BEST KEYWORD COVER SEARCH

Sirigineedi. Venu Gopal Durga Sai , Miss G.Keerthana, Sri.V.Bhaskara Murthy
MCA Student, Assistant Professor, Associate Professor
Dept Of MCA
B.V.Raju College, Bhimavaram

ABSTRACT

It is common that the objects in a spatial database (e.g., restaurants/hotels) are associated with keyword(s) to indicate their businesses/services/features. An interesting problem known as Closest Keywords search is to query objects, called keyword cover, which together cover a set of query keywords and have the minimum inter-objects distance. In recent years, we observe the increasing availability and importance of keyword rating in object evaluation for the better decision making. This motivates us to investigate a generic version of Closest Keywords search called Best Keyword Cover which considers inter-objects distance as well as the keyword rating of objects. The baseline algorithm is inspired by the methods of Closest Keywords search which is based on exhaustively combining objects from different query keywords to generate candidate keyword covers. When the number of query keywords increases, the performance of the baseline algorithm drops dramatically as a result of massive candidate keyword covers generated. To attack this drawback, this work proposes a much more scalable algorithm called keyword nearest neighbor expansion (keyword-NNE). Compared to the baseline algorithm, keyword-NNE algorithm significantly reduces the number of candidate keyword covers generated. The in-depth analysis and extensive experiments on real data sets have justified the superiority of our keyword-NNE algorithm.

I. INTRODUCTION

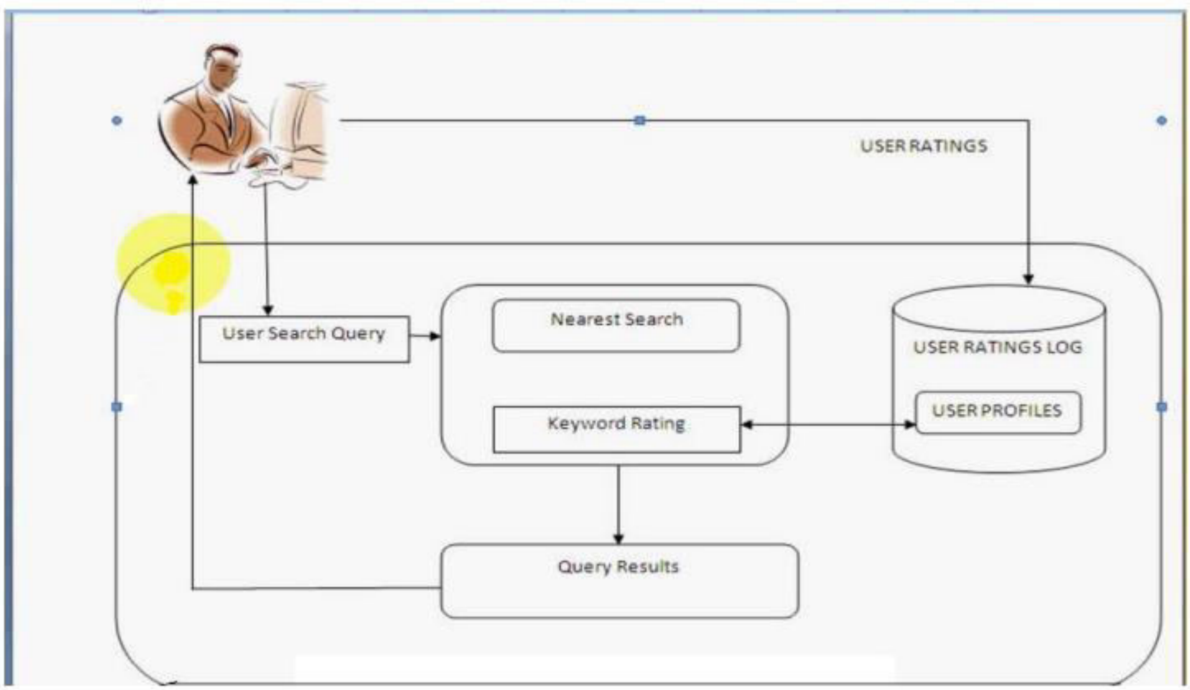
Driven by mobile computing, location-based services and wide availability of extensive digital maps and satellite imagery (e.g., Google Maps and Microsoft Virtual Earth services), the spatial keywords search problem has attracted much attention recently. In a spatial database, each tuple represents a spatial object which is associated with keyword(s) to indicate the information such as its businesses/services/features. Given a set of query keywords, an essential task of spatial keywords search is to identify spatial object(s) which are associated with keywords relevant to a set of query keywords, and have desirable spatial relationships (e.g., close to each other and/or close to a query location).

This problem has unique value in various applications because users' requirements are often expressed as multiple keywords. For example, a tourist who plans to visit a city may have particular shopping, dining and accommodation needs. It is desirable that all these needs can be satisfied without long distance traveling. Due to the remarkable value in practice, several variants of spatial keyword search problem have been studied.

The works aim to find a number of individual objects, each of which is close to a query location and the associated keywords (or called document) are very relevant to a set of query keywords (or called query document). The document similarity is applied to measure the relevance between two sets of keywords. Since it is likely none of individual objects is associated with all query keywords, this motivates the studies to retrieve multiple objects, called keyword cover, which together cover (i.e., associated with) all query keywords and are close to each other. This problem is known as m Closest Keywords (mCK) query in.

The problem studied in [4] additionally requires the retrieved objects close to a query location. This paper investigates a generic version of mCK query, called Best Keyword Cover (BKC) query, which considers inter-objects distance as well as keyword rating. It is motivated by the observation of increasing availability and importance of keyword rating in decision making.

II. SYSTEM ARCHITECTURE



III. EXISTING SYSTEM

This work develops two BKC query processing algorithms, baseline and keyword-NNE. The baseline algorithm is inspired by the mCK query processing methods. Both the baseline algorithm and keyword-NNE algorithm are supported by indexing the objects with an R*-tree like index, called KRR*-tree. In the baseline algorithm, the idea is to combine nodes in higher hierarchical levels of KRR*-trees to generate candidate keyword covers. Then, the most promising candidate is assessed in priority by combining their child nodes to generate new candidates. Even though BKC query can be effectively resolved, when the number of query keywords increases, the performance drops dramatically as a result of massive candidate keyword covers generated.

IV. PROPOSED SYSTEM

To overcome this critical drawback, we developed much scalable keyword nearest neighbor expansion (keyword- NNE) algorithm which applies a different strategy. Keyword-NNE selects one query keyword as principal query keyword. The objects associated with the principal query keyword are principal objects. For each principal object, the local best solution (known as local best keyword cover (lbkc)) is computed. Among them, the lbkc with the highest evaluation is the solution of BKC query. Given a principal object, its lbkc can be identified by simply retrieving a few nearby and highly rated objects in each non-principal query keyword (2-4 objects in average as illustrated in experiments). Compared to the baseline algorithm, the number of candidate keyword covers generated in keyword-NNE algorithm is significantly reduced. The indepth analysis reveals that the number of candidate keyword covers further processed in keyword-NNE algorithm is optimal, and each keyword candidate cover processing generates much less new candidate keyword covers than that in the baseline algorithm.

V. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Modules:

After careful analysis the system has been identified to have the following modules:

1. **Spatial Database Module**
2. **Point of Interests Module**
3. **Keyword Cover Search Module**
4. **Baseline Vs Keyword-NNE Alogorithm Module**

1.Spatial Database Module:

Recently, the spatial keyword search has received considerable attention from research community. Some existing works focus on retrieving individual objects by specifying a query consisting of a query location and a set of query keywords (or known as document in some context). Each retrieved object is associated with keywords relevant to the query keywords and is close to the query location. The similarity between documents are applied to measure the relevance between two sets of keywords. Since it is likely no individual object is associated with all query keywords, some other works aim to retrieve multiple objects which together cover all query keywords. While potentially a large number of object combinations satisfy this requirement, the research problem is that the retrieved objects must have desirable spatial relationship. Authors put forward the problem to retrieve objects which 1) cover all query keywords, 2) have minimum inter-objects distance and 3) are close to a query location. The work study a similar problem called m Closet Keywords (mCK). mCK aims to find objects which cover all query keywords and have the minimum inter-objects distance. Since no query location is asked in mCK, the search space in mCK is not constrained by the query location. The problem studied in this paper is a generic version of mCK query by also considering keyword rating of objects.

2. Point of Interests Module

The goal of the interface is to provide point of interest information (static and dynamic ones) with, at least, a location, some mandatories attributes and optionals details (description,). In order to provide those information, the component that implements the interface uses the map database information to locate and display point of interest (POI) or to select a POI as route waypoint and favourite. This component not only provides search functionalities for the local database but also a way to connect external search engine to this component and enhance the search criteria and the list of results It also proposes a solution to get custom POIs (not part of the local map database) or to dynamically update content and description of local POI.

This is achieved by specifying and providing interfaces to:

- Select POIs from one of their attributes (e.g., Category, Name,…)
- Retrieve POI attributes (e.g., Location and Description)
- Get dynamic content for a given POI.
- Add custom POI to the map display
- Import new POIs and POIs categories from local file.

3. Keyword Cover Search Module

Given a spatial database, each object may be associated with one or multiple keywords. Without loss of generality, the object with multiple keywords are transformed to multiple objects located at the same location, each with a distinct single keyword.When further processing a candidate keyword cover, keyword-NNE algorithm typically generates much less new candidate keyword covers compared to BF-baseline algorithm. Since the number of candidate keyword covers further processed in keyword-NNE algorithm is optimal the number of keyword covers generated in BF-baseline algorithm is much more than that in keyword- NNE algorithm. In turn, we conclude that the number of keyword covers generated in baseline algorithm is much more than that in keyword-NNE algorithm. This conclusion is independent of the principal query keyword since the analysis does not apply any constraint on the selection strategy of principal query keyword.

4.Baseline Vs Keyword-NNE Algorithm Module

BaseLine Algorithm:

BF-baseline algorithm is not feasible in practice. The main reason is that BF-baseline algorithm requires to maintain H in memory. The peak size of H can be very large because of the exhaustive combination until the first current best solution bkc is obtained. To release the memory bottleneck,

the depth-first browsing strategy is applied in the baseline algorithm such that the current best solution is obtained as soon as possible. Compared to the best-first browsing strategy which is global optimal, the depth-first browsing strategy is a kind of greedy algorithm
which is local optimal. As a consequence, if a candidate keyword cover kc has kc:score > bkc:score, kc is further processed by retrieving the child nodes of kc and combining them to generate more candidates. Note that bkc:score increases from 0 to BKC:score in the baseline algorithm. Therefore, the candidate keyword covers which are further processed in the baseline algorithm can be much more than that in BF-baseline algorithm. Given a candidate keyword cover kc, it is further processed in the same way in both the baseline algorithm and BF-baseline algorithm, i.e., retrieving the child nodes of kc and combine them to generate more candidates using Generate Candidate function in Algorithm. Since the candidate keyword covers further processed in the baseline algorithm can be much more than that in BF-baseline algorithm, the total candidate keyword covers generated in the baseline algorithm can be much more than that in BF- baseline algorithm. Note that the analysis captures the key characters of the baseline algorithm in BKC query processing which are inherited from the methods for mCK query processing.

Keyword-NNE Algorithm:

In keyword-NNE algorithm, the best-first browsing strategy is applied like BF-baseline but large memory requirement is avoided. For the better explanation, we can imagine all candidate keyword covers generated in BF-baseline algorithm are grouped into independent groups. Each group is associated with one principal node (or object). That is, the candidate keyword covers fall in the same group if they have the same principal node (or object). Given a principal node Nk, let GNk be the associated group. The example in Figure 5 shows GNk where some keyword covers such as kc1; kc2 have score greater than BKC:score, denoted as G1 Nk, and some keyword covers such as kc3; kc4 have score not greater than BKC:score, denoted as G2 Nk. In
BF-baseline algorithm, GNk is maintained in H before the first current best solution is obtained, and every keyword cover in G1 Nk needs to be further processed. In keyword-NNE algorithm, the keyword cover in GNk with the highest score, i.e., lbkcNk, is identified and maintained in memory. That is, each principal node (or object) keeps its lbkc only.

VI. CONCLUSION

Compared to the most relevant mCK query, BKC query provides an additional dimension to support more sensible decision making. The introduced baseline algorithm
is inspired by the methods for processing mCK query. The baseline algorithm generates a large number of candidate keyword covers which leads to dramatic performance drop when more query keywords are given. The proposed keyword-NNE algorithm applies a different processing strategy, i.e., searching local best solution for each object in a certain query keyword. As a consequence, the number of

candidate keyword covers generated is significantly reduced. The analysis reveals that the number of candidate keyword covers which need to be further processed in keyword-NNE algorithm is optimal and processing each keyword candidate cover typically generates much less new candidate keyword covers in keyword-NNE algorithm than in the baseline algorithm.

REFERENCES

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in Proc. 20th Int. Conf. Very Large Data Bases, 1994, pp. 487–499.

[2] T. Brinkhoff, H. Kriegel, and B. Seeger, "Efficient processing of spatial joins using r-trees," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1993, pp. 237–246.

[3] X. Cao, G. Cong, and C. Jensen, "Retrieving top-k prestige-based relevant spatial web objects," Proc. VLDB Endowment, vol. 3, nos. 1/2, pp. 373–384, Sep. 2010.

[4] X. Cao, G. Cong, C. Jensen, and B. Ooi, "Collective spatial keyword querying," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 373–384.

[5] G. Cong, C. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial web objects," Proc. VLDB Endowment, vol. 2, no. 1, pp. 337–348, Aug. 2009.

[6] R. Fagin, A. Lotem, and M. Naor, "Optimal aggregation algorithms for middleware," J. Comput. Syst. Sci., vol. 66, pp. 614–656, 2003.

[7] I. D. Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," in Proc. IEEE 24th Int. Conf. Data Eng., 2008, pp. 656–665.

[8] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, "Processing spatialkeyword (SK) queries in geographic information retrieval (GIR) systems," in Proc. 19th Int. Conf. Sci. Statist. Database Manage., 2007, pp. 16–23.

[9] G. R. Hjaltason and H. Samet, "Distance browsing in spatial databases," ACM Trans. Database Syst., vol. 24, no. 2, pp. 256–318, 1999.

[10] Z. Li, K. C. Lee, B. Zheng, W.-C. Lee, D. Lee, and X. Wang, "IRTree: An efficient index for geographic document search," IEEE Trans. Knowl. Data Eng., vol. 99, no. 4, pp. 585–599, Apr. 2010.

[11] N. Mamoulis and D. Papadias, "Multiway spatial joins," ACM Trans. Database Syst., vol. 26, no. 4, pp. 424–475, 2001.

[12] D. Papadias, N. Mamoulis, and B. Delis, "Algorithms for querying by spatial structure," in Proc. Int. Conf. Very Large Data Bases, 1998, pp. 546–557