processing: An Experimental Evaluation," in *PVLDB*, 2013, pp. 217–228.
[2] G. Sterling. (2015) http://screenwerk.com/2015/05/11/data-suggest-that-local-intent-queries-nearly-half-of-all-search-volume/.
[3] T. Abeywickrama, M. A. Cheema, and A. Khan, "K-spin: Efficiently processing spatial keyword queries on road networks," *IEEE Trans. Knowl. Data Eng.*, 2019, to appear.
[4] A. V. Goldberg and C. Harrelson, "Computing the Shortest Path: A* Search Meets Graph Theory," in *SODA*, 2005, pp. 156–165.
[5] T. Akiba, Y. Iwata, K.-i. Kawarabayashi, and Y. Kawata, "Fast Shortest-path Distance Queries on Road Networks by Pruned Highway Labeling," in *ALENEX*, 2014, pp. 147–154.
[6] T. Abeywickrama and M. A. Cheema, "Efficient Landmark-Based Candidate Generation for kNN Queries on Road Networks," in *DASFAA*, 2017, pp. 425–440.
[7] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, "Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks," in *WEA*, 2008, pp. 319–333.
[8] R. Zhong, G. Li, K. Tan, L. Zhou, and Z. Gong, "G-Tree: An Efficient and Scalable Index for Spatial Search on Road Networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 8, pp. 2175–2189, 2015.
[9] K. C. K. Lee, L. W.-Chien, Z. Baihua, and T. Yuan, "ROAD: A New Spatial Object Search Framework for Road Networks," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 3, pp. 547–560, 2012.