Asian Journal of Information Technology 13 (11): 678-683, 2014
ISSN: 1682-3915

Efficient Cluster Based Processing of Joint Top-K Spatial Keyword Queries

A. Sasi Kumar and P. Anandhakumar
Department of Computer Technology, Madras Institute of Technology,
Anna University, Chennai, Tamil Nadu, India

Abstract: Due to the large number of spatial relations in databases, much research focuses on developing algorithms to efficiently extract the spatial query processing. In this study, the problem is defined for top-k spatial keyword search which retrieves the top-k spatial objects that are most relevant to query in terms of joint spatial and textual relevance. The proposed work aims to provide the solution by introducing a complete geospatial knowledge discovery framework for the detection of spatial patterns with pruning. These spatial patterns will not contain any query keywords. A novel approach and indexing structure is proposed for the joint processing of top-k spatial keyword queries using cluster based association rule mining. If the load is heavy, then the joint processing keyword query is used to improve the robustness of the proposed work.

Key words: Clustering, association rules, pruning, top-k query, spatial databases, textual databases

INTRODUCTION

Spatial keyword search is a widely investigated topic along with the development of geo-positioning techniques (Hariharan *et al*., 2007). Here, the problem of top-k spatial keyword search (De Felipe *et al*., 2008) which retrieves the top-k objects that are most relevant to the query in terms of joint spatial and textual relevance has been studied. The IR tree (Li *et al*., 2011) is derived from the basic structure R tree (Guttman, 1984). Existing methods index data objects in the IR tree which supports textual and spatial pruning simultaneously and processes the query by traversing tree nodes along with associated inverted files. However, these searching methods suffer from a vast number of times of accessing inverted files which results in slow query time and increased I/O cost.

Different emerging applications warrant efficient support for top-k queries. Most of these applications compute queries that involve joining and aggregating multiple inputs to provide users with the top-k results. Spatial keyword queries are important in geographic search technology (Chen *et al*., 2006) services such as those offered by travel sites, web portal, etc.

LITERATURE REVIEW

Processing of joint top-k spatial keyword queries: The processing of joint top-k spatial keywords are discussed with various algorithms. This research studied the existing algorithms to process the joint top-k spatial keywords.

The ITERATE algorithm: The ITERATE algorithm for computing the joint top-k spatial keyword query has been adapted from an existing algorithm (Wu *et al*., 2012) that considers a single query.

The joint top-k spatial keyword query Q consists of a set of subqueries q_i. The ITERATE algorithm computes the top-k results for each subquery separately. The arguments consist of joint query Q, the root of an index root and the number of results k for each subquery.

Algorithm (ITERATE (joint query Q, tree root, integer k):

Input: Spatial joint keyword, Tree
Output: Joint top-k spatial keyword
Step 1: A joint top-k spatial keyword query Q consists of a set of subqueries q_i.
Step 2: It computes the top-k results for each subquery separately.
Step 3: The arguments are a joint query Q, the starting node of an index is denoted as root and k is the notation used to represent the number of results k for each subquery q_i.
Step 4: When processing a subquery $q_i \in Q$, a priority queue Q on the nodes to be visited is maintained.
Step 5: The key of an element $e \in U$ is the minimum distance mindist $(q_i.\lambda, e.\lambda)$ between the query q_i and the element e. Where, λ represents the location for the subquery and key element.
Step 6: The keyword information is utilized from step 5 to prune the search space. It only loads the posting lists of the words in q_i.
Step 7: Repeat the step 5 and 6 until it reaches non-leaf entry.
Step 8: If the non-leaf entry is pruned if it does not match all the keywords of q_i then
Step 9: Return k elements that have the smallest Euclidean distance to the query and contain the query keywords.

The W-IR tree: The W-IR tree (Wu *et al*., 2012) organizes data objects according to both location and keywords.

Corresponding Author: A. Sasi Kumar, Department of Computer Technology, Madras Institute of Technology, Anna University, Chennai, Tamil Nadu, India

Word partitioning: The first step in presenting the index structure is the partitioning of a data set according to the keywords. The hypothesis is that a keyword query will often contain a frequent word (say w). This inspires to partition the data set D into the subset D^+ whose objects contain w and the subset D^- whose objects do not contain w. Therefore, then processing a query containing the subsets and not be examined. It aims at partitioning D into multiple groups of objects such that the groups share as few keywords as possible. However, this problem is equivalent to, e.g., the clustering problem and is NP-hard.

Algorithm (WordPartition (Data set D, sorted list of words W, integer B, list of tree nodes L):

Input: Data set, Set of sorted words, Tree
Output: Partitioned data set D from multiple groups of objects
Step 1: Let a keyword query will often contain a frequent word (say w).
Step 2: Let a partition data set D into the subset D^+ whose objects contain w and the subset D whose objects do not contain w.
Step 3: When processing a query containing w, partitioning D into multiple groups of objects such that the groups share a few keywords.
Step 4: Let the list W of keywords of objects sorted in descending order of their frequencies be w1, w2, ..., wm where m is the number of words in data set D.
Step 5: Initially, frequent words are handled before infrequent words.
Step 6: Partitioning the objects into two groups namely word w1 and word w2.
Word w1: The group whose objects contain w1 and the group whose objects do not.
Word w2: Then partition each of these two groups by word w2.
Step 7: The data set can be partitioned into at most 2, 4, ..., 2m groups. By construction, the word overlap among groups is small which will tend to reduce the number of groups accessed when processing a query.
Step 8: Word partitioning algorithm recursively applies the above partitioning to construct a list of tree nodes L.
Step 9: To avoid underflow and overflow, each node in L must contain between B/2 and B objects where B is the node capacity.
Step 10: When the number of objects in D (i.e., |D|) is between B/2 and B, D is added as a node to the result list L.
Step 11: If |D|<B/2 then D is returned.
Step 12: else If |D|>B then partition the data set D.
Step 13: If W is empty then, all the objects in D must have the same set of words and cannot be partitioned further by words.
Step 14: Use the main-memory R-tree with node capacity B to partition D based on location and adds the leaf nodes of the R-tree to the result list L.
Step 15: When W is nonempty, derive the most frequent word in W.
Step 16: The objects in D are partitioned into groups D^+ and D^- based on whether or not they contain w.
Step 17: Repeat the step 16 until all the words are partitioned from the data set.
Step 18: The remaining objects from those recursive calls (i.e., the sets T^+ and T^-) and then combined into the set T'.
Step 19: If T' has enough objects, it is added as a node to L.
Step 20: Otherwise, If the initial call of the algorithm returns a group with less than B/2 objects, it is added as a node to the result list L.
Step 21: Return the combined set T'.

W-IBR tree and variants: The W-IBR tree exploits keyword partitioning and inverted bitmaps for indexing spatial keyword data. The W-IBR tree requires less space than the W-IR tree for both data sets. Each node in the W-IR tree contains a pointer to its corresponding inverted file. By replacing each such inverted file by an inverted bitmap, the storage of the W-IR tree can be reduced and also save I/O cost during query processing. The resulting tree is called the W-IBR tree.

The GROUP algorithm: The GROUP algorithm (Wu *et al*., 2012) aims to process all subqueries of Q concurrently by employing a shared priority queue U to organize the visits to the tree nodes that can contribute to closer. Unlike ITERATE, GROUP guarantees that each node in the tree is accessed at most once during query processing.

Pruning strategies: The algorithm uses three pruning rules. Let e be an entry in a nonleaf tree node. We utilize the MBR and keyword set of e to decide whether its sub-tree may contain only objects that are relevant or irrelevant to all subqueries of Q:

- Prunes a non-leaf entry whose sub-tree contains objects that have too few relevant keywords for subqueries of Q. This rule is effective when the union keyword set Q. ψ of Q is small, e.g. when many keywords are shared among subqueries of Q
- Prunes a non-leaf entry whose sub-tree contains objects that are located too far away from subqueries of Q. This rule is effective when the MBR Q is small, i.e. when the subqueries of Q are located within a small region

Algorithm (group (joint query Q, tree root root, integer k)):

Input: Joint query Q, root of an index and keywords k in each subquery
Output: Top-k results of each subquery
Step 1: The priority queue is defined as U which is used to maintain the top-k objects of each subquery $q_i \in Q$.
Step 2: For each and every iteration, the top entry e (with the smallest key) is dequeued from the priority queue. e is defined as a minimum distance of subquery q_i.
Step 3: This iteration loop executes, i.e., whether the e entry is dequeued for subquery and also it checks all the child nodes 'e' is processed or not.
Step 4: In case, if e.key value has increased, then insert that e value into the priority queue U to arrange the result in ascending order.
Step 5: If entry 'e' survives, read the child node CN of e.
Step 6: To retrieve the relevant keywords for entries in CN, the posting lists of CN are read only for the computed query value named as Q^*.
Step 7: If CN is a non-leaf node each entry e' in CN must be enqueued into the priority queue U.
Step 8: If CN is a leaf node, discard the non-qualifying objects in CN. The remaining objects update the result for relevant subqueries of Q^*.
Step 9: Finally, return the top-k results of each sub-query.

ASSOCIATION RULE MINING

Association rule mining, one of the most important and well researched techniques of data mining was first introduced by Agrawal *et al.* (1993). It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories.

CLUSTERING

Classification can be taken as a supervised learning process; clustering is another mining technique which is similar to classification. However, clustering is a unsupervised learning process. Clustering is a process of grouping a set of physical or abstract objects into classes of similar objects, so that the objects within the same cluster must be similar to some extent and they should be dissimilar to those objects in other clusters. In classification which record belongs to which class is predefined while in clustering there are no predefined classes. In clustering, objects are grouped together based on their similarities. Similarity, between objects is defined by similarity functions. Usually similarities are quantitatively specified as distance or other measures by corresponding domain experts (Han and Kamber, 2000). A comprehensive survey of the current clustering techniques and algorithms has been done by Berkhin (2002).

PROPOSED CLUSTER BASED PROCESSING OF JOINT TOP-K SPATIAL KEYWORD QUERIES

System overview: In this thesis, an efficient cluster based processing of joint top-k spatial keyword queries has been proposed to adapt the solution to existing index structures for spatial keyword data. The system architecture for the proposed efficient cluster based processing of joint top-k spatial query processing is shown in Fig. 1. This prunes considerable amount of data, reduces the time needed to perform data scans and requires less contrast but also ensures the correctness of the mined result. Combinations of two algorithms and a range of indexes demonstrate that the GROUP algorithm on the W-IBR tree is the most efficient combination for processing joint top-k spatial keyword queries. It is straightforward to extend the solution to process top-k spatial keyword queries in spatial networks. It is effective at pruning early irrelevant nodes that do not contain query keywords. It reduces the response times for spatial indexes, scope indexes and their combination with textual indexes. It reduces the storage cost for spatial indexes and scope indexes.

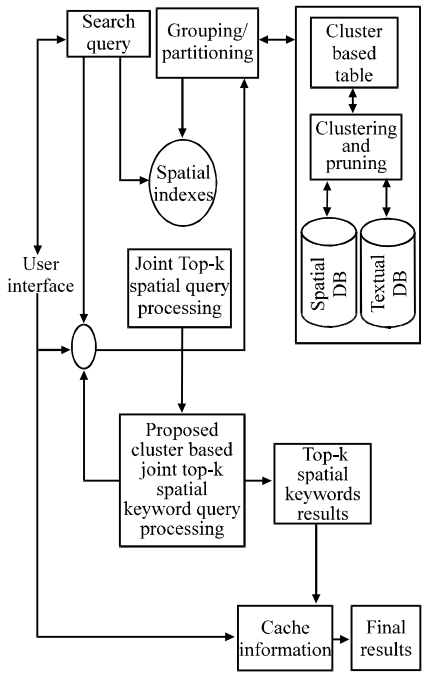

Fig. 1: System architecture for the proposed efficient cluster based processing of joint top-k spatial query processing

Caching information: Hu *et al.* (2005) cache the top-k result from the spatial query processing database if the same search keyword is given, the result is retrieved from the cache. So, automatically the response time is reduced.

Enhanced query processing with Top-K Spatial Keyword Join (TKSKJ): Association rule mining is used to find out association rules that satisfy the predefined minimum support and confidence from a given database which was first introduced by Agrawal *et al.* (1993). It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories.

A standard association rule is a rule of the form X→Y which says that if X is true of an instance in a database, so is Y true of the same instance with a certain level of significance as measured by two indicators, support and confidence. To mining the data for producing best results using association rule the following steps are applied:

- Identify associative variables
- Select and transform attribute information; derive non-spatial predicates
- Identify and quantify spatial components involved; derive spatial predicates
- Mine spatial association rules
- Visualize and evaluate intermediate mined results

Cluster management is the discipline of planning, organizing and managing resources to bring about the successful completion of specific objectives. Pruning techniques in particular such to support-based pruning have been exploited to reduce the complexity of the extraction task. The Minimum Bounding Rectangle (MBR) and keyword set, decides whether its subtree may contain only objects that are relevant or irrelevant to all sub-queries.

Algorithm (Cluster_Table_Generate (D, Minsup)):

Input: Database D, Minum Support
Output: Answer (Answer = UB_k for $l \leq k \leq M$)
Step 1: Assign 'k' as the length of the transaction record
Step 2: Assign M as the length of the longest transaction record in database D
Step 3: For each transaction record k iterate the loop from $1 \leq k \leq M$.
Step 4: If the big itemset is reached then
Step 5: Assign the B_l (First big itemset)
Step 6: Repeat the steps 3-5 until it reaches the biggest itemset B_k.
Step 7: Each L_k value is inserted as entry in the cluster table.
Step 8: After creation of cluster table, generate the set of candidate k-itemsets C_k.
Step 9: The candidate itemset C_k is passed as argument to find the big itemset B_k.
Step 10: Step 8 and 9 are repeated until it reaches the L_k value
Step 11: Return the biggest k object from the cluster table

Algorithm (Proposed Top-K Shortest Path Join (TKSKJ):

Input: Query Q, Data set D, Cluster table CT, WordPartitioning W_k, List of tree nodes L
Output: Joint top-k spatial keyword
Step 1: Search query is given by the user.
Step 2: Spatial database and textual database are maintained.
Step 3: Clustering and pruning technique are applied to these data sets.
Step 4: The search result is stored in the cluster table.
Step 5: Group algorithm is used to store the query results as clusters in spatial index table. (Refer Group Algorithm).
Step 6: To divide the data set D into partition word (Refer WordPartition Algorithm)
Step 7: By using the step 5 and 6, find the joint top-k query processing and the result is stored in the spatial index.
Step 8: For this spatial index, apply the technique (Refer Cluster_Table_Generate Algorithm) with association rule.
Step 9: Finally, the top-k joint spatial query is returned as a result.

EXPERIMENTS AND RESULTS

Performance of Top-K Spatial Keyword Join (TKSKJ): To evaluate the efficiency of the proposed method, TKSKJ algorithm is implemented. When there is an increase in the number and the size of pattern discovered, the performance gap of the proposed algorithms becomes more evident. In this experiment, the efficiency of the TKSKJ algorithm is compared with various numbers of attributes where minsupport is 40% and shown in Fig. 2.

The comparative study based on performance between W-IR tree and the proposed TKSKJ has been analyzed. The results in Fig. 3 also demonstrate that the

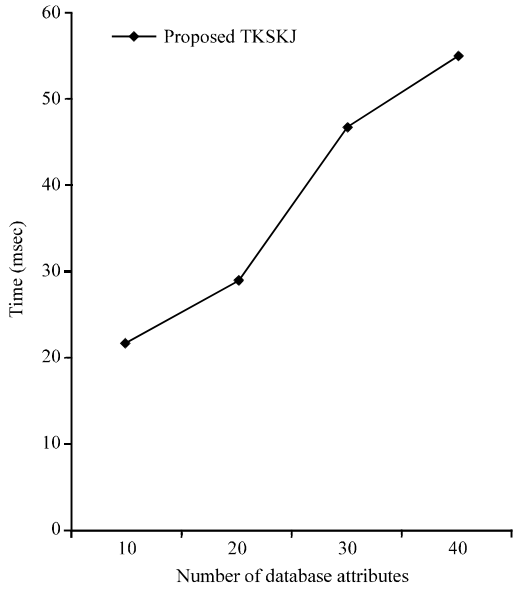

Fig. 2: Performance of the proposed system (TKSKJ)

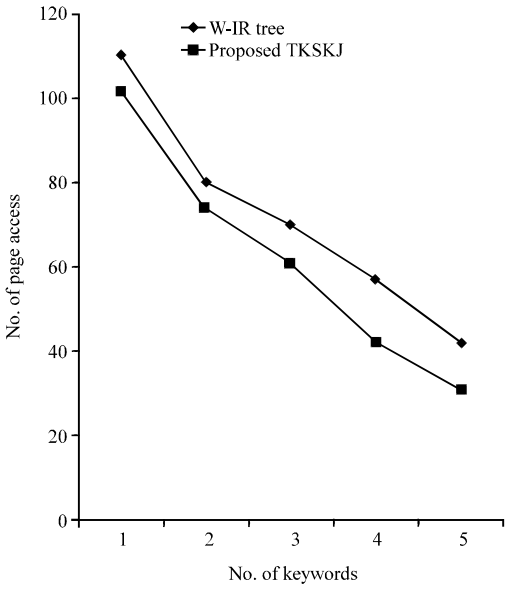

Fig. 3: Performance analysis of W-IR Tree and Top-k Spatial Keyword Join (TKSKJ)

performance analysis of W-IR Tree and TKSKJ. The runtime analysis of the proposed system is shown in Fig. 4.

Figure 5 shows how the GROUP algorithm processes queries jointly; it exploits opportunities to share disk accesses among subqueries. Such shared disk pages are only visited once. Therefore, the cost of the proposed TKSKJ is lower than that of the ITERATE.

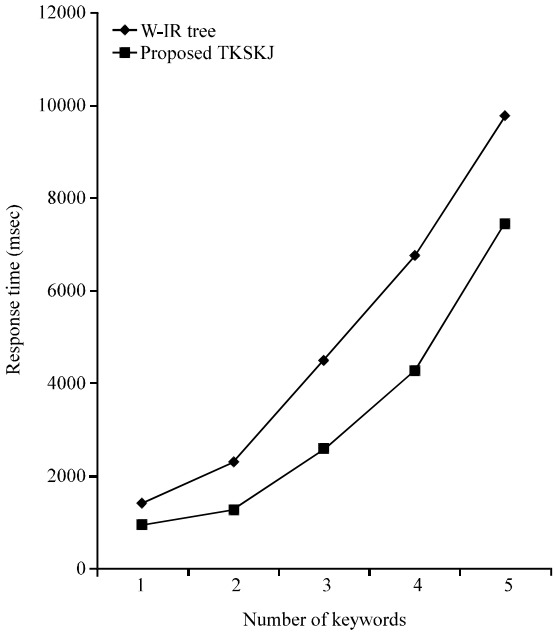

Fig. 4: Run time analysis of the proposed system

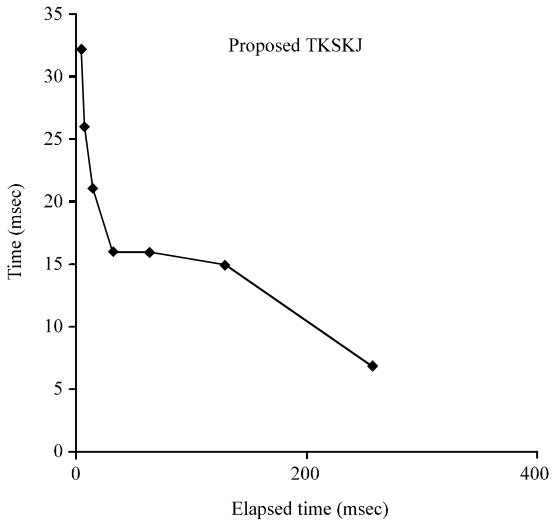

Fig. 5: Elapsed time analysis for the proposed system

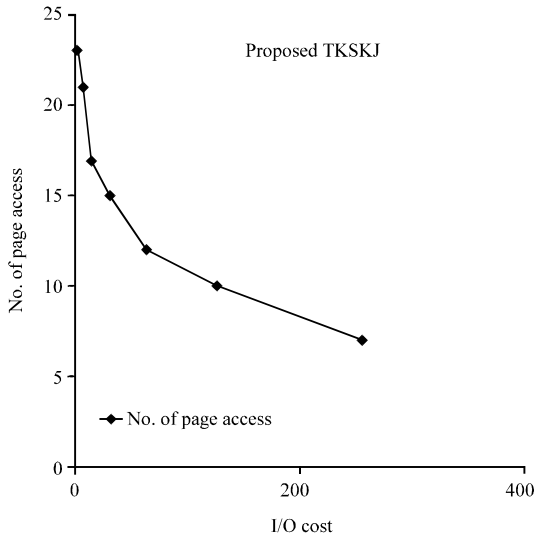

Fig. 6: I/O Cost Model of the proposed system

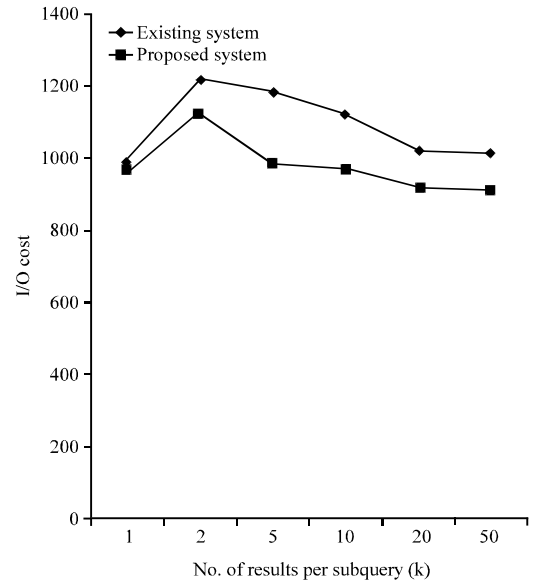

Fig. 7: Comparative results of I/O Cost Model between proposed and existing system

It finds that the performance of the inverted-R trees and the ITERATE based solutions are insensitive to the size of the joint query's Minimum Bounding Rectangle (MBR). This is because they process subqueries one by one so that the performance is not affected by the locations of the subqueries. The I/O costs of the GROUP-based methods increases. This is because the GROUP algorithm shares the computation among subqueries, so that a smaller size of the joint query's MBR (subqueries that are close to each other) favours it.

In this experiment, GROUP and ITERATE algorithms are applied on the W-IBR tree and the W-IR tree, respectively to evaluate the effectiveness of the inverted bitmap optimization. I/O costs of the methods vary with varying k (the number of results per subquery). The W-IBR tree accesses compact inverted bitmaps to retrieve keywords of entries/objects that it incurs a lower cost than does the W-IR tree (using inverted files). Based on the I/O cost, the proposed method TKSKJ reduces the number of page access has been depicted in the Fig. 6.

Comparing the I/O cost of the existing with the proposed system, it has been observed that the latter reduces the storage space and costs. The I/O cost of the proposed TKSKJ Method comparatively less than the

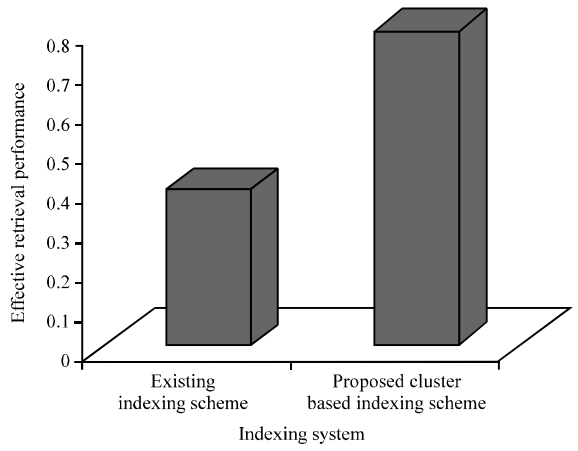

Fig. 8: Comparative performance analysis between existing and proposed indexing structures based on support value

existing methods as depicted in Fig. 7. In the proposed cluster based indexing system, the support value was empirically estimated and lies between 0.8 and 0.6, depending on the query attributes.

Figure 8 shows the comparison of the performance of the proposed and existing indexing structure. To summarize, the proposed system significantly outperforms the existing methods shows that cluster enhancement improves the performance and reduces the I/O cost.

CONCLUSION

Joint top-k spatial keyword query presents an efficient means of computing the query. Efficient mining approach presents a good computational cost in all datasets, obtaining a good scalability and the best performance in accuracy. Introduced cluster based association rule quality measures into the spatial domain; it only requires a single scan of the database and delivering a complete geo-spatial knowledge discovery framework. W-IBR-tree exploits keyword partitioning and inverted bitmaps for indexing spatial keyword data and the GROUP algorithm that processes multiple queries jointly. By the better prune powers with combinations of two algorithms and a range of indexes demonstrate that the GROUP algorithm on the W-IBR-tree is the most efficient combination for processing joint top-k spatial keyword queries.

First it should verify the meaning of that word (ontology verification) then finally gives the search result. Assume the pre-existence of an ontology associated with the database. Therefore, develop a semantic model and basic criteria like correctness and completeness. By uploading the application in cloud server more user can interacts with the application. Thus, each and every user can get equal benefits in the process of file sharing. These things bring some advantages if this map application might enhance in future means. Finally, this research creates a report for the search results in the graphical way. It also generates the report for the frequently used keywords.

REFERENCES

Agrawal, R., T. Imielinski and A. Swami, 1993. Mining association rules between sets of items in large databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, May 25-28, 1993, Washington, DC., USA., pp: 207-216.

Berkhin, P., 2002. Survey of clustering data mining techniques. Technical Report, Accrue Software, Inc. http://www.ee.ucr.edu/~barth/EE242/clustering_survey.pdf.

Chen, Y.Y., T. Suel and A. Markowetz, 2006. Efficient query processing in geographic web search engines. Proceedings of the ACM SIGMOD International Conference on Management of Data, June 27-29, 2006, ACM, New York, USA., pp: 277-288.

De Felipe, I., V. Hristidis and N. Rishe, 2008. Keyword search on spatial databases. Proceedings of the IEEE 24th International Conference on Data Engineering, April 7-12, 2008, IEEE Computer Society, Washington, DC, USA., pp: 656-665.

Guttman, A., 1984. R-Tree: A dynamic index structure for spatial searching. Proceedings of the of 13th ACM SIGMOD International Conference on Management of Data, June 18-21, 1984, Boston, USA., pp: 47-57.

Han, J. and M. Kamber, 2000. Data Mining: Concepts and Techniques. Morgan-Kaufman Publishers, New York.

Hariharan, R., B. Hore, C. Li and S. Mehrotra, 2007. Processing Spatial-Keyword (SK) queries in Geographic Information Retrieval (GIR) systems. Proceedings of the IEEE 19th International Conference on Scientific and Statistical Database Management, Banff, Alta, July 9-11, 2007, IEEE Computer Society Washington, DC, USA., pp: 16.

Hu, H., J. Xu, W.S. Wong, B. Zheng, D.L. Lee and W.C. Lee, 2005. Proactive caching for spatial queries in mobile environments. Proceedings of the 21st International Conference on Data Engineering, April 5-8, 2005, Tokyo, Japan, pp: 403-414.

Li, Z., K.C.K. Lee, B. Zheng, W.C. Lee, D. Lee and X. Wang, 2011. IR tree: An efficient index for geographic document search. IEEE Trans. Knowledge Data Eng., 23: 585-599.

Wu, D., M.L. Yiu, G. Cong and C.S. Jensen, 2012. Joint Top-K spatial keyword query processing. IEEE Trans. Knowledge Data Eng., 24: 1889-1903.