Proceedings of National Conference on Recent Advances in Computer Science & Engineering (NCRACSE-2017)

International Journal for Modern Trends in Science and Technology
Volume: 03, Special Issue No: 01, February 2017
ISSN: 2455-3778
http://www.ijmtst.com

Efficient Adjacent Neighbor Expansion Search Keyword

Samuel Chepuri[1] | Yallamati Prakasarao[2] | Jagajeevan Nandigama[3]

[1]Assistant Professor, Department of CSE, KMIT, Hyderabad, India.
[2]Assistant Assistant Professor, Dept of CSE, VISIT Engineering College, Tadepalligudem, West Godhavari, A.P., India.
[3]Assistant Assistant Professor, Dept of CSE, CIET, Guntur, A.P., India.

To Cite this Article
Samuel Chepuri, Yallamati Prakasarao and Jagajeevan Nandigama, "Efficient Adjacent Neighbor Expansion Search Keyword", *International Journal for Modern Trends in Science and Technology*, Vol. 03, Special Issue 01, 2017, pp. 92-97.

ABSTRACT

Spatial databases store the information about the spatial objects which are associated with the keywords to indicate the information such as its business/services/features. Very important problem known as closest keywords search is to query objects, called keyword cover. In closest keyword search, it covers a set of query keywords and minimum distance between objects. From last few years, keyword rating increases its availability and importance in object evaluation for the decision making. This is the main reason for developing this new algorithm called best keyword cover which is considers interdistance as well as the rating provided by the customers through the online business review sites. Closest keyword search algorithm combines the objects from different query keywords to generate candidate keyword covers. Baseline algorithm and keyword nearest neighbor expansion algorithms are used to find the best keyword cover. The performance of the closest keyword algorithm drops dramatically, when the number of query keyword increases. To solve this problem of the existing algorithm, this work proposes generic version called keyword adjacent neighbor expansion which reduces the resulted candidate keyword covers.

Copyright © 2017 International Journal for Modern Trends in Science and Technology
All rights reserved.

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. The problem studied in additionally requires the retrieved objects close to a query location

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.

II. Literature Survey

A geographic query [1] that is composed of query keywords and a location, a geographic search engine retrieves documents that are the most textually and spatially relevant to the query keywords and the location, respectively, and ranks the retrieved documents according to their joint textual and spatial relevance's to the query. The lack of an efficient index that can simultaneously handle both the textual and spatial aspects of the documents makes existing geographic search engines inefficient geographic queries. In this paper, we propose an efficient index, called IR-tree, that together with a top-k document search algorithm facilitates four major tasks in document searches, namely, 1) spatial filtering, 2) textual filtering, 3) relevance computation, and 4) document ranking in a fully integrated manner. In addition, IR-tree allows searches to adopt different weights on textual and spatial relevance of documents at the runtime and thus caters for a wide variety of applications. A set of comprehensive experiments over a wide range of scenarios has been conducted and the experiment results demonstrate that IR-tree outperforms the state-of-theart approaches for geographic document searches.

The location-aware keyword query[2] returns ranked objects that are near a query location and that have textual descriptions that match query keywords. This query occurs inherently in many types of mobile and traditional web services and applications, e.g., Yellow Pages and Maps services. Previous work considers the potential results of such a query as being independent when ranking them. However, a relevant result object with nearby objects that are also relevant to the query is likely to be preferable over a relevant object without relevant nearby objects. The paper proposes the concept of prestige-based relevance to capture both the textual relevance of an object to a query and the effects of nearby objects. Based on this, a new type of query, the Location-aware top-k Prestige-based Text retrieval (LkPT) query, is proposed that retrieves the top-k spatial web objects ranked according to both prestige-based relevance and location proximity. We propose two algorithms that compute LkPT queries. Empirical studies with real-world spatial data demonstrate that LkPT queries are more effective in retrieving web objects than a previous approach that does not consider the effects of nearby objects; and they show that the proposed algorithms are scalable and out Performance baseline approach significantly.

The conventional Internet is acquiring a geo-spatial dimension. [3] Web documents are being geo-tagged, and georeferenced objects such as points of interest are being associated with descriptive text documents. The resulting fusion of geo-location and documents enables a new kind of top-k query that takes into account both location proximity and text relevancy. To our knowledge, only naive techniques exist that is capable of computing a general web information retrieval query while also taking location into account. This paper proposes a new indexing framework for location aware top-k text retrieval. The framework leverages the inverted file for text retrieval and the R-tree for spatial proximity querying. Several indexing approaches are explored within the framework. The framework encompasses algorithms that utilize the proposed indexes for computing the top-k query, thus taking into account both text relevancy and location proximity to prune the search space. Results of empirical studies with an implementation of the framework demonstrate that the paper's proposal offers scalability and is capable of excellent performance.

Users often search spatial databases like [4]yellow page data using keywords to and businesses near their current location. Such searches are increasingly being performed from mobile devices. Typing the entire query is cumbersome and prone to errors, especially from mobile phones. We address this problem by introducing type-ahead search functionality on spatial databases. Like keyword search on spatial data, type-ahead search needs to be location-aware, i.e., with every letter being typed, it needs to return spatial objects whose names (or descriptions) are valid completions of the query string typed so far, and which rank highest in terms of proximity to the user's location and other static scores. Existing solutions for type-ahead search cannot be used directly as they are not location-aware. We show that a straight-forward combination of existing techniques for performing type-ahead search with those for performing proximity search perform poorly. We propose a formal model for query processing cost and develop novel techniques that optimize that cost. Our empirical evaluations on real and synthetic datasets demonstrate the effectiveness of our techniques. To the best of our knowledge, this is the rst work on location-aware type-ahead search. [5] Mapping mashups are emerging Web 2.0 applications in which data objects such as blogs, photos and videos fromdifferent sources are combined and marked in a map using APIsthat are released by online mapping solutions such as Googleand Yahoo Maps. These objects are typically associated witha set of tags capturing the embedded semantic and a set ofcoordinates indicating their geographical locations. Traditionalweb resource searching strategies are not effective in such anenvironment due to the lack of the gazetteer context in the tags.Instead, a better alternative approach is to locate an object by tag matching. However, the number of tags associated with each object is typically small, making it difficult for an object to capture the complete semantics in the query objects. In this paper, we focus on the fundamental application of locating geographical resources and propose an efficient tag centric query processing strategy. In particular, we aim to finds a set of nearest co-located objects which together match the query tags. Given the fact that there could be large number of data objects and tags, we develop an efficient search algorithm that can scale up in terms of the number of objects and tags.Further, to ensure that the results are relevant, we also propose a geographical context sensitive geo-tf-idf ranking mechanism. Our experiments on synthetic data sets demonstrate its scalability while the experiments using the real life data set confirm its practicality.

III. EXISTING SYSTEM

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 approaches proposed by Cong et al. and Li et al. employ a hybrid index that augments nodes in non-leaf nodes of an R/R*-tree with inverted indexes.

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

IV. PROPOSED SYSTEM

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 overcome this critical drawback, we developed much scalable keyword adjacent neighbor expansion (keyword- ANE) 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 near by 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 in depth 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.

KEYWORD-ANE ALGORITHM

Step1. One Query Keyword as Principal Query keyword (The Objects associated with the principal query keyword are principal objects)

Step2. For each Principal Object , the local best keyword cover computed(lbkc)

Step 3. Next, BKC is identified;
Step 4. Return BKC.

V. ARCHITECTURE

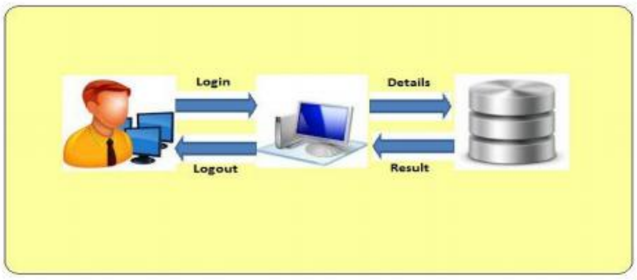


Fig 1: System Architecture

VI. FLOW CHARTS

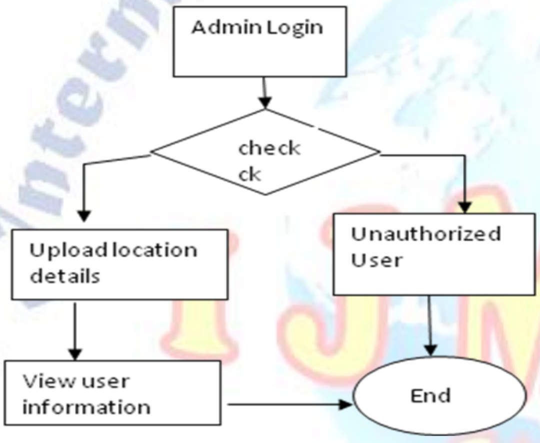


Fig 2: Admin Login

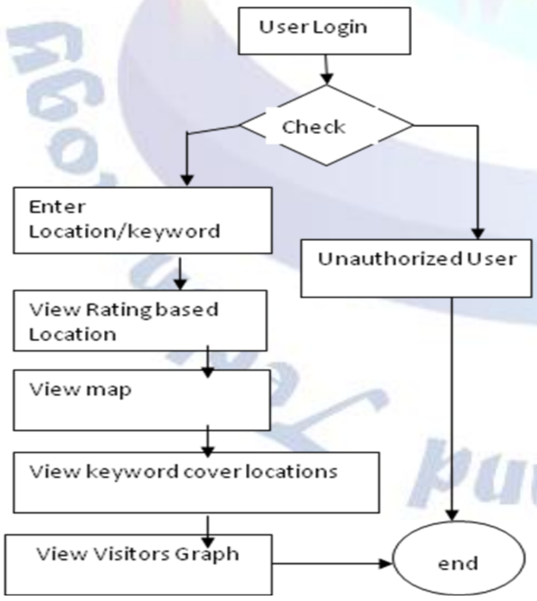
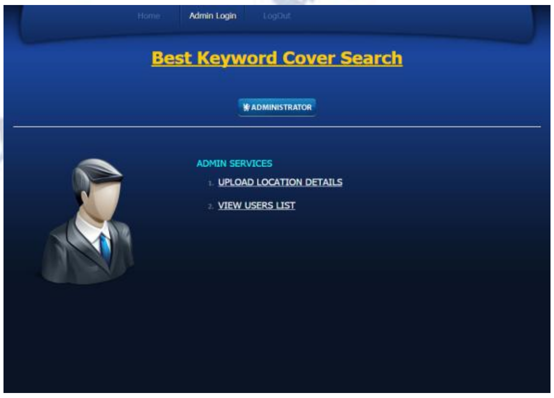


Fig 3: User Login activities

VII. DATABASE TABLES

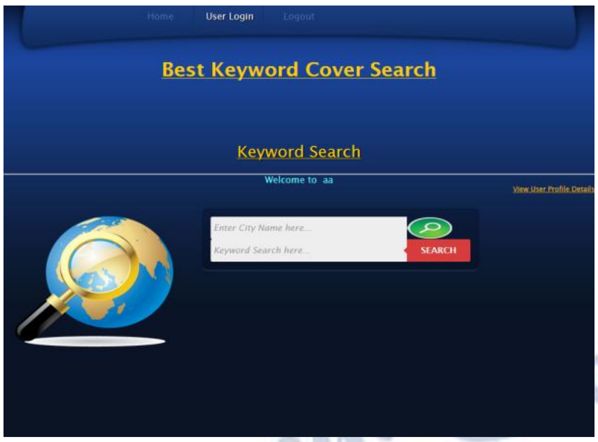
- CREATE TABLE `adminlogin` (`name` varchar(255) NOT NULL,`password` varchar(255) NOT NULL,PRIMARY KEY (`name`))
- CREATE TABLE `admupload` (`keyword` varchar(255) NOT NULL,`location` varchar(255) NOT NULL, `keyname` varchar(255) NOT NULL,`descen` varchar(255) NOT NULL,`keylat` varchar(255) default NULL,`keylon` varchar(255) default NULL,`oth1` varchar(255) default NULL,`othlat1` varchar(255) default NULL,`othlon1` varchar(255) default NULL,`oth2` varchar(255) default NULL,`othlat2` varchar(255) default NULL, `othlon2` varchar(255) default NULL,`oth3` varchar(255) default NULL, `othlat3` varchar(255) default NULL,`othlon3` varchar(255) default NULL,`oth4` varchar(255) default NULL,`othlat4` varchar(255) default NULL,`othlon4` varchar(255) default NULL,`oth5` varchar(255) default NULL,`othlat5` varchar(255) default NULL,`othlon5` varchar(255) default NULL,`rating` varchar(255) default NULL,PRIMARY KEY (`keyname`))
- CREATE TABLE `userratings` (`keyname` varchar(255) NOT NULL,`descc` varchar(255) NOT NULL,`ratings` varchar(255) NOT NULL, `oth1` varchar(255) NOT NULL, `des1` varchar(255) NOT NULL,`rating1` varchar(255) NOT NULL,`oth2` varchar(255) NOT NULL,`des2` varchar(255) NOT NULL,`rating2` varchar(255) NOT NULL,`oth3` varchar(255) NOT NULL,`des3` varchar(255) NOT NULL, `rating3` varchar(255) NOT NULL,`oth4` varchar(255) NOT NULL,`des4` varchar(255) NOT NULL,`rating4` varchar(255) NOT NULL,`oth5` varchar(255) NOT NULL,`des5` varchar(255) NOT NULL,`rating5` varchar(255) NOT NULL, PRIMARY KEY (`keyname`))

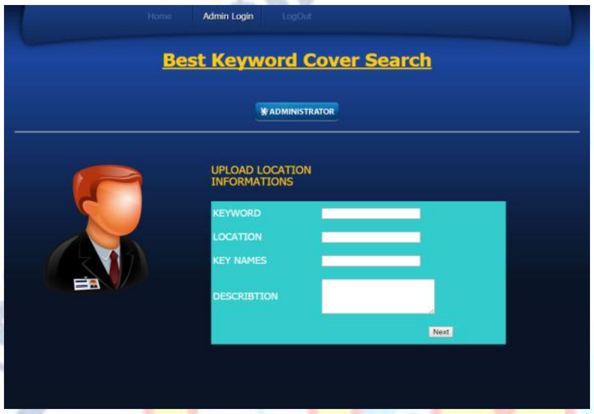
VIII. SCREEN SHOTS

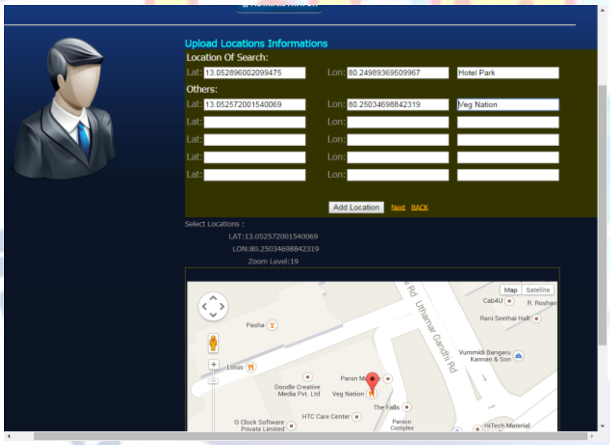


Best Keyword Cover Search

ADMIN SERVICES
1. UPLOAD LOCATION DETAILS
2. VIEW USERS LIST







Location Of Search:

Lat: 13.052896002099475 Lon: 80.24989369509967 Hotel Park

Others:

Lat: 13.052572001540069 Lon: 80.25034698842319 Veg Nation

Lat: Lon:

Lat: Lon:

Lat: Lon:

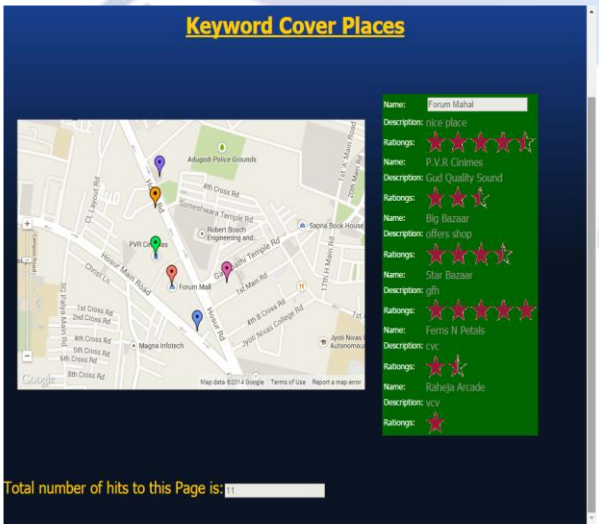
Lat: Lon:

Select Locations :

LAT:13.052572001540069

LON:80.25034698842319

Zoom Level:19



Name: Forum Mahal
Description: nice place
Rationgs: ★★★★½

Name: P.V.R Cinimes
Description: Gud Quality Sound
Rationgs: ★★½

Name: Big Bazaar
Description: offers shop
Rationgs: ★★★½

Name: Star Bazaar
Description: gfh
Rationgs: ★★★★★

Name: Ferns N Petals
Description: cvc
Rationgs: ★½

Name: Raheja Arcade
Description: vcv
Rationgs: ★

Total number of hits to this Page is: 11

IX. RESULTS

Fig 4 shows barchart representing execution time versus keyword cover count of keyword nearest neighbour expansion variant algorithm. When keyword cover count is one , its execution time is 82 milliseconds. When keyword cover count is two, its execution time is 121 milliseconds. When keyword cover count is three, its execution time is 127 milliseconds.

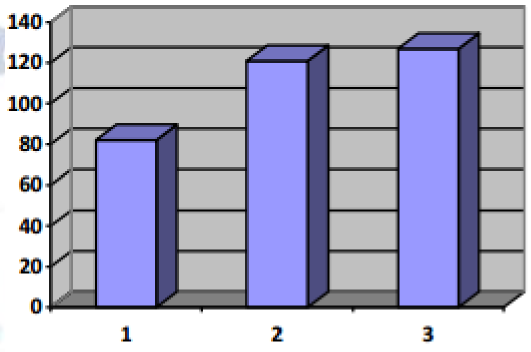


Fig. 4 Execution time versus keyword covers count of KNNE Variant algorithm

Fig 5 shows barchart indicating memory consumption required while executing multiple query keyword using keyword nearest neighbour expansion variant algorithm. When keyword cover count is one , memory consumption required is 38417360 KB. When keyword cover count is two, memory consumption required is 41220920 KB. When keyword cover count is three, memory consumption required is 37853264 KB. As keyword cover count increases, memory consumption time varies

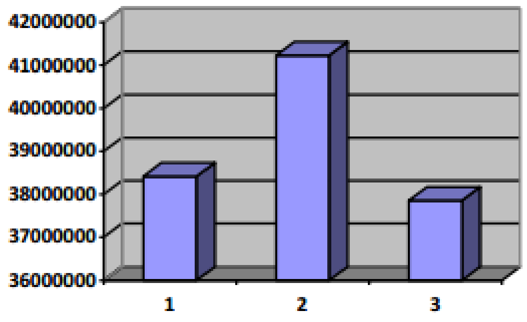


Fig. 5 Memory consumption versus keyword cover count of Keyword-ANE Algorithm

From the experiments conducted, it has been observed that execution time required by keyword adjacent neighbor expansion algorithm is much faster compared to baseline algorithm

X. 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-ANE algorithm than in the baseline algorithm

REFERENCES

[1] Zhisheng LI Singapore Management University Ken C. K. LEE University of Massachusetts Dartmouth RTree: An efficient index for geographic document search

[2] Xin Cao ,Gao Cong Christian S. Jensen, Retrieving top-k prestige-based relevant spatial web objects.

[3] Gao Cong Christian S. Jensen Dingming Wu Efficient retrieval of the top-k most relevant web objects

[4] Senjuti Basu Roy,Kaushik Chakrabarti, Location-aware type ahead search on spatial databases: emetics and efficiency.

International Journal for Modern Trends in Science and Technology IJMTST