ISSN No. 0976-5697

Volume 5, No. 1, Jan-Feb 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Probability Based String Transformation in Spatial Databases

Dr. M.Bala Ganesh[1], Mrs. V.Sathya[2] and Mrs. M.Priya[3]
[1]Professor & Head, [2]Assistant Professor, PG Scholar[3]
[1,2,3]Department of Computer Science and Engineering
Sembodai Rukmani Varatharajan Engg., College. Sembodai
bmbalaganesh@rediffmail.com[1], ath.saro@yahoo.co.in[2] and shyamnithy@gmail.com[3]

Abstract: Searching a string in large spatial database is a challenging task since it deals with geo-spatial capabilities. Specifically, the range queries are augmented with a string similarity search predicate in Euclidean space. Here, an probabilistic based string transformation techniques is proposed to perform efficient and effective query keyword search in IR^2 index tree .The probabilistic based string transformation techniques embeds a log linear model which is used for learning based string transformation and top k candidate generation algorithm for generating optimal top –k candidate strings. It is possible to systematically illustrate combination of these techniques in Euclidean space and searching the index to find the most relevant answers.

Keywords: Approximate string search, probabilistic string transformation, spatial databases, IR^2 tree, String Predicates

I. INTRODUCTION

A lot of searching algorithm, for searching a spatial object is available. Some of the algorithm based on approximate string search. Some of them are based on exact string search. In both the cases, the main problem is creating a list of string predicates. Approximate String search is a special case of exact search, and it is clear that keyword search by approximate string matches has a much larger pool of applications. Approximate string search is necessary when users have a fuzzy search condition, or a spelling error when submitting the query, or the strings in the database contain some degree of uncertainty or error. A straightforward solution to any spatial query is to use any existing techniques for answering the spatial component of a query and verify the approximate string match predicate either in post-processing or on the intermediate results of the spatial search. It is referred as the spatial solution.

The baseline spatial solution is based on the Dijkstra's algorithm. Given a query point q, the query range radius r, and a string predicate, expands from q on the road network using the Dijkstra's algorithm until reach the points distance r away from q and verify the string predicate either in a post-processing step or on the intermediate results of the expansion. Its performance degrades quickly when the query range enlarges and/or the data on the network increases. This motivates us to find a novel method to avoid the unnecessary expansions, by combining the pruning from both the spatial and the string predicates simultaneously.

Similarly, another straightforward solution is to build a string matching index and evaluate only the string predicate, completely ignoring the spatial component of the query. After all similar strings are retrieved points that do not satisfy the spatial predicate are pruned in a post-processing step. This is called as string solution. First, the string solution suffers the same scalability and performance issues as the spatial solution. Second, it is possible to enable the efficient processing of standard spatial queries while being able to answer queries additionally in existing spatial databases. Another interesting problem is the selectivity estimation.

String transformation can be defined in the following way. Given an input string and a set of operators, we are able to transform the input string to the k most likely output strings by applying a number of operators. Here the strings can be strings of words, characters, or any type of tokens. Each operator is a transformation rule that defines the replacement of a substring with another substring. The likelihood of transformation can represent similarity, relevance, and association between two strings in a specific application. Although certain progress has been made, further investigation of the task is still necessary, particularly from the viewpoint of enhancing both accuracy and efficiency, which is precisely the goal of this work.

This paper introduces a new technique to find the suitable solution for string predicates by applying probabilistic string transformation mechanisms. The probabilistic string transformation approach consists of two phases learning and generation. It automatic the learning process and generate the top-k candidate for the given string predicate using keyword. This reduces the comparison space for string predicate. It improves the efficiency of the searching in spatial database.

II. RELATED WORK

Keyword search on spatial databases [4] proposed a method to efficiently answer top-k spatial keyword queries, which is based on the tight integration of data structures and algorithms used in spatial database search and Information Retrieval (IR^2). The method consists of building an Information Retrieval R-Tree (IR^2-Tree), which is a structure based on the R-Tree. The method can be applied to arbitrarily-shaped and multi-dimensional objects. A solution which is dramatically faster than current approaches and is based on a combination of R-Trees and signature files techniques. Keyword search spatial databases: Towards searching by document [8] proposed a new index called the bR*-tree, which is an extension of the R*-tree.

Retrieving top-k prestige-based relevant spatial web objects [1] proposed a new type of query, called Location-aware top-k Prestige-based Text retrieval (LkPT) query that

retrieves the top-k spatial web objects ranked according to both prestige-based relevance and location proximity. Efficient retrieval of the top-k most relevant spatial web objects [2] introduced a new indexing framework for location ware top-k text retrieval. The framework leverages the inverted file for text retrieval and the R-tree for spatial proximity querying.

Cost-based variable-length-gram selection for string collections to support approximate queries efficiently [7] uses variety of functions to measure the similarity between strings, including edit distance, Jaccard similarity, and cosine similarity. Spatial Approximate String Search [3] provides a complete study for spatial approximate string queries in both the Euclidean space and road networks. It uses the edit distance as the similarity measurement for the string predicate and focus on the range queries as the spatial predicate. The query is referred as spatial approximate string (SAS).

Definition (SAS Query): An SAS query Q: (Qr, Qs)
retrieves
The set of points A = Ar ∩ As.

Another work related to the spatial databases are selectivity estimation of approximate string queries. Selectivity estimation is useful for query optimizers were the result size estimation is needed so the optimizer can run the query in the most efficient way. It can also be used to give the user a feedback about the size of a result from a query. R-tree index based partitioning mechanisms used to constructing a tree data structure in which each level introduces more details on the data. The groups are partitioned trying to minimize the area of the regions, overlap, etc. The R-tree method creates many non-uniform regions.

A. Minimum Bound Rectangle:

The Create Minimum Bounding Rectangle tool is used to create a minimum bounding rectangle based on the combined spatial extent or envelope of one or more selected features. The rectangle that is created is either a polygon or poly line feature, depending on the template you are using as the target.

The minimum bounding rectangle (MBR), alternatively, the minimum bounding box (MBB), is the single orthogonal rectangle that minimally encloses or bounds the geometry of a geographic feature or the collection of geometries in a geographic data set. It can be used to describe the extent of either vector or raster data. All coordinates for the vector feature or data set or the raster grid fall on or within this boundary. In its simplest and most common form, the MBR is a rectangle oriented to the x - and y -axes, which bounds a geographic feature or a geographic data set. It is specified by two coordinates: the minimum x - and y -coordinates (x min, y min), at the lower left of the coordinate space, and the maximum x - and y -coordinates (x max, y max), at the upper right.

Our proposed work consists of three steps. Get the data relevant to the given query from the spatial database and construct IR^2 tree. Next construct the possible combination of top-k string by applying probabilistic string transformation approach. Compare the result and get the output using adaptive algorithm.

III. MODEL USED IN SPATIAL DATABASES

A. String Transforamtion:

String Transformation may be based on Learning approach and approximate string search. Dreyer *et al.* [10] also proposed a log linear model for string transformation, with features representing latent alignments between the input and output strings. Finite-state transducers are employed to generate the candidates. Efficiency is not their main consideration since it is used for offline application. Our model is different from Dreyer *et al.*'s model in several points. Particularly our model is designed for both accurate and efficient string transformation, with transformation rules as features and non-positive values as feature weights.

There are two possible settings for string transformation. One is to generate strings within a dictionary, and the other is to do so without a dictionary. In the former, string transformation becomes approximate string search, which is the problem of identifying strings in given dictionary that are similar to an input string. In approximate string search, it is usually assumed that the model is fixed and the objective is to efficiently find all the strings in the dictionary.

Most existing methods attempt to find all the candidates within a fixed range and employ *n*-gram based algorithms or trie based algorithm. There are also methods for finding the top *k* candidates by using *n*-grams [11],[12]. Efficiency is the major focus for these methods and the similarity functions in them are predefined.

Out of these two, our model consists approximate string search is the best foe search in Euclidean Space. More over our model consists of log linear model for string transformation. There is a possibility to perform searching both in terms of efficiency and accuracy. A number of transformations are relevant to this matching task. These include conference and journal abbreviations(VLDB -> Very Large Data Bases), subject related abbreviations (Soft -> Software), date related variations (Nov -> November, and '76 -> 1976), number related abbreviations(8th -> Eighth), and a large number of variations which do not fall into any particular class (pp -> pages, eds->editors).

B. Model:

In this system, the notion of SAS queries is formalized [3]. The index IR^2 for answering spatial queries is introduced.

A probabilistic string transformation approaches in fig 1 is used to create inverted index used to give solution for string predicate. This approach consists of two phases learning and generation. In learning phase applies the log linear model to generate rules for string predicates automatically. In generation phase applied Top-K candidate pruning algorithm for pair of string using rules obtained in learning phase. Instead of storing all the candidate of q–gram in inverted index, store only the TOP-K candidate or the string in the inverted index. It gives better search result than the existing system.

The log linear model is defined as a conditional probability distribution of an output string and a rule set for the transformation given an input string. The learning method is based on maximum likelihood estimation. Thus, the model is trained toward the objective of generating strings with the largest likelihood given input strings. The

generation algorithm efficiently performs the top *k* candidates generation using top *k* pruning. It is guaranteed to find the best *k* candidates without enumerating all the possibilities. An Aho-Corasick tree is employed to index transformation rules in the model. When a dictionary is used in the transformation, a trie is used to efficiently retrieve the strings in the dictionary.

The inverted index file is stored separately, for two reasons: First, it is more efficient to store each inverted file contiguously, rather than as a sequence of blocks or pages that are scattered across a disk. Second, the inverted file can be distributed across several machines while this is not easily possible for the R-tree. An inverted file consists of the following two main components. A vocabulary for all distinct terms is available in a collection of documents. Now searching for the spatial object is made.

C. Learning:

All the possible rules are derived from the training data based on string alignment. Fig. 2 shows derivation of character-level rules from character-level alignment. First we align the characters in the input string and the output string based on edit-distance, and then derives rules from the alignment. Expand the derived rules with surrounding contexts. We consider expanding only 0 to 2 characters from left and right sides. Derivation of word-level rules can be performed similarly. If a set of rules can be utilized to transform the input string *si to* the output target string *so*, then the rule set is said to form a "transformation" for the string pair *si and so*

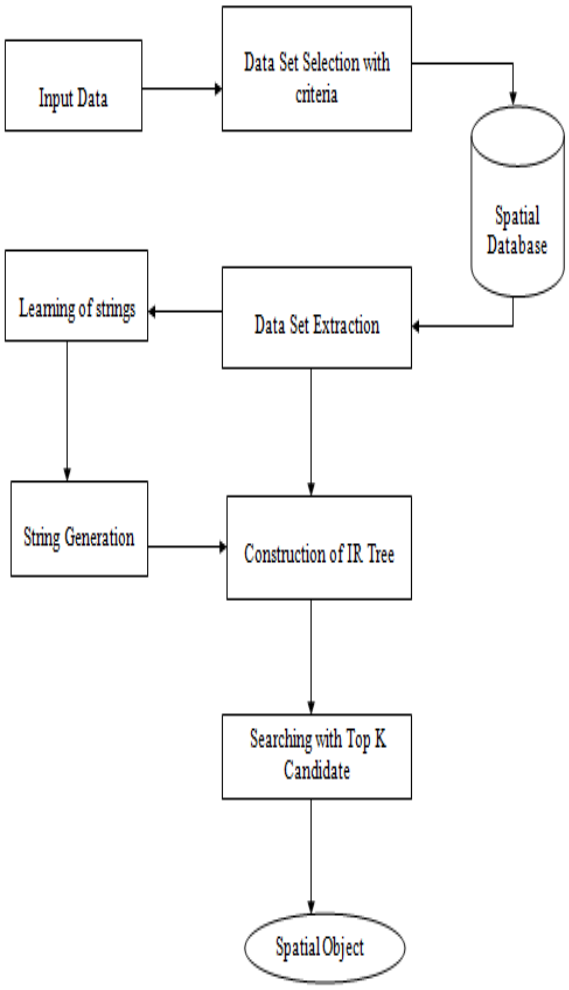

Figure 1. Probability String transformation Model in Spatial Databases

Note that for a given string pair, there might be multiple possible transformations. For example, both ("n"->"m", "tt"->"t") and ("ni"->"mi", "t$"->$) can transform "nicrosoftt" to "microsoft". We assume that the maximum number of rules applicable to a string pair is predefined. As a result, the number of possible transformations for a string pair is also limited.

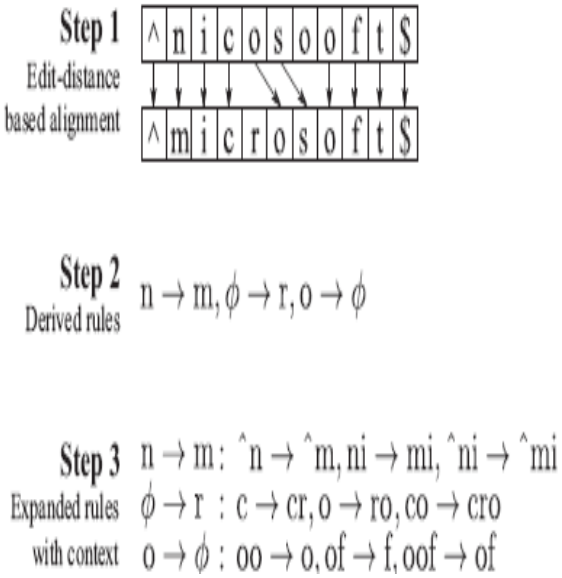

Figure 2 Example of Rule Extraction

D. String Generation:

The rule index stores all the rules and their weights using an Aho-Corasick tree (AC tree) which can make the references of rules very efficient.

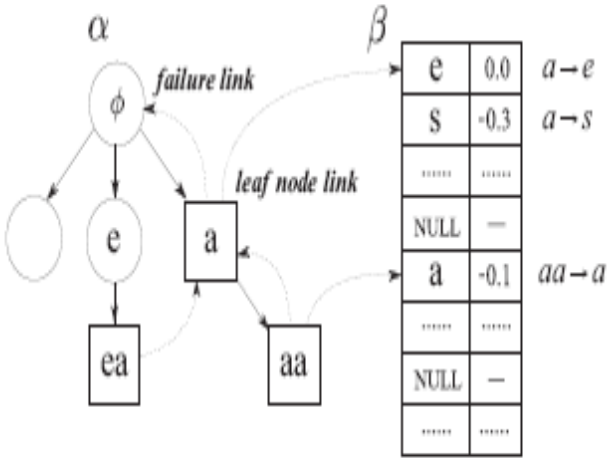

Figure 3.Rule index based on Aho Corasick tree

The AC tree is a trie with "failure links", on which the Aho-Corasick string matching algorithm can be executed. The Aho-Corasick algorithm is a well-known dictionary-matching algorithm which can quickly locate the elements of a finite set of strings within an input string. The time complexity of the algorithm is of linear order in the length of input string plus the number of matched entries. In string generation, given an input string, we first retrieve all the applicable rules and their weights from the AC tree in time complexity of input string length plus number of matched entries.

IV. CONSTRUCTION OF IR^2 TREE

The query is divided into two predicates. The spatial predicates are solved with the help of spatial Databases. String transformation techniques applied to extracted data. There are two processes shown in Fig 4, learning and generation. In the learning process [9], rules are first extracted from training string pairs. Then the model of string transformations constructed using the learning system, consisting of rules and weights. In the generation process, given a new input string, the generation system produces the top k candidates of output string by referring to the model stored in the rule index.

In this method, the model is a log linear model representing the rules and weights, the learning is driven by maximum likelihood estimation on the training data, and the generation is efficiently conducted with top-k pruning. The top k candidate is now made to be available in Inverted File associated with the R tree.

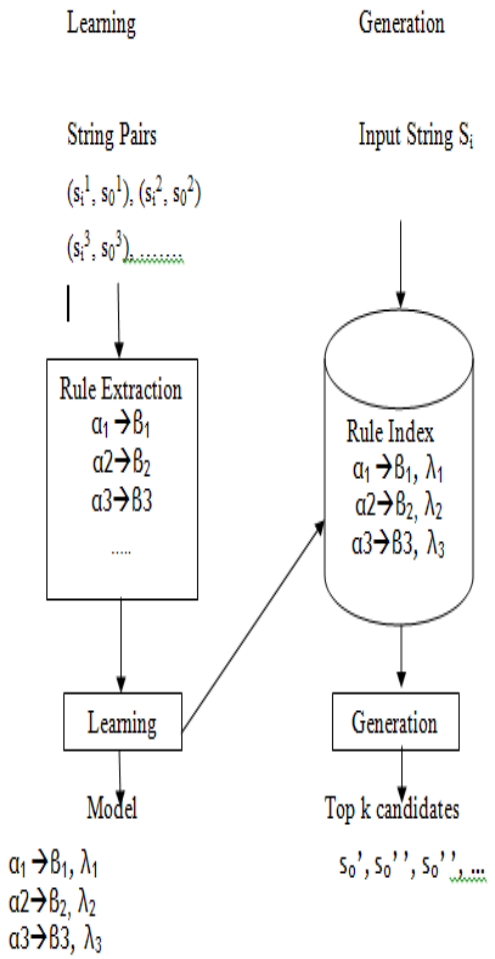

Figure.4 Learning and Generation

Using the extracted data and the top-k candidate the IR^2 tree is constructed [8]. The R-tree is arguably the dominant index for spatial queries, and the inverted file is the most efficient index for text information retrieval. These were developed separately and for different kinds of queries. The IR-tree is essentially an R-tree, each node of which is enriched with reference to an inverted file for the objects contained in the sub-tree rooted at the node as shown in Fig 5

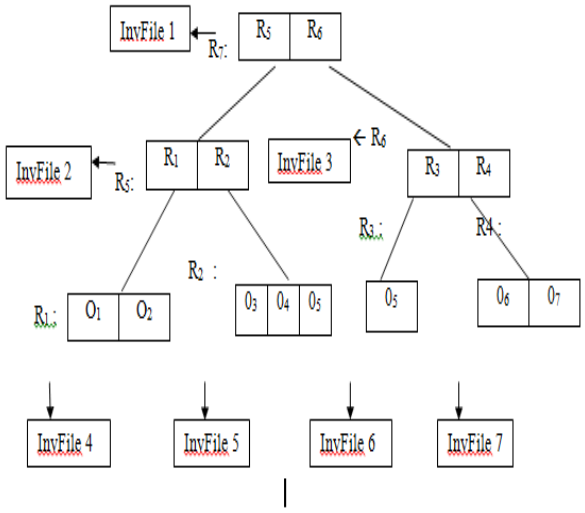

Figure 5.IR2 tree with Inverted File

For searching only the top k candidate for the input string are referred. So that the search space is reduced and there is a possibility of reducing time for searching.

A. Searching Alogrithm:

The given query is divided into two parts .For example the query "Find the theatre with facility="AC" and Balcony" within 2 km. The query consists of spatial predicates Q_s and Q_t.

The string predicates Q_t is given as input to the learning .It will produce the rule model as output. Based on this model for the given input, the string generation algorithm produces the top k candidate .This top k candidates available within the inverted file associated with IR^2 tree.

For processing the query, we exploit the pruning in spatial predicates. The location of the objects is identified first using MBR (Minimum Bound Rectangle).The searching from the root node gives the path to the specific region for the object. If the region of the object id identified properly, then the inverted file is used to search the string predicates for the given query. The inverted File has only top k candidate searching will result in good time complexity and space..

V. CONCLUSION

A comprehensive study for spatial approximate string queries in the Euclidean space. The existing paper uses edit distance as the similarity measurement for the string predicate and focus on the range queries as the spatial predicate. The problem of query selectivity estimation for queries in the Euclidean space is addresses. The concepts of edit distance are replaced with the probabilistic approach for finding string similarity. Using this approach there is a possibility of introducing automation for string predicate comparison. A dynamic programming algorithm is proposed for computing a tight lower bound on the number of strings used to find similar strings in order to improve query performance. An efficient model for automatically generating a high-quality top-k candidate dictionary and perform searching is proposed.

VI. REFERENCES

[1] X.Cao, G.Cong, and C.S.Jensen, 'Retrieving top-k prestige-based relevant spatial web objects' Proc. VLDB.2010

[2] G.Cong, C.S.Jensen, and D.Wu, 'Efficient retrieval of the top-k most relevant patial web objects' PVLDB, 2(1):337–348.2009

[3] Feifei Li, Bin Yao, Mingwang Tang, and Marios Hadjieleftheriou, 'Spatial Approximate String Search', IEEE Transaction on Knowledge and Data Engineering 2013.

[4] D.Felipe, V.Hristidis, and N.Rishe, 'Keyword search on spatial databases' in ICDE, pages 656–665,2008.

[5] S.Ji, G.Li, C.Li, and J.Feng, 'Efficient interactive fuzzy keyword search', in Proceedings of the 18th international conference on Worldwide web ,2009.

[6] S.Sahinalp, M.Tasan, J.Macker, and Z.Ozsoyoglu,'Distance based indexing for string proximity search', In ICDE, pages 125–136,2003

[7] X.Yang, B.Wang, and C.Li, 'Cost-based variable-length-gram selection for string collections to support approximate queries efficiently', In SIGMOD, pages 353–364. 2008

[8] D.Zhang, Y.M.Chee, A.Mondal, A.K.H.Tung, and H.kitsuregawa, 'Keyword search spatial databases: Towards searching by document' , In ICDE, pages 688–699 2009.

[9] Ziqi Wang, Gu Xu, Hang Li, and Ming Zhang , 'A Probabilistic Approach to String Transformation', IEEE Transaction on Knowledge and Data Engineering 2013

[10] M. Dreyer, J. R. Smith, and J. Eisner, "Latent-variable modeling of string transductions with finite-state methods," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 1080–1089.

[11] S. Ji, G. Li, C. Li, and J. Feng, "Efficient interactive fuzzy keyword search," in *Proceedings of the 18th international conference on Worldwide web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp.371-380

[12] R. Vernica and C. Li, "Efficient top-k algorithms for fuzzy search in string collections," in *Proceedings of the First International Workshop on Keyword Search on Structured Data*, ser. KEYS '09. New York, NY, USA: ACM, 2009, pp. 9–14.