Document Clustering Using Suffix Tree.
Mrs.Chetana Hemant Nemade
MIT,Academy Of Engineering,Alandi(D),Pune
chetananemade@gmail.com

ABSTRACT
For improving the effectiveness of document clustering phrases has been considered. For computing the pair wise similarity of document use a phrase based document similarity. The term tf-idf weighting scheme is used in computing the document similarity with the phrases which are represented in the STD and VSD model. Our evaluation experiments indicate that the new clustering approach is very effective on clustering the documents of two standard document benchmark corpora OHSUMED and 20newsgroups. The quality of the clustering results significantly surpasses the results of traditional single-word tf-idf similarity measure in the same HAC algorithm, especially in large document data sets. Furthermore, by studying the property of STD model, we conclude that the feature vector of phrase terms in the STD model can be considered as an expanded feature vector of the traditional single-word terms in the VSD model. This conclusion sufficiently explains why the phrase-based document similarity works much better than the single-word tf-idf similarity measure.

Index Terms: Suffix tree, document model, similarity measure, document clustering.

I. INTRODUCTION
The two widely used models for document clustering are Vector Space Document (VSD) model and the Suffix Tree Document (STD) model. Each of these models has their own advantages and disadvantages.

Vector space model is an algebraic model for representing text documents as vectors of identifiers. VSD used in information filtering, information retrieval, indexing and relevancy rankings. Each document is represented as a feature vector of the words that appear in the documents of the dataset. The similarity between two documents is calculated by using the cosine measure, Jaccard measure and Euclidean distance.

Phrases are used in the STD model instead of words to compare documents. A phrase is defined as an ordered sequence of one of more words. 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. STD model is used in the Suffix Tree Clustering algorithms.

F-measure is used to measure the overall quality of the clustering method, the Purity parameter is used to measure the precision of the clusters, and the Entropy parameter is used to count how various classes of documents are distributed within each cluster. We want high F-measure and Purity values and low Entropy values.

II. LITERATURE SURVEY
Hierarchical Agglomerative Clustering (HAC) algorithm might be the most commonly used algorithm among numerous document clustering algorithms. Generally, there are three variants from this algorithm: single-link, complete-link, and group-average. In practice, the HAC algorithm can often
generate high-quality clusters with a tradeoff of the higher computation complexity [2]. K-Nearest Neighbor (K-NN) algorithm is well known for classification [3]. It has also been used for document clustering [4],[5].

Suffix tree is a data structure that admits efficient string matching and querying. It has been studied and used extensively in fundamental string problems and applications such as large volumes of biological sequence data searching[6], approximate string matches [7], and text features extraction in spam e-mail classification [8]. The STD model was first proposed in 1997 [9],[10]. Different from document models which treat a document as a set of words and ignore the sequence order of the words [11], 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 Suffix Tree Clustering (STC) algorithm is developed based on this model and works well in clustering Web document snippets. However, the properties of STD model and STC algorithm have not been analyzed in their papers [12],[13],[14]. Yamamoto and Church [15] presented a method to compute all substrings’ (phrases) term frequencies and document frequencies in large document corpora by using suffix array [16]. Their approach provided an efficient way for extracting phrases and computing their tf-idf weights in a static document set. Another relevant work is the DIG proposed by Hammouda and Kamel [5], which allows for the incremental construction of a phrase-based index for a document set. The quality of clustering achieved by using the hybrid similarity measure based on this model significantly surpassed the approaches based on the traditional VSD model.

III. METHODOLOGY

Let N,M, and k 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 and C1,C2,.....Ck to denote each one of the k clusters.

In the VSD model, each document d is considered to be a vector in the M-dimensional term space. In particular, we usually employ the term tf-idf weighting scheme, in which each document can be represented as

Vector d=(w(1,d),w(2,d),.....,w(M,d))

Where w(i,d)=(1+logtf(i,d)).log(1+N/df(i))

and 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 STD model, we use suffix tree to represent the documents. The suffix tree of a document d is a compact trie containing all suffix substrings of the document d. The similarity of two documents is defined as the more internal nodes shared by the two documents the more similar the documents tend to be. The original STC algorithm is based on the STD model. It has three steps namely: the common suffix tree generation, base cluster selection, and cluster merging. The phrase-based document similarity based on the STD model proposed by the authors [1] is a hybrid of the STD model and the VSD model.

Here we represent each document d as a feature vector of the weights of M node terms in the suffix tree of the documents. The document frequency of each node df(v) is the number of the different documents that have traversed node v; the term frequency tf(v,d) of a node v with respect to document d is the total
traversed times of the document d through node v. The weights are calculated using the same formula as in the VSD model.

Once we have the term weights of all nodes we can use a cosine similarity measure to compute the similarity of any two documents. Let vectors \( d_x = \{x_1, x_2, x_3, \ldots, x_M \} \) and vector \( d_y = \{y_1, y_2, \ldots, y_M \} \) denote two documents \( d_x \) and \( d_y \), where \( x_i \) and \( y_i \) are the weights of corresponding node term \( v_i \), respectively. Then, the similarity of two documents is calculated by the following formula:

\[
\text{sim}_{xy} = \frac{d_x \cdot d_y}{|d_x| \times |d_y|} = \frac{\sum_{i=1}^{M} x_i y_i}{\sqrt{\sum_{i=1}^{M} x_i^2 \sum_{i=1}^{M} y_i^2}}
\]

Let \( C = \{C_1, C_2, \ldots, C_k\} \) is a clustering of dataset \( D \) with \( N \) documents and let \( C^n = \{C_{11}, C_{12}, \ldots, C_{1k}\} \) be the "correct" class set of \( D \). Then the recall of cluster \( j \) with respect to class \( i \), \( \text{rec}(i,j) \) is defined as

\[
\text{rec}(i,j) = \frac{|C_{ij}|}{|C_i|}
\]

The precision of cluster \( j \) with respect to class \( i \), \( \text{prec}(i,j) \) is defined as

\[
\text{prec}(i,j) = \frac{|C_{ij}|}{|C_j|}
\]

Then the F-measure, \( F(i,j) = (2 \times \text{prec}(i,j) \times \text{rec}(i,j)) / (\text{prec}(i,j) + \text{rec}(i,j)) \). Based on this formula, the F-Measure for overall quality of cluster set \( C \) is defined by the following formula:

\[
F = \frac{1}{k} \sum_{i=1}^{k} \max_{j=1\ldots k} F(i,j).
\]

The Purity measure is calculated using the formula:

\[
Purity = \frac{1}{N} \sum_{j=1}^{k} |C_j| \max_{i=1\ldots k} \text{prec}(i,j).
\]

And finally, the Entropy measure is calculated using the formula:

\[
\text{Entropy} = -\frac{1}{\log k} \sum_{j=1}^{k} \frac{|C_j|}{N} \sum_{i=1}^{l} \text{p}_{ij} \log \text{p}_{ij},
\]

where \( \text{p}_{ij} \) is the probability that a member of cluster \( C_j \) belongs to class \( C_i \).

IV. RESULTS

In order to test the effectiveness and efficiency of the phrase-based document similarity, here is conducted a series of experiments to compare the new document similarity with the traditional keyword tf-idf similarity measure in the same GHAC and STC algorithm. For fair comparison, first some standard document data sets without any bias must be provided, and then some standard clustering quality metrics shall be examined. In the comparison experiments, here are used eight data sets from two document collections.

The first four document collections are generated from OHSUMED cardiovascular datasets the next four document collections are from the 20-Newsgroups. The second document collection is a subset (mini_20newsgroup) of the well-known 20-newsgroups collection.

The code is in eclipse IDE and conducted all experiments on a PENTIUM IV 2.6 GHz using jar file. Since there is no binary or source code of the original STC algorithm available, here is use to implement the STC algorithm by following the description in Zamir's papers.
A. Document Pre-processing

Before the document clustering, a document “cleaning” procedure is executed for all documents in the data sets: First, all non-word tokens are stripped off. Second, the text is parsed into words. Third, all stop words are identified and removed. Fourth, the Porter’s suffix-stripping algorithm is used to stem the words. Finally, all stemmed words are concatenated into a new document.

Since the length of a word is variable, it is quite difficult to implement a suffix tree based on words directly. To solve the problem, here is build a wordlist to store all keywords in alphabetical order.

The similar ideas are often used in some text retrieval approaches for simplifying the computation complexity, such as the inverted index systems. In the wordlist, a unique integer number (called a word_id) is assigned to each keyword so that it can use the word_id to replace the corresponding word in the “cleaned” document. Finally, each document becomes an array of word_ids for the suffix tree construction.

Table 1 Overview of 8 different datasets (O-OHSUMED, N-20-Newsgroups)

<table>
<thead>
<tr>
<th>Document Set</th>
<th>DS1</th>
<th>DS2</th>
<th>DS3</th>
<th>DS4</th>
<th>DS5</th>
<th>DS6</th>
<th>DS7</th>
<th>DS8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corpus type</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Number of documents</td>
<td>270</td>
<td>500</td>
<td>300</td>
<td>700</td>
<td>1000</td>
<td>509</td>
<td>1080</td>
<td>1000</td>
</tr>
<tr>
<td>Number of unique words</td>
<td>4752</td>
<td>6870</td>
<td>6971</td>
<td>8942</td>
<td>10784</td>
<td>13984</td>
<td>11357</td>
<td>11109</td>
</tr>
<tr>
<td>Number of nodes for analysis</td>
<td>3315</td>
<td>5290</td>
<td>6345</td>
<td>7715</td>
<td>10157</td>
<td>11839</td>
<td>11018</td>
<td>9355</td>
</tr>
</tbody>
</table>

Figure 1. Overview of the 8 different dataset

The figure 1 shows the graph for the all eight different dataset. The red line in the graph shows the number of unique words in the document. The light green line shows the number of node for the analysis. These numbers are used for the next stage that for the finding the phrase cluster depending upon the scores. The highest scores document are taken for merging the cluster. The blue line shows the number of documents.

See the table 1 which gives the document count, number of unique words and the number of nodes for the analysis. From the table it shows that number of documents for the dataset 7 is greater than all
dataset. The number of unique words in the dataset 6 is higher than the all. The number of unique words in the dataset 1 is l.

B. Comparison of Four Clustering Algorithms

Document clustering has been investigated for a long time. A number of document clustering approaches have been developed based on different document models and various clustering algorithms for decades. In general, the approaches can be categorized into agglomerative solutions (e.g., hierarchical clustering) and partitional solutions (e.g., K-means clustering).

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 key of HAC algorithm is the method used to determine which pair of clusters will be the most similar pair for merging at each iteration. The building method is quite simple but needs to specify how to compute the similarity of two clusters. In this work, the group-average similarity is chosen for the HAC algorithm, which measures the similarity of two clusters with the average of pairwise similarities of the documents from each cluster.

K-NN algorithm is well known for text-based document classification. It has also been used for document clustering. For each document, the K-NN algorithm calculates its pairwise similarities to all other documents in the data set. The top K documents are selected as its nearest neighbors, and then the document is assigned to the cluster where the majority of the K documents belong to. Thus, the K-NN clustering algorithm uses the parameter K to refine the resolution of the clusters, in order to obtain a given bound m on the number of the clusters that are generated for representing the data set.

In this experiment, the performance of four clustering algorithms is evaluated in the comparison experiment: the original STC algorithm, the GHAC with the phrase based document similarity, the GHAC with the traditional single-word tf-idf cosine similarity, and the K-NN clustering algorithm with the phrase-based document similarity. The original STC algorithm selects the 500 highest scoring base clusters for further cluster merging, but only chooses the top 10 clusters as the final result of the clustering. However, this still record all clusters generated by the cluster merging of the STC algorithm, and compute the three kinds of measure scores for the clustering result as well as the other clustering algorithms. Figs. 2, 3, and 4, respectively, illustrate the F-measure, Purity, and Entropy scores computed from the clustering results of four clustering algorithms on the eight document data sets, where STC designates the results of all clusters generated by the STC algorithm, and STC-10 designates the results of the top 10 clusters selected by the original STC algorithm; GHAC designates the results of GHAC with the phrase-based document similarity; GHAC designates the results of the same GHAC with traditional keyword tfidf cosine document similarity; K-NN designates the results of K-NN clustering algorithm with the phrase-based document similarity. Since the STC algorithm allows a document to appear at more than one cluster, the Purity and Entropy measures are not suitable for evaluating such clustering. From these three graph. Here is conclude that the purity and F-measure for the eight different datasets for STC algorithms is greater than the GHAC (phrase), GHAC (word), K-NN (phrase). But in figure 4 the entropy score are only for the STC, STC-10, and K-NN here also the entropy for the STC is higher than the STC-10 and K-NN.
Figure 2. The F-measure scores for 8 datasets for four clustering algorithm

Figure 3. The Purity scores for 8 datasets for four clustering algorithm

From the above figure it is observed that the purity score for the STC algorithm is higher than the other three algorithms. The purity score for the dataset 6 is highest.

Figure 4. The entropy scores for 8 datasets for four clustering algorithm

From the above figure it is observed that the Entropy score for the STC algorithm is higher than the other three algorithms. The entropy score for the dataset 3 is highest. The entropy scores are for the only STC, STC-10, KNN and the entropy scores for the GHAC (phrase) and GHAC (words) are nil.
Figure 5. F-Measure, Purity, Entropy scores of various documents with threshold value 0.3 for the STC algorithm

From the fig 5 it is observed that the purity scores are higher than the entropy and f-measure. The red line shows the purity scores, the green line shows the entropy score and the blue line shows the f-measure scores.

The fig 6 shows the F-measure scores for the 8 different datasets for the threshold values 0.3 to 1.8. The F-measure values for the DS5 which is from the 20-Newsgroups is highest.

Figure 6. F-measure of various 8 different data with various threshold values for cosine, correlation, euclidean similarity in ms STC

V. REFERENCES


