International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 6763-6772

http://www.ripublication.com

CFIR Tree – A Novel Method to Build Index For Handling Geographic Document Search

A.Krishna Mohan and MHM Krishna Prasad

Associate Professor Department of CSE JNTUKakinada
Krishna.ankala@gmail.com , Krishnaprasad.mhm@gmail.com

ABSTRACT

In response to a location based query, a geographic search engine is expected to return the documents that are highly relevant to both textual and spatial information. To access the documents in short time they have to be indexed properly. As there are many documents indexing plays a vital role. In this paper, we propose an efficient index called CFIR Tree i.e., CosInfo Feature Clustering Information Retrieval tree, which indexes textual and spatial information in an integrated way. CFIR Tree along with CosInfo Feature Clustering technique performs spatial, textual filtering and relevance computation. Many applications like Google maps are retrieving the information for both textual and geographical information. These kinds of applications need less computational time and high accuracy. To provide the solution to those types of problems, authors propose a new novel and efficient cosInfo method for efficient clustering of textual information. This makes searching easy and with this proposed algorithm, satisfactory results are obtained.

Keywords Geographic document searching, indexing, Goecode, feature clustering

1 INTRODUCTION

Internet made many things easy in this modern world. It is the best way to obtain information these days. To store and access the required information effective index must be designed because one by one checking of documents for required information is a tedious task. A search engine is said to be efficient if it retrieves documents that are highly relevant, and retrieved in a short latency [3][6]. To do that an adept index is needed. Index is a data structure that improves the speed of data retrieval operations on a database table. It helps in efficient retrieval and access of required documents. Location based queries are the queries which are related to some location. Location based query are used in many ways like finding weather in particular location, knowing tourist spots. In some cases interviewer wants to check the capability of the

applicant. At times the applicant is given some unfamiliar address as the interview venue. Another instance let a user say a foreigner wants to know information about TajMahal. The tourist want to know the information regarding TajMahal and also nearby best hotel to stay. Tourist just searches for TajMahal, Agra and nearby hotels in such type of case.

The search engine must search location Agra. In our searching technique, the process is as follows:

First it checks the root node, it moves from World to Asia and from there to India and to Delhi. Under the Delhi location there are still other locations, moving to Agra, the search engine searches for tourist places cluster. Every location has its own clusters like tourist places, temples, offices, hotels, etc. Now TajMahal comes under tourist places so the information related to TajMahal is extracted and also hotels is also considered "and" and "nearby" are stop words they are not considered. And finally the information having TajMahal and hotels is retrieved.

2. PROBLEM DEFINITION

Documents set be denoted as D= $\{d_1, d_2, \ldots, d_n\}$ which consists of say n pages. Each page consists of set of textual keywords T_d and set of spatial locations S_d. Location based query is the query which specifies set of textual keywords Tq and spatial scope Sq. In this paper, query is represented as Q (Tq, Sq) and each document as $d(T_d, S_d)$.

2.1 General Textual Relevance computation:

A document's' is said to be textually relevant to query Q(Tq, Sq) if all or some of the textual keywords of the query are present in document [1][4].

2.2 Spatial Relevance computation:

A location in the document'd' is said to spatially related to query location Sq, if document location S_d overlaps [1] either completely or partial with the query location S_q.

2.3 Geographic web page searching:

Searching is finding the documents that are both textually and spatially related to a given query. In technical way, Geographic document search engine identifies the documents in the document set that are having high combined relevance to the query Q (Tq, Sq) in both textual and spatial aspects. The documents are chosen such that it matches the location and the highest term frequency.

Obtaining the documents which are spatial and textual relevant are to be retrieved in short time.

3 CFIR TREE

CFIR tree is a tree data structure which is used as an index to handle location based queries. CFIR tree is designed such that it performs spatial clustering first and then textual filtering. Here first spatial filtering is done so that the search space can be

abridged because there may be many documents that are textually related but only very few of those are bounded within spatial scope. Now textual clustering is done to reduce search cost and search space. Finally, based on the frequency conditions the documents relevant are retrieved. As soon as k documents are obtained the search process halts.

Coming to the design issue, index structure must be designed in proper way, as each textual word in documents is treated as a dimension. Document space need to cover many very high dimensional spaces. Such high dimensionality is a severe obstacle for classification algorithms. We use efficient CosInfo feature clustering method for efficient clustering of Textual indexing. In addition to that issue spatial locations and textual words have their own representations and measurements. So index must integrate these two aspects so that they must be compatible.

Our CFIR Tree is designed to perform spatial filtering, textual filtering, relevance computation, and ranking simultaneously. Even storage and access overheads are considered.

3.1 CFIR Tree Structure

CFIR tree is designed in such a way that it clusters spatial documents and textual documents under various granularities [5]. All the spatially related documents are clustered so that any document that does not belong to that region requested by the user, can be pruned as and then as unrelated. All textual words are represented as clusters. Each node has document précis [2] such that if the query spatial scope matches with that node then it can traverse according to the nodes pointing it. After reaching leaf nodes, clusters are scanned such that only interested clusters are searched for required information.

CFIR tree is a collection of nodes. It consists of a root node, few non leaf nodes, and few leaf nodes. All the non leaf nodes consist of document précis. Document précis is nothing but collection of information regarding node's spatial region, number of documents that come under that particular node. It is shown in fig 1. In brief, let the non leaf node be node i, then assuming node i will have

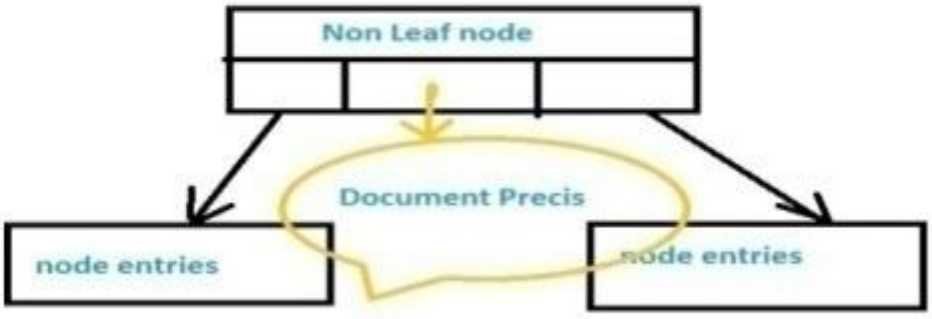

Figure 1: Non leaf node representation

Many children nodes are represented as:

1. M_i: It is the Minimal Bounding Box that covers all the locations of the documents under node i. It is nothing but a small rectangular region that covers all the locations in the document set under the node i.
2. $|W_i|$: It is the cardinality of the documents that come under the node i. i.e., the number of documents that come under node i.

Complete tree structure can be seen in figure 2.

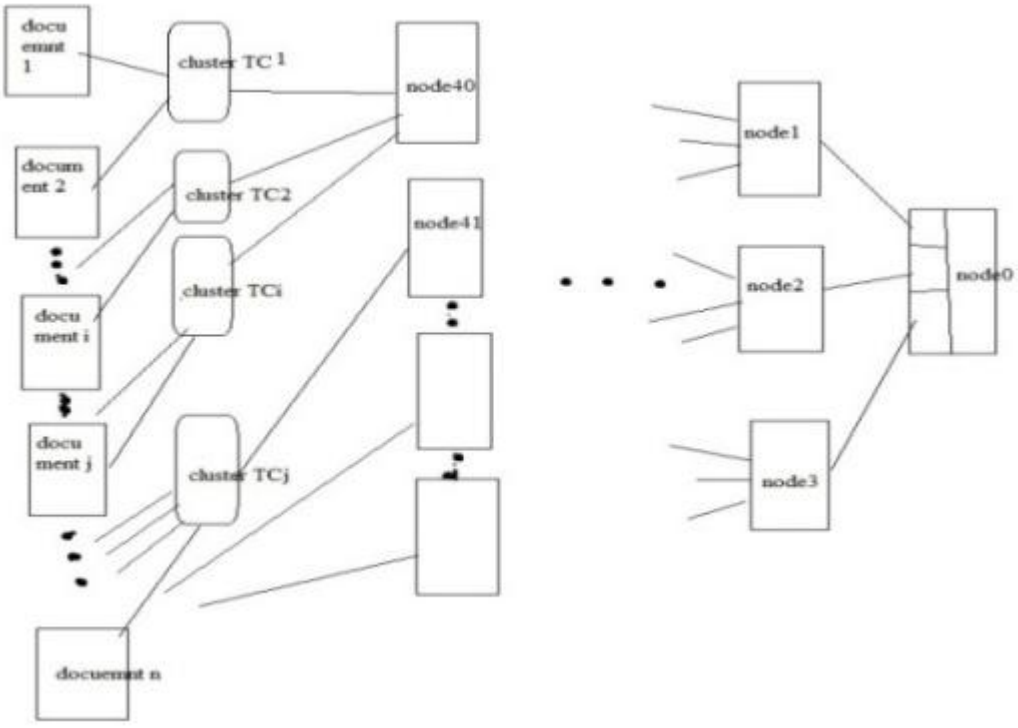

Figure 2:General CFIR Tree Structure Representation.

3.2 CFIR Tree operations

Major operations performed on any Tree structures are insertion, deletion, traversal, and update. First tree has to be constructed, for that set of documents is needed. As mentioned earlier, CFIR tree first spatially clusters all the documents. i.e., all the spatially related locations are grouped into a set[2][3][5][7].

CFIR tree construction involves a bottom up fashion. The general way is shown in the figure 2.

Generalization is done in CFIR tree construction.

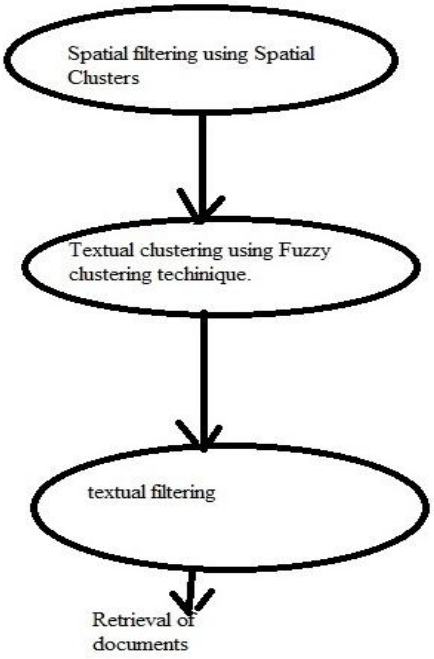

Figure3: General Structure of Searching process

The processing operation is performed as shown in figure 3. Here we assume that every document is associated with one location.

3.3 Algorithm for CFIR tree construction:

Input: A document set, W; Minimal node Fan out, Min; Maximal node Fan out, Max;
Output: Root of CFIR tree.
Method:

1: $N_i \leftarrow \emptyset$

2: for each d ϵ D do
3: {
4: geocode d and represent S_d with MBB M_d;

5: if $\exists i \epsilon N_i$, $M_i = M_d$ then
6: update d to i's document set D_i;
7: else
8: create new entry i;

9: set $M_i \in M_d$ and also D_i {d};
10: N_i N_i {i};
11: }
12: for each i N_i do
13: {

14:

Built clusters based on fuzzy split function .construct inverted list for each word in the document of N_i;

15: }
16: while $|N_i| > max$ do
17: {
18: cluster N_i based on the min and max into nodes represented as new entries $N`_I$;

19: Prepare document précis for i in $N`_I$;
20: N_i $N`_I$;
21: }
22: create root node to layer N_i and its document précis;
23: return root node as output;

In brief, (Line 1) first a root node is initialized to null and then (Line 2–11)every webpage in the webpage set is geocoded i.e., searched for location in the document with the help and all the locations of the web pages are represented using Maximum Bounding Boxes. At the same time, clustering is done by the spatially relevant ones. If the document location matches with the existing node set then it can be grouped, otherwise we create another node set. (Line 12-15) clusters are constructed based on the CosInfo Feature Clustering technique and then inverted list for each word in that cluster is also constructed. After that arrange nodes as per max and minimum fan outs and then root is created and returned as output.

Insertion, deletion involves updating of node sets, document précis and these insertion and deletion are similar to R tree operations.

4 CONSTRUCTION OF TEXTUAL CLUSTERS USING COSINFO FEATURE CLUSTERING TECHNIQUE:

The Authors develop a feature clustering procedure which uses the parts of speech components like Nouns, Pronouns, Verbs, Adverbs, and Adjectives as the pre specified clusters. Later an information retrieval function is used to calculate the importance of each feature in the each specified cluster. Then unimportant features are removed from the clusters based on pre specified threshold value which can be identified based on experiments.

Documents in Local System are classified based on the feature terms they contain and their frequency. We cluster documents using Fuzzy Clustering Algorithm used in feature vectors are calculated for documents. To process documents, The bag-of-words model [5] was used. Let the S be a set of n documents. $S = \{f_1, f_2 \ldots f_n\}$. f_i be a one of the document in the set S. Let M be the set of features/words which contains all the words in each and every documents of set S. $M = \{m_1, m_2 \ldots m_m\}$. Each document fi , $1 \le i \le n$, can contain occurrences of different words. It can be represented as $fi = <t_{i1}, t_{i2} \ldots t_{im}>$, where each t_{ij} denotes the number of occurrence of m_j

in document f_i. The feature reduction task is to reduce the number of features in the new word set. The new word set M'= {m'_1, $m'_2 \ldots m'_m$}, such that M and M' will work equally well and shows the results for all the desired properties and applications with the document set S. Let S' be the converted and reduced feature set after feature reduction. So, then $S' = \{f'_1, f'_2 \ldots f'_k\}$. If k is very much smaller than m, then it can be assumed that the computation cost of text classification is drastically reduced. On these feature clusters we can apply some pre-processing techniques like removal invalid terms, removal of stop words. CosInfo Feature clustering method and algorithm is explained in the next sub section.

4.1 Preprocessing:

The preprocessing process consists of removing the invalid terms. At the beginning the document is scanned for finding the various parts of speech. All the words are segregated in to Nouns, Pronouns, Verbs, Adverbs and Adjectives. All the remaining words are ignored. Next step is to assign the class labels to each document. The class labels are of two types. The class label is assigned to C_1 if the Answer document is closer to the Original Answer document other side assigned to classes C_2, C_3 and C_4 for not related. The closeness is decided based on Cosine similarity measure. Later expected information will be calculated to remove the less important words.

4.2 Algorithm for cosinfo feature clustering method:

Input: Set of Documents d_0, d_1, $d_2 \ldots d_n$ d_0 should be the ideal document . It shall contain all the necessary information. The categorization and clustering is done based on the content of d_0.

Output: Set of feature clusters.

Process: Calculate the document similarity between two documents only with grammar word patterns, using cosine similarity formula. Convert the pronouns in to nouns based on situation and identify the subject of sentence from each noun. Get all the nouns from each document. Get the cosine similarity for all the nouns. In the same way get cosine similarity for adverbs, verbs, adjectives. Ignore the remaining parts of speech and words.

Calculate cosine average similarity between each document. Compare the document d_0 with $d_1 \ldots\ldots d_n$ .Categorize all the documents as per the average cosine similarity value as $C_1, C_2, C_3 and C_4$

$$0.9 \le C_1 \le 1.0$$
$$0.5 \le C_2 \le 0.9$$
$$0.2 \le C_3 \le 0.5$$
$$0.0 \le C_4 \le 0.2$$

Divide the words of W in to four clusters based on its parts of speech.
Calculate the expected information I of each word as per equation (8).

```
Initial expected information β= 0.6
Loop (1 to end of all the word patterns)
    If (the I is < β) {
  Add word pattern to the cluster
} else {
 Remove the word pattern from W.
}
end if;
end loop;
Return with the created clusters;
End procedure
```

This algorithm is used to create clusters based on the CosInfo Feature Clustering technique.

5 EXPERIMENTAL RESULTS

In this paper, we have considered data set LATimes'94. The data set LATimes'94 consists of 110,273 documents which includes 2,119 locations. Its average number of words per a document is 504 and number of indexed document words are 90,986. The total size is 421 MB. First, all the location names are extracted from every single document and these locations are geocoded into Maximum Bounding Boxes (MBB). Geocode is done based on the ontology that covers over 1, 29,784 worldwide locations. Considering some factors like area size, population size...etc of locations they are divided as small city, medium city, large city, state, country these MBB are constructed. Experimental results are generated based on the search time by considering some parameters.

We implemented CFIR tree with the C#.net and SQL server 2005 which provides the geographic related dataset directly. Here search time is compared with IR tree and $Hybrid_R$. IR tree is the index structure that searches the textual words and spatial objects in each and every non leaf nodes and traverse in that direction. A rank function is used to rank the documents that are relevant and then retrieves only top k documents. Storage cost of IR tree is more as each and every node has to be checked for the term frequency. $Hybrid_R$ is index structure which is implemented by filtering spatial documents first and then textual relevant ones as R tree is placed on the top of the inverted files.

5.1 Searching Efficiency:

Efficiency is estimated based on some factors which have their own impact on searching. There are two most important factors to be considered. They are:

1. Size of the query spatial scope |S|;
2. Number of requested documents k;

5.1.1 Impact of |S|:

Query spatial scope size is the important factors that change the performance of the searching. The values that suit the query spatial scope (in km) are: 10^2, 20^2, 100^2, 500^2. As the spatial locations are represented as Maximum Bounding Boxes they are measured in the form of square area. So in our CFIR implementation the scope is limited from 10*10 km^2 to 500*500 km^2.

First by keeping k value, i.e., number of documents value constant, say k=100 i.e., retrieval of top 100 documents and varying the query spatial scope size, the results are obtained as per the range of the spatial scope.

For the sample dataset the results are plotted as shown in fig 4.

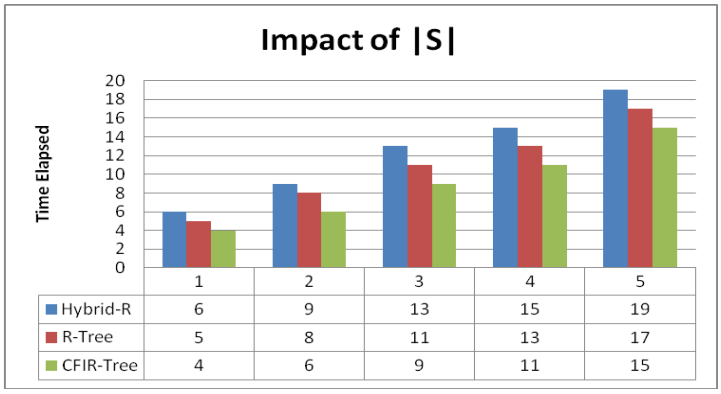

Figure 4: Effect of |S| on the sample data set.

5.1.2 Impact of K:

The number of documents to be retrieved, K is one important factor which is the main estimate of time. The k value can be 10, 30, 50, 100, 300...etc. By fixing |S| =100*100 km^2 and varying k and implementing on the dataset leads to the shown variations in the fig 5. CFIR tree performs well when compared to IR tree structure and $Hybrid_r$. Of all the index structures, CFIR performs best and $Hybrid_r$ performs very badly. Whatever may be the k value, CFIR Tree retrieved the documents in very efficient time.

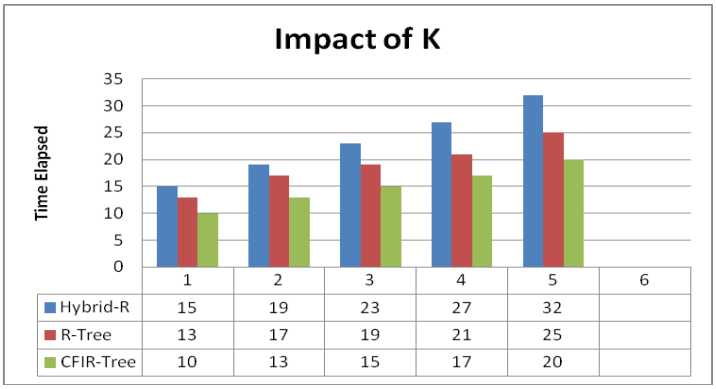

Figure 5: Effect of K on sample data set .

CFIR tree performed sound when compared to other index structures. Even the filtering is done in usual way; the time taken to retrieve documents is very short.

6 CONCLUSION

In this paper, we proposed an efficient index structure namely CFIR Tree, that handles textual filtering, spatial filtering in an adept way. We mainly focused on textual filtering which is implemented by using a technique called fuzzy self clustering technique. The experiment results proved that CFIR Tree is an adept index and performed sound. Future work can be enhanced to perform semantic similarity between query keywords and also CosInfo Feature Clustering to spatial information.

7 REFERENCES

[1] Zhisheng Li, Ken C.K Lee, Baihua Zheng, Wang-Chien, Dik Lun Lee, Xufa Wang, *"IR Tree: An Efficient Index for Geographic Document Search"* IEEE Vol 23, No 4, April 2011.

[2] Khodaei A, Cyrus Shahabi, Chen Li, "*SKIF-P: A point based Indexing and Ranking of Web Documents for Spatial Keyword Search"* Geoinformatica.

[3] L.X.Wang A course in fuzzy systems and control. Prentice-Hall International, Inc., 1997.

[4] E. Amitay, N.Har' El, R. Sivan, A. Soffer, " Web-A-Where: Geotagging Web Content", Proc. ACM Sigir '04, pp 273-280.2004.

[5] I.D. Felipe, V. Hristidis, N.Rishe, "Keyword Search on Spatial Databases", Proc IEEE 24th Int'l conference Data Engg(ICDE '08), pp 656-665, 2008.

[6] Khodaei A, Shahabi C, Li C(2010), "Hybird Indexing and Seamless Ranking of Spatial and Textual Features of Web Documents in DEXA, PP 450-466.

[7] D. Hiemstra, "A Probabilistic Justification for using TF IDF Term Weighting in Information Retrieval", Int'l j .Digital Libraries, Vol 3, No 2, pp-131-139, 2000.