Concept-Based Document Similarity Based on Suffix Tree Document

*P.Perumal
Sri Ramakrishna Engineering College
Associate Professor
Department of CSE, Coimbatore
perumalsrec@gmail.com

R. Nedunchezhian
Sri Ramakrishna Engineering College
Professor and Head
Department of Information Technology, Coimbatore
rajuchezhian@yahoo.co.uk

M. Indra Priya
Sri Ramakrishna Engineering College,
Department of CSE(PG), Coimbatore.
indrapriya27@gmail.com

Abstract- **Document clustering has been studied as a post retrieval document visualization technique to provide an intuitive navigation and browsing mechanism by organizing documents into groups and each group represents a different topic. The clustering techniques are based on four concepts: Data representation model, Similarity measure, Clustering model, and Clustering algorithm. In the previous work, phrase has been considered as an informative feature term for improving the effectiveness of document clustering. In this paper, we propose a Concept-based document similarity to compute the similarities of documents based on the Suffix Tree Document (STD) model. By mapping each node in the suffix tree of STD model into a unique feature term in the Vector Space Document (VSD) model, the concept-based document similarity inherits the term ctf (conceptual term frequency), tf (term frequency), df (document frequency) weighting scheme in computing the document similarity with concept. In this paper the concept-based document similarity is applied to the Hierarchical Agglomerative Clustering (HAC) algorithm to develop a new document clustering approach. The new concept-based model analyzes the terms on the sentence, document, and in corpus levels. The similarity between documents is calculated based on a new concept-based similarity measure (Euclidean distance Measure.). The proposed similarity measure takes full advantage of using the concept analysis measures.**

Key words—Concept-based model, Suffix tree document Model, Suffix tree, Similarity measure, Document clustering.

I. INTRODUCTION

Document clustering has been investigated for improving the performance of search engines by pre-clustering the entire corpus [1]. A document model is a concept that describes how a meaningful set of features, d, is computed from a document's (preprocessed) words [2]. Document models can be considered as vectors of words, the suffix tree document model, related similarity measures are graph-based. Both types of document models provide an efficient means to compute document similarities [2]. Most of the current document clustering methods is based on the Vector Space Document (VSD) model [3].To achieve accurate document clustering, an informative feature term phrase has been considered. A phrase of a document is an ordered sequence of one or more words [4].

The Vector Space Model is used to represents document as a feature vector of the terms (words) that appear in the document set [3]. The suffix tree document model preserves the complete word order information [5]. It defines the similarity between two documents in terms of string overlaps in their common suffix tree [6], [7]. In the suffix tree of STD model, each nodes are mapped into a unique feature term in the Vector Space Document (VSD) model, the phrase-based document similarity inherits the term tf-idf (term frequency and inverse document frequency) weighting scheme in computing the document similarity with phrases. The phrase-based document similarity is applied to the group-average Hierarchical Agglomerative Clustering (HAC) algorithm.

The STD model considers a document as a sequence of words, not characters. A document is represented by a set of suffix substrings, the common prefixes of the substrings are selected as phrases to label the edges (or nodes) of a suffix tree [8]. The STC algorithm was used in their metasearching engine to realtime cluster of the document snippets. It is a linear time clustering algorithm (linear in the size of the document set), which is based

on identifying the phrases that are common to groups of documents. The existing work focus on to combine the two document models in document clustering.

In this paper, a concept-based model is proposed. The proposed model captures the semantic structure of each term within a sentence and document rather than the frequency of the term within a document only. In the proposed model, three measures for analyzing concepts on the sentence, document, and corpus levels are computed.

II. RELATED WORK

Text document clustering has been investigated as a means of improving the performance of search engines by preclustering the entire corpus [1], and a post retrieval document browsing technique. Hierarchical Agglomerative Clustering (HAC) algorithm might be the most commonly used algorithm among numerous document-clustering algorithms.

A. Vector space Document Model

Most of the current documents clustering methods are based on the Vector Space Document (VSD) model and starts with a representation of any document [3]. In the document models (VSD model) words or characters are considered to be the basic terms in statistical feature analysis and extraction. The statistical features of all words are taken into account of the term weights (usually tf-idf) and similarity measures, whereas the sequence order of words is rarely considered in the clustering approaches based on the VSD model

B. *Suffix Tree Document (STD) Model*

The STD model considers a document as a sequence of words, not characters. A document is represented by a set of suffix substrings, the common prefixes of the substrings are selected as phrases to label the edges (or nodes) of a suffix tree. An effective method has not been found in the previous work to evaluate the effect of each phrase in document clustering algorithms [1], [2], [3].

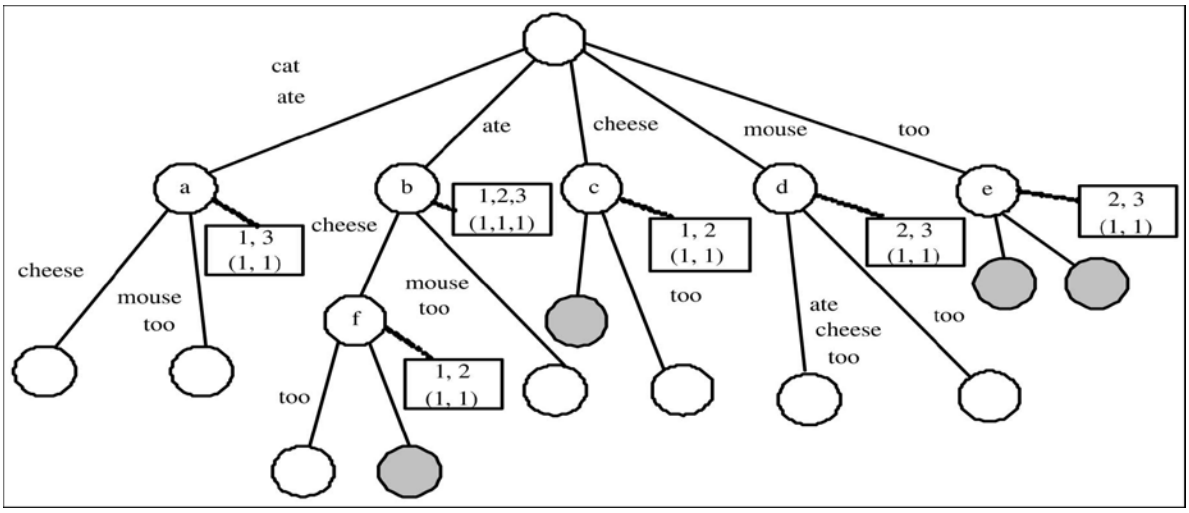

Fig. 1. The suffix tree of tree documents "cat ate cheese," "mouse ate cheese too," and "cat ate mouse too."

In Fig. 1, each internal node is attached to an individual box. The numbers in the box designate the documents that have traversed the corresponding node. Each upper number designates a document identifier, the number below designates the traversed times of the document.

C. Algorithms

In general, the approaches can be categorized into

- Agglomerative solutions
  (e.g., hierarchical clustering)
- Partitional solutions
  (e.g., K-means clustering).

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. In hierarchical clustering the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions takes place. HAC algorithm recursively merges clusters until a stop criterion is met. Each iteration step results in a certain level of clustering. The results depend on the threshold of granulation. There are three variants from this algorithm: Single-link, Complete-link, and Group-average. The group average HAC algorithm is used to develop a new document clustering approach. In existing work, the group-average similarity is chosen for the HAC [6] algorithm, which measures the similarity of two clusters with the average of pairwise similarities of the documents from each cluster. The STC algorithm is developed based on the STD model [8]. The suffix tree of a document *d* is a compact tree containing all suffix substrings of the document d. The suffix tree is composed from three documents. The nodes of the suffix tree are drawn in circles. There are three kinds of nodes in the suffix tree: the root node, internal nodes, and leaf nodes.

The STC algorithm has three logical steps [9].

Step 1: The common suffix tree generation

A suffix tree S for all suffixes of each document in $D = \{d_1, d_2, ..., d_N\}$ is constructed. Each internal ode containing at least two different documents is selected to be a base cluster, which is composed of the documents designated by the box, and labeled by the phrase of the node.

Step 2: Base cluster selection

Each base cluster *B* is assigned a score s *(B):* $s(B) = |B| . f(|P|)$, where |B| is the number of documents in *B*, and |*P*| is the number of words in *P*. Then, all base clusters are sorted by the scores, and the top *k* base clusters are selected for cluster merging in Step 3.

Step 3: Cluster merging

A similarity graph consisting of the *k* base clusters is generated. An edge is added to connect two base clusters B_i and B_j.

D. *Similarity Measure*

The term weights (usually tf-idf, term-frequencies and inverse document-frequencies) [6] of the words are also contained in each feature vector [3]. Term frequency - inverse document frequency (tf-idf) is a commonly used information retrieval technique for assigning weights to individual word terms appearing in all documents. The weight (tf-idf) of each node is recorded in building the suffix tree, and then the cosine similarity measure is used to compute the pairwise similarities of documents. Applying the document similarity to the group-average HAC algorithm (GHAC), we developed a document clustering approach.

E. Stopword or Stop node

Stopwords are frequently occurring, insignificant (unimportant) words that appear in the documents. Stopwords Lists and stemming algorithms are two commonly used information retrieval techniques for preparing text document. Standard Stopwords List and Porter stemming algorithm are used to preprocess the documents to get ``clean" documents. The (tf-idf) weighting scheme has provided a solution to reduce the negative effect of these words, almost all popular document clustering algorithms including STC algorithm prefer to consider these words as new stopwords, and ignore them in their document similarity measure. In the document similarity measure, the term of a word is replaced by the term of a node in the suffix tree and a new definition "stopnode" is proposed, which applies the same idea of stopwords in the suffix tree similarity measure computation.

F. The Phrase Based Document Similarity

In text-based information retrieval, a document model is a concept that describes how a set of meaningful features is extracted from a document. Most of the current document clustering methods uses the VSD model [2] to represent documents. In this paper, we use the symbols *N*, *M*, and *k* to denote the number of documents, the number of terms, and the number of clusters, respectively. We use the symbol *D* to denote the document set of *N* documents that we want to cluster, the C_1, C_2 ... C_k to denote each one of the *k* clusters. Most of the current document clustering method uses the VSD model to represent documents. In the model, each document *d* is considered to be a vector in the *M*-dimensional term space. In particular, the term tf-idf weighting scheme [8], in which each document can be represented as

$$\vec{d} = \{w(1,d), w(2,d), \ldots, w(M,d)\}, \quad (1)$$

where $w(i,d) = (1+\log tf(i,d)).\log(1+N/df(i)), tf(i,d)$ is the frequency of the *i*th term in the document *d*, and $df(i)$ is the number of documents containing the *i*th term. In the VSD model, the cosine similarity is the most commonly used measure to compute the pairwise similarity of two document d_i and d_j, which is defined as

$$sim_{i,j} = \frac{\vec{d}_i \bullet \vec{d}_j}{\left|\vec{d}_i\right| \times \left|\vec{d}_j\right|} \quad (2)$$

III. PREVOIUS WORK

The phrase-based document similarity is presented in the previous work. In this work document clustering methods are based on the Vector Space Document (VSD) model [2]. A word appearing in the documents is usually considered to be an atomic feature term. The term weights (usually tf-idf, term-frequencies and inverse document-frequencies) of the words are contained in feature vector [6]. Different from document models which treat a document as a set of words and ignore the sequence order of the words, the STD model considers a document to be a set of suffix substrings, and the common prefixes of the suffix substrings are selected as the phrases to label the edges of the suffix tree.

The STC algorithm has three logical steps.

Step1: The common suffix tree generations.

Step 2: Base cluster selection.

Step 3: Cluster merging.

In the previous algorithm implementation, a bottom-up search is developed to collect all nodes that traversed by a document.

Algorithm: Bottom-up search for collecting all nodes traversed by a document.

Input *id***:** document id

Output *Node***:** a node list

1: *Leaf* ← List of leaf nodes
2: *Node* ← Empty List of nodes
3: *lnode* ← Empty List of nodes (*lnode* is a list for all leaf nodes matching *id*)
4: **for** each leaf node $leaf_i$ in *Leaf* **do**
5: **if** *id* exist in $leaf_i$ **then**
6: Add $leaf_i$ to *lnode*
7: **end if**
8: **end for**
9: **for** each node v_i in *lnode* **do**
10: **if** v_i has an nonempty phrase **then**
11: Add lf_i to *Node*
12: **end if**
13: $v_i \leftarrow$ parent node of v_i
14: **while** v_i is not the root node **do**
15: Add v_i to *Node*
16: $v_i \leftarrow$ parent node of v_i
17: **end while**
18: **end for**
19: return *Node*

Assuming the average length of the documents is *m* (words), and then there are a total of *N*xm leaf nodes in the suffix tree generated from the *N* documents.

IV. PROPOSED WORK

To achieve an accurate document clustering, the proposed work has been implemented a new concept-based model that analyzes terms on the sentence, document, and corpus levels. The concept-based effectively discriminate non-important terms with respect to sentence semantics and terms. The proposed model can efficiently find significant matching concepts between documents, according to the semantics of their sentences. To analyze each concept at the sentence level a new concept-based frequency measure, called the conceptual term frequency (ctf) is proposed. To analyze each concept at the document level, the concept based term frequency (tf) is used. The process in the measures in a corpus is attained by the proposed algorithm which is called Concept-based Analysis Algorithm.

V. EXPERIMENTAL RESULTS

To evaluate the effectiveness of the document clustering three quality measures are used. The first is F-Measure. The F-Measure is commonly used in evaluating the effectiveness of clustering and classification algorithms .The second measure is Purity. The cluster purity indicates the percentage of the dominant class members in the given cluster. The third measure is Entropy. The third measure is Entropy, which provides a measure of "goodness" for unnested clusters or the clusters at one level of a hierarchical clustering.

VI. PERFORMANCE EVALUATION

The concept based document similarity was implemented based on the suffix tree algorithm and suffix tree document model. HAC algorithm starts with each instance representing a cluster. The algorithm recursively merges clusters until a stop criterion is met. Each iteration step results in a certain level of clustering. The final results depend on the threshold of granulation.

The original STC algorithm selects the 500 highest scoring base clusters for further cluster merging, but only chooses the top 20 clusters as the final result of the cluster.

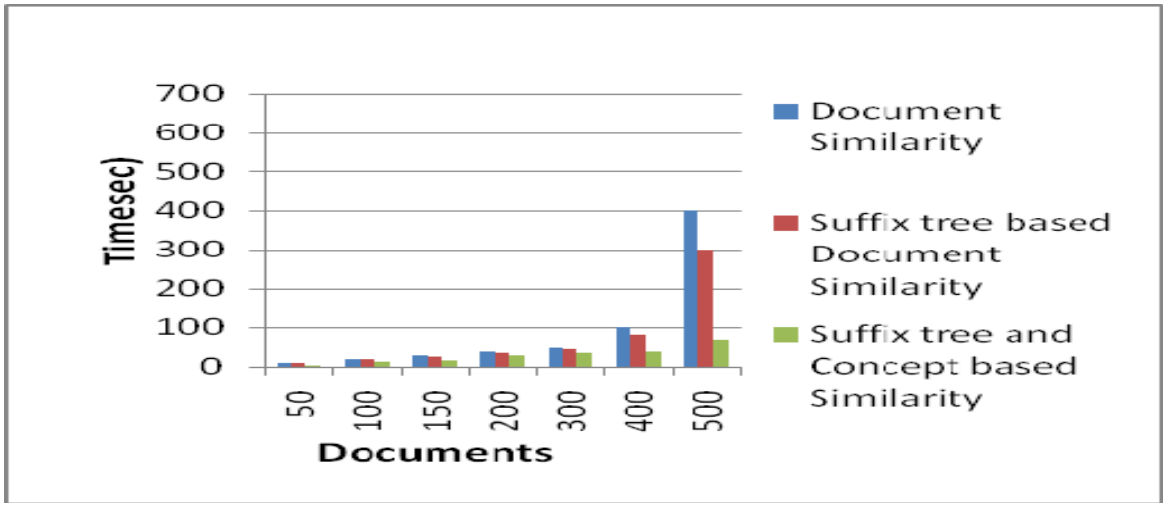

Fig. 2 Documents vs Time

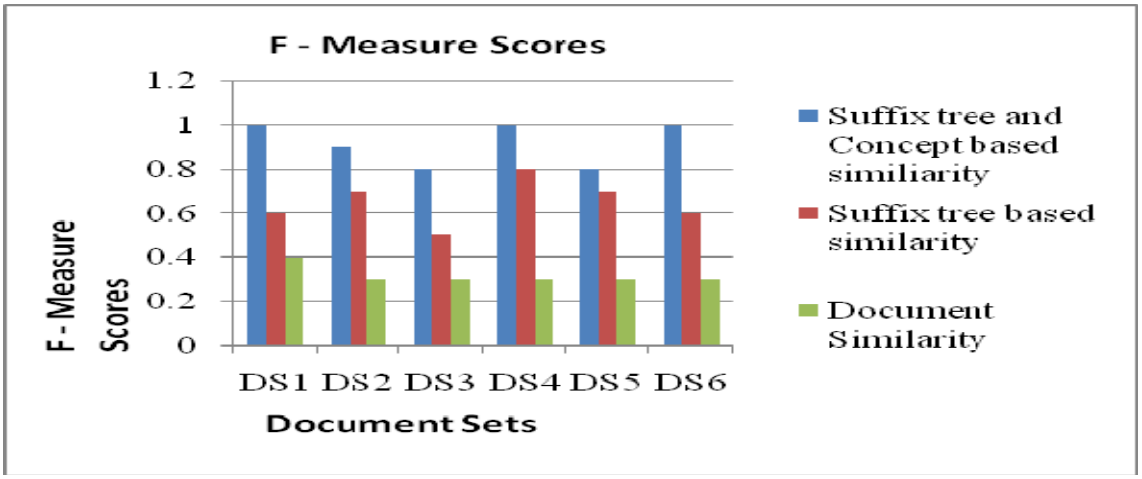

Fig. 3 F-Measure Score

All clusters are generated by the cluster merging of the STC algorithm, and compute the three kinds of measure scores for the clustering results.

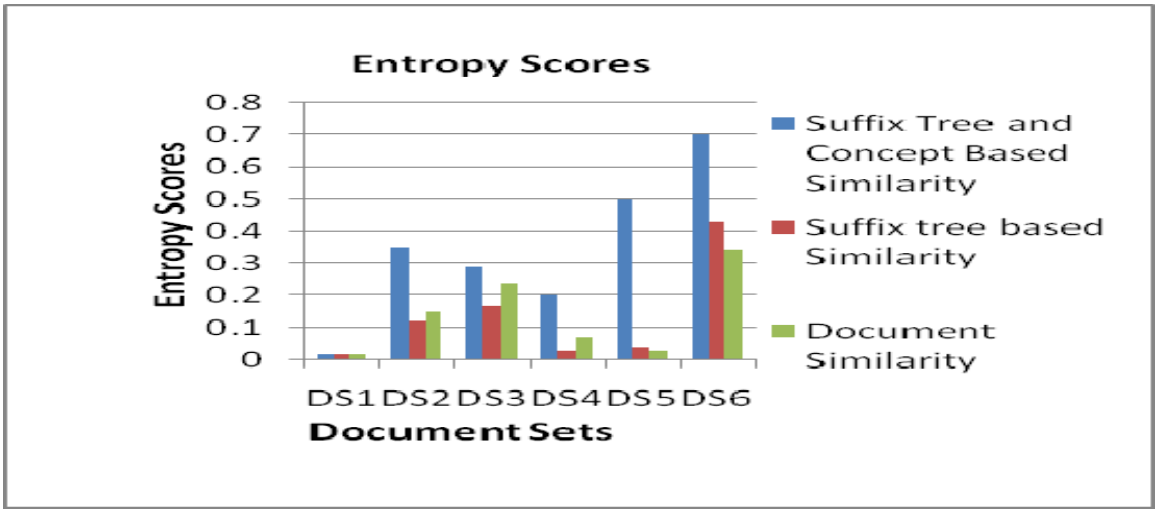

Fig. 4 Entropy Score

In the comparison experiments, data sets are constructed from the 20-newsgroups collection. Figs. 2, 3, 4 and 5 illustrate the F-measure, Purity and Entropy scores computed from the clustering results of clustering algorithm on the document data sets, where STC designates the results of all clusters generated by the STC algorithm.

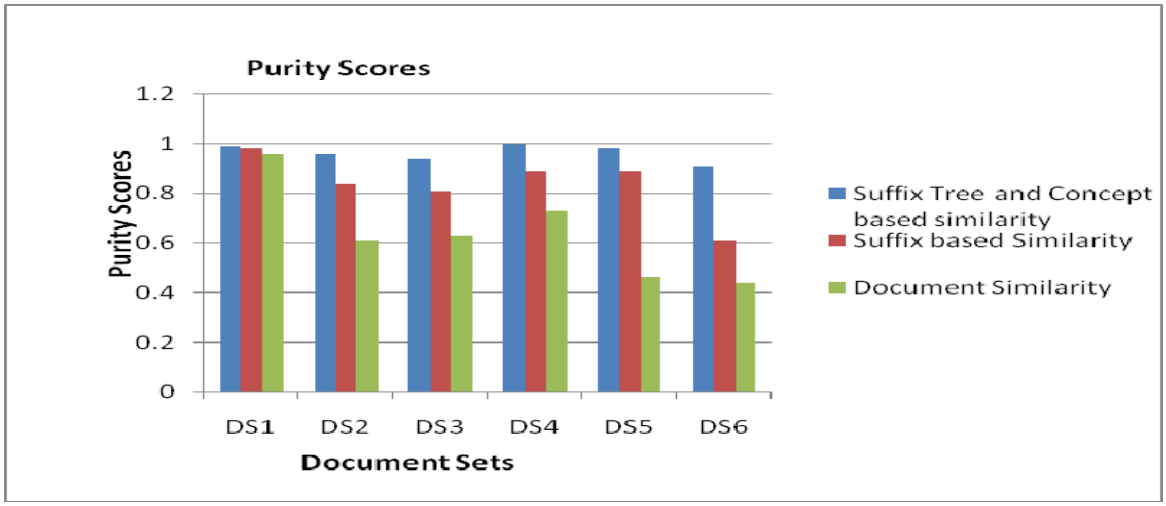

Fig. 5 Purity Score

The STC algorithm allows a document to appear at more than one cluster. This comparison indicates that the concept-based document similarity can efficiently improve the ability of HAC against the noise.

CONCLUSION AND FUTURE ENHANCENENT

A concept based model is composed of and proposed to improve document clustering quality. By exploiting the semantic structure of the sentences in documents, a better text clustering result is achieved. There are a number of possibilities for extending this paper. One direction is to link this work to Web document clustering. Another direction is to apply the same model to text classification. The intention is to investigate the usage of such model on other corpora and its effect on classification.

REFERENCES

[1] Efficient Phrase-Based Document Similarity for Clustering Hung Chim and Xiaotie Deng, Senior Member, IEEE IEEE Trans.Knowledge and Data Eng., vol. 20, no. 9, pp. 1217-1229, Sept. 2008.

[2] K.M. Hammouda and M.S. Kamel, "Efficient Phrase-Based Document Indexing for Web Document clustering," IEEE Trans.Knowledge and Data Eng., vol. 16, no. 10, pp. 1279-1296, Oct. 2004.

[3] G. Salton, A. Wong, and C.S. Yang, "A Vector Space Model for Automatic Indexing," Comm. ACM, vol. 18, no. 11, pp. 613-620,1975.

[4] O. Zamir and O. Etzioni, "Grouper: A Dynamic Clustering Interface to Web Search Results," Computer Networks, vol. 31,nos. 11-16, pp. 1361-1374, 1999.

[5] M. Yamamoto and K.W. Church, "Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus," Computational Linguistics, vol. 27, no. 1,pp. 1-30, 2001.

[6] U. Manber and G. Myers, "Suffix Arrays: A New Method for On-Line String Searches," SIAM J. Computing, vol. 22, no. 5, pp. 935-948, 1993.

[7] D.S. Sven Meyer zu Eissen and M. Potthast, "The Suffix Tree Document Model Revisited," Proc. Fifth Int'l Conf. Knowledge Management (I-Know '05), pp. 596-603, 2005.

[8] C.J. van Rijsbergen, Information Retrieval. Butterworths, 1975.