R- Tree based Filtering Algorithms for Location-Aware based System

Priya H Jagtap
Computer Engineering Department
Affiliated to Kalyani Charitable Trust
LGNSCOE Nasik, India

N. K. Zalte
Computer Engineering Department
Affiliated to Kalyani Charitable Trust
LGNSCOE Nasik, India

ABSTRACT
A Location Based Services (LBS) have attracted and valuable attention from industrial and academic communities. Current LBS system used a pull model also called as user-initiated model, from where user can issues a query to system and which replies location-aware answer. A push model also known as server-initiated model is becoming an unavoidable computing model in LSB to provide user with instant replies and subscribers to capture their interest. Multiple research problems arises when designing this system. Efficient filtering algorithms and pruning techniques are used to accomplish high performance and to render user with instant reply. Further proposed another algorithm called FlexRPset, which gives one extra parameter to make trade-off between result size and efficiency. FlexRPset generate fewer representative patterns than RPLocal and MinRPset is design to improve scalability.

1. INTRODUCTION
These days many systems are widely accepting Location-based services. A LBS services provided result by convincing location awareness like Google map and Foursquare. For example, if mobile or tab user wants to find electronic shops nearby , they can use LBS system by giving query “electronic shops”. Considering user location and keywords, LBS system displays result. Current LBS system used a pull model or user- initiated model [1],[2] from where a user can fire query to system which responds by location awareness. For Location-aware system, a system gives result to achieve high-performance. This is attain by using push model or server-initiated model. Subscribers enroll spatial-textual in push or server-initiated model that consist region based information[3] and return pertinent result.

The challenge we have to resolve in a publish/subscribe system is to gain high performance. A publish/subscribe system should handle hundreds of millions of subscribers & deliver messages to proper subscribers in milliseconds(ms). As messages & subscriptions comprise both location information & textual description, it is significantly too costly to deliver messages to proper subscribers. These calls for an efficient filtering technique to define location-aware publish/subscribe services.

To tackle these problems, use Rt-tree which is a token-based R-tree index structure that merge each R-tree node with a set of tokens choosen from subscriptions. Rt-tree is a(BST) balanced search tree. All leaf comparison between a and A data entries, where each entry goes to subscription. Each internal node has between a & A node entries. Each entry is a triple <Child, MBR, TokenSet> , Child is a pointer to its child node, MBR is nothing but minimum bounding rectangle of all entries within this child, and TokenSet is a set of tokens choosen from subscriptions. A leaf node’s token set is the sum of tokens of all subscriptions between this node & an internal node’s token set is the sum of token sets of all entries within this node. As an entry matches to a node, for directness a node is cited exchangeable with its matching entry if the circumstance is clear. Using the Rt-tree, we build a filter-and-certification framework which will deliver a message. To lessen the number of tokens with Rt-tree nodes, select few superiority representative tokens from subscriptions and join them with Rt-tree nodes. Where the Rt+ select single representative pattern and has low pruning power. This technique not only cut down index sizes but also emblish the performance, go through following steps:

1) Introduce a new computing model to validate the Location-aware problem.
2) Use novel index structure, the Rt++-tree, by incorporating superiority representative tokens chosen from subscriptions into the Rt++-tree nodes. Our techniques allows conjunctive queries & ranking queries.
3) Using the proposed indexes, an effective pruning techniques and effective filtering algorithm for better performance is develpoed.
4) FlexRP [5] Set algorithm for to filtration for better relevancy of result in less time period.

2. RELATED WORK
2.1 Spatial Keyword Search Method
Existing methods emphasis on finding relevant points-of-interest based on users locations and query keywords [6]. Deep study is present on spatial keyword search [7, 8, 9]. One problem is knn based keyword search, which, given a query comprising of a location and a set of keywords, discovers top-k relevant POIs by considering distance and textual relevance. Felipe et al. [10] combined signature files into R-tree, and Cong et al. [11] combined inverted files and R-tree. Another issue is region-based keyword search, which, given a query consisting of a region and a set of key-words, finds the applicable POIs relevant to the keywords in the region. The methods addressing the problem also emphasis incorporated inverted lists of keywords into R-tree nodes [7, 8, 9]. Another problem is spatio-textual similarity search problem that considerable different from the above-listed problems. The underlying data is a set of spatio-textual objects that contain regions and tokens. Authors highlights on spatio-textual similarity between objects & queries & design efficient filtering algorithms for similarity search.

2.2 Keyword Search Method
Firstly it uses existing content based publish/subscribe methods to generate the candidates that meet the textual constraint [6]. This method then verifies candidates used to

check whether they satisfy spatial constraint. This method generates huge numbers of candidates and causes poor performance.

2.3 Spatial Search Method

This method first generates the location based candidates that satisfy the spatial constraint, applying current methods, e.g., segment tree or R-tree [11]. Then it filters candidates which do not meet textual constraint. This technique also generates large numbers of candidates which causes poor performance.

2.4 Spatial-Keyword Search based method

We can lead current spatial keyword search methods, e.g., IRTree [1], to support this model. Another method, IRTree construct, where IRTree visit from the root node to leaves. For each node, it uses the minimum bounding rectangle to see the spatial constraint & employ the inverted index to examine the textual constraint. However the IRTree has to impose many nodes unnessarily & extends to low efficiency since if a node contains a token and satisfies the spatial constraint, it should access this node. To cover this problem, it suggests new indexes and filtering algorithms. In this paper, authors concentrate on the in-memory setting to attain instant performance.

2.5 Frequent Pattern Search Method

An MinRPset algorithm find a minimal representative pattern set with error guarantee. MinRPset produces the smallest solution that we can possibly employ under the given problem setting, when the number of frequent closed patterns is below one million it takes a reasonable amount of time to finish. But MinRPset have some issues. They are very space-consuming and time-consuming on some compress datasets when the number of common closed patterns is large. To solve this problem, we propose another algorithm called FlexRPset, which renders one spare parameter K to allow users to make a trade-off between result size and efficiency[5].

3. PROPOSED SYSTEM

In the standard R-tree, it doesn't has textual pruning power, use a token-based R-tree, called Rt -tree, by incorporating tokens of subscriptions into R-tree nodes. Rt -tree is a balanced search tree. However an Rt -tree node may have huge numbers of leaf descendants and it is expensive to check whether a node is a pivotal node. Based on this study, a filter-and-verification framework is enforce.

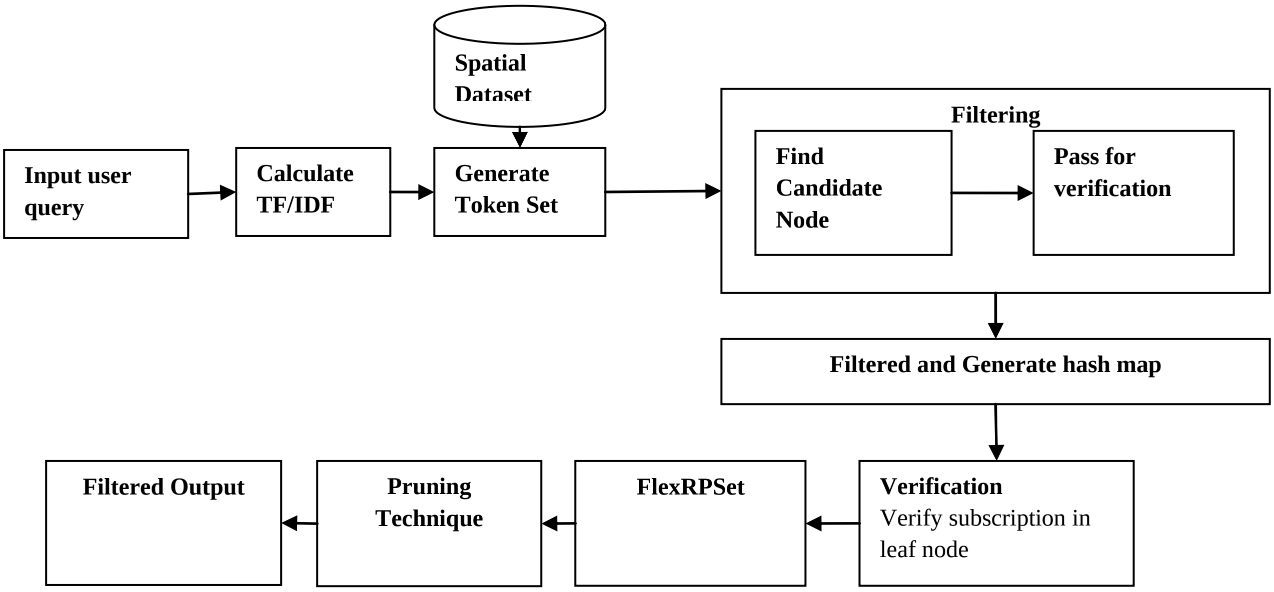

Fig 1: System Architecture

In filtering, it discovers a set of candidate nodes which is a superset of pivotal nodes. In the verification step, firstly assert the subscriptions in the leaf candidate nodes generated in the filter step. Efficient pruning techniques is used to achieve high performance.

In the standard R-tree, it doesn't has textual pruning power, so here it use a token-based R-tree which is called as Rt -tree, by integrating tokens of subscriptions into R-tree nodes. Rt -tree is a balanced search tree. However an Rt -tree node may have large numbers of leaf descendants and it is expensive to check whether a node is a pivotal node. Based on this, a filter-&-verification framework is enforced. Where it find a set of candidate nodes which is a superset of pivotal nodes. In verification step, verify the subscriptions in the leaf candidate nodes generated in the filter step. For this, two filters are introduce. Are as below.

Given a node n, let n:R denote its MBR and n:T denote its token set which can be obtained from the corresponding entry in its parent node. We prune node n, if it satisfies

MBR Filter: It invalidates spatial constraint as subscriptions under node n have no overlap with m; or

Token Filter: It invalidates the textual constraint. The reason is that any subscription under node n must contain a token in n:T which does not look in m:T, thus the subscription does not meet the textual constraint.

The nodes that are not pruned by the MBR filter and token filter are called candidate nodes. The subscriptions in the leaf candidate nodes are candidate answers of message m. Based on the two filters, a filter-and-verification algorithm is implement.

3.1 Algorithm

Input :

1) Subscription set
2) Messages

Output :

Filtered Output for given messages

Begin :

1) Build Rt-Tree Algorithm
2) Call Filtration Function
3) Call Verification Fuction
4) Call FlexRP Set Algorithm
5) Call Pruning Algorithm

End

Given a message m, we traverse the R t -tree in pre-order. From the root node, we scan each of its entries, e.g., node n. If node n satisfies one of the two filters, we prune node n; otherwise we visit n's children and repeat the above steps. repetatively, we can get all leaf candidate nodes. For each subscription s on leaf candidate nodes, we check whether nodes filtered by MBR filter or Token filter. If yes, s is an answer.

3.2 Mathematical Model

Let the system be described by S,

S= { D, GTS, F, V, PT, F }

Where,

S: is a System.
D: is the set of Spatial Dataset.
GTS: Generate Token Set.
F: Filtering
V: Verification
PT: Pruning Technique
F: Flex RP set

Table 1. Functional Dependency

	Fn1	Fn2	Fn3	Fn4	Fn5
Fn1	1	0	0	0	0
Fn2	0	1	0	0	0
Fn3	0	0	1	0	0
Fn4	0	0	0	1	0
Fn5	0	0	0	0	1

Here Fn1: Generates Token Set, Fn2: Filtered, Fn3: Verified, Fn4: Pruned, Fn5: Flex RP set

4. RESULT DISCUSSION

The results depicts that proposed method gives better results compared with existing approaches. An observation is that our algorithm always achieve better performance for any type of massage having wide message length. Fig. 2 show the results of proposed system. It shows that proposed system reduces an execution time by minimum 0.3 second for any type of message length.

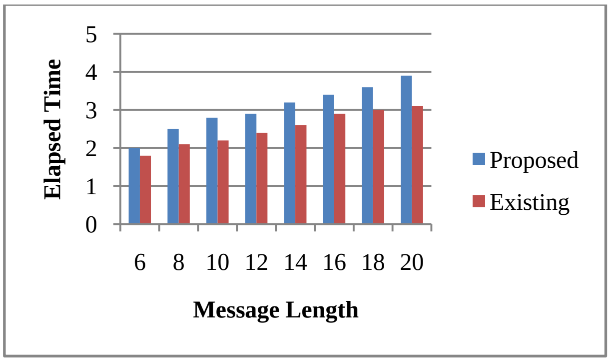

Fig 2: Result of proposed system compare with Existing studies

Fig .3 Shows time result set for minimum support using existing algorithm and proposed algorithm. From where it is clear that our proposed system required less time than existing system. Here performance of newly implemented method Rt-tree is compared with existing IR Tree. Rt-tree required less time than Rt-Tree for given minimum support.

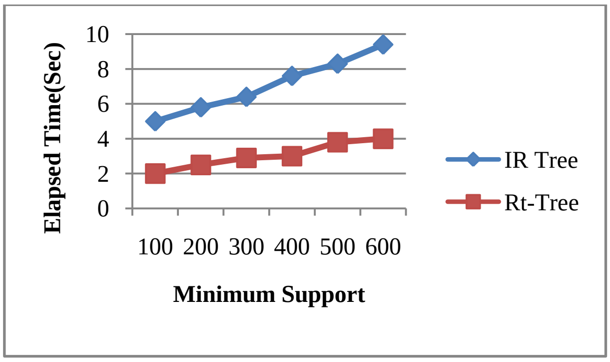

Fig 3: Time Result Set

Fig. 4 show the performance of FlexRPset when *minimum support is* varied. FlexRPset achieve better number of representative patterns when *minimum support* increases, but their running time also increases.

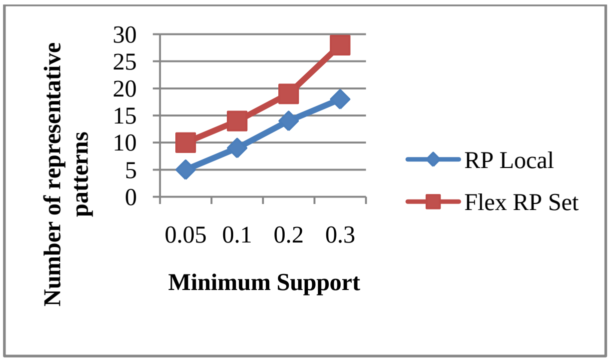

Fig 4: FlexRP Result Set

5. CONCLUSION

Studying the location-aware publish/subscribe problem in literature survey. Further proposed an effective index structure Rt-tree by integrating textual description into R-tree nodes. Then employ reducing the number of tokens in each node. Also enforce efficient algorithm to directly find answer without verification steps by using a filter-and-verification framework. Used algorithms support both conjunctive queries and ranking queries. Then proposed another algorithm called

FlexRPSet, which gives one additional parameter K to allow users to make a trade-off between result size & efficiency. Also adopt an additive approach to let the users make the trade-off conveniently. FlexRPset produce fewer representative patterns than RP-Local. An efficient algorithm that is developed for solving the issues and also achieves high performance and good scalability.

6. ACKNOWLEDGMENTS

With all respect and gratitude, I would like to thank all people who have helped us directly or indirectly for the paper presentation .I would like to thank my guide, Prof.N.K.Zalte, for his guidance and support. I will forever remain grateful for the constant support and guidance extended by guide, in making this report. Through our many discussions, he helped me to form and solidify ideas. The invaluable discussions I had with him, the penetrating questions he has put to me and the constant motivation, has all led to the development of this report.

I wish to express my sincere thanks to the Head of department, Prof.N.R.Wankhade also grateful thanks to the departmental staff members for their support. With deep sense of gratitude I thank to our Principal Prof. Dr. V.J.Gond and Management of the Kalyani Charitable Trusts for providing all necessary facilities and their constant encouragement and support.

I would also like to thank to my friends for listening to my ideas, asking questions and providing feedback and suggestions for improving our ideas.

7. REFERENCES

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

[2] I.D. Felipe , V. Hristidis, and N. Rishe, "Keyword search on spatial database ," in Proc. Int. Conf. Data Eng.,2008, pp.656-665.

[3] J. Fan, G.Li, L.Zhou, S. Chen, and J.Hu, " Seal: Spatio-textual similarity search," Proc. VLDB Endowment, vol.5, no. 9, pp. 824-835, 2012.

[4] G. Grahne and J. Zhu, "Efficiently using prefix-trees in mining frequent itemsets," in Proc. FIMI, 2003.

[5] Guimei Liu, Haojun Zhang, and Limsoon Wong , "A Flexible Approach to Finding Representative Pattern Sets," vol.26, no. 7, pp. 1-1, 2014.

[6] T. W. Yan and H. Garcia-Molina, "Index structures for selective dissemination of information under the boolean model," ACMTrans. Database Syst., vol. 19, no. 2, pp. 332–364, 1994.

[7] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma. Hybrid index structures for Location-based web search. In CIKM, 2005.

[8] Y.-Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In IGMOD Conference, pages 277–288, 2006.

[9] R. Hariharan, B. Hore, C. Li, and S. Mehrotra. Processing spatial-keyword (SK) queries in geographic information retrieval (GIR) systems. In SSDBM, 2007.

[10] I. D. Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In ICDE, 2008.

[11] G..cong, CS. Jensen,D.Wu.Efficient retrival of the top-k most relevant spatial web objects PVLDB, 2009