BEST KEYWORD COVER SEARCH USING KEYWORD-NNE ALGORITHM

K Pushpa Rani, M Jhansi, Chandrasekhara Reddy T
MLR Institute of Technology, Hyderabad, India

ABSTRACT

It is normal that the objects in a spatial database (e.g., ATM’s/Hospitals/Hotels) are related with keyword(s) to demonstrate their business and services. An intriguing issue 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 neighbour 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.

Key words: Spatial Database, Point of Interests, Keywords, Keyword Rating, Keyword Cover.


1. INTRODUCTION

Generally, data mining (sometimes called data or knowledge discovery) is the process of analysing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analysing data. It allows users to analyse data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. How Data Mining Works? While large-scale information
technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. [1] Data mining software analyses relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks. [2]

2. BACKGROUND STUDY

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration are taken into account for developing the proposed systems.

3. EXISTING METHODOLOGY

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. Some existing approaches employ a hybrid index that augments nodes in non-leaf nodes of an R/R* tree with inverted indexes. [3] In virtual bR* tree based method, an R*-tree is used to index locations of objects and an inverted index is used to label the leaf nodes in the R*-tree associated with each keyword. Since only leaf nodes have keyword information the mCK query is processed by browsing index bottom-up. [4] [5]

4. PROPOSED SYSTEM

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. Millions of businesses/services/features around the world have been rated by customers through online business review sites such as Yelp, Citysearch, ZAGAT and Dianping, etc. This work develops two BKC query processing algorithms, baseline and keyword-NNE. [6] 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. We developed much scalable keyword nearest neighbour 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. [7]

4.1. The Merits of the Proposed Methodology

Compared to the baseline algorithm, the number of candidate keyword covers generated in keywordNNE algorithm is significantly reduced. The in-depth analysis reveals that the number of candidate keyword covers further processed in keywordNNE algorithm is optimal, and each keyword candidate cover processing generates much less new candidate keyword covers than that in the baseline algorithm. 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. [8] [9]

![System Architecture](http://www.iaeme.com/IJM/MET/index.asp)

**Figure 1** System Architecture

The input design is the link between the information system and the user shown in fig.1. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

### 4.2. Keyword nearest Neighbour Expansion (Keyword-NNE)

Definition (Keyword Nearest Neighbor (Keyword-NN)): Given a set of query keywords T, the principal query keyword is $k \in T$ and a non-principal query keyword is $ki \in T \setminus \{k\}$. Ok is the set of principal objects and Oki is the set of objects of keyword $ki$. The keyword nearest neighbor of a principal object $ok \in$ Ok in keyword $ki$ is $oki \in$ Oki iif $\{ok, oki\}.score \geq \{ok, o0 ki\}.score$ for all $0 \leq ki \in$ Oki. The first keyword-NN of $ok$ in keyword $ki$ is denoted as $ok.nn1 ki$ and the second keyword-NN is $ok.nn2 ki$, and so on. These keyword-NNs can be retrieved by browsing KRR*ki-tree. Let $Nki$ be a node in KRR*ki-tree. $\{ok, Nki\}.score = score(A, B)$. (6) $A = dist(ok, Nki.)$ $B = min(Nki.maxrating, ok.rating)$. where $dist(ok, Nki)$ is the minimum distance between ok and Nki in the 2-dimensional geographical space defined by x and y dimensions, and $Nki.maxrating$ is the maximum value of $Nki$ in keyword rating dimension. [9]
Algorithm for Keyword-NNE

Step 1. One query keyword \( k \in T \) is selected as the principal query keyword;
Step 2. For each principal object \( \text{ok} \in \text{Ok} \), \( \text{lbkcok} \) is computed;
Step 3. In \( \text{Ok} \), \( \text{GBKCK} \) is identified;
Step 4. return \( \text{GBKCK} \).

5. RESULTS AND DISCUSSION

This section shows the results of the keyword cover search using Keyword-NNE algorithm. Fig.2. shows the Home page of Best Keyword Cover Search. Indexing Keyword Ratings A single tree structure is used to index objects of different keywords.

Figure 2 Home page

The single tree can be extended with an additional dimension to index keyword rating. A single tree structure suits the situation that most keywords are query keywords. Fig.3 shows the admin login page.

Figure 3 Admin Login

Fig.4 shows the user login page where the new user can able to register here and the existing users can login in this page.
In this situation, a single tree is poor to approximate the spatial relationship between objects. Given a principal node \( N_k \), let \( G_{N_k} \) be the associated group. The example in Figure 5 shows \( G_{N_k} \) where some keyword covers such as \( k_{c1}, k_{c2} \) have score greater than \( B_{KC} \cdot \text{score} \), denoted as \( G_{2 \cdot N_k} \). In BF-baseline algorithm, \( G_{N_k} \) is maintained in H before the first current best solution is obtained, and every keyword cover in \( G_{1 \cdot N_k} \) needs to be further processed.

For example, in the experiments, only less than 5 percent keywords are query keywords. 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. User can request his point of interest in the point of interest page shown in Figure 5. 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). [10] That is, the candidate keyword covers fall in the same group if they have the same principal node (or object). Given a principal node \( N_k \), let \( G_{N_k} \) be the associated group. The example in Figure 5 shows \( G_{N_k} \) where some keyword covers such as \( k_{c1}, k_{c2} \) have score greater than \( B_{KC} \cdot \text{score} \), denoted as \( G_{1 \cdot N_k} \), and some keyword covers such as \( k_{c3}, k_{c4} \) have score not greater than \( B_{KC} \cdot \text{score} \), denoted as \( G_{2 \cdot N_k} \). In BF-baseline algorithm, \( G_{N_k} \) is maintained in H before the first current best solution is obtained, and every keyword cover in \( G_{1 \cdot N_k} \) needs to be further processed. [10]
In the fig.6 shows some results of all keywords, i.e., “hotel”, “restaurant” and “bar”, are query keywords.

![Best Keyword Cover Search](image.png)

**Figure 6** Keyword Ratings

### 6. CONCLUSIONS

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.

### REFERENCES


