Identifying Data Set Texture using Normalized Compression Distance

Shahany Habeeb¹, Syam Gopi²

#P. G Scholar & Computer Science & Engineering & Amal Jyothi College of Engineering Kanjirapally,
Erumely Rd,
Kovapally, Kerala, 686518

Abstract — Models that do not preserve text structure or that preserve text structure can be used for presenting text data sets. Here the main hypothesis is that depending on the nature of data set, there can be advantages of using a model that preserves text structure over one that does not, and vice versa. The key is to determine the best way of presenting a particular data set, based on the data set itself. Here different distortion techniques are analyzed on the bases of compression distances for identifying texture of data sets.

Keywords— Data compression, Word removal, Normalized compression distances.

I. INTRODUCTION

The application of cluster analysis to textual documents is termed as Text clustering or Document clustering. Initially, document clustering was tested for improving the precision in information retrieval systems. It is also used as an expert way of finding the nearest neighbors of a document. It has applications in areas like topic extraction, fast information filtering and automatic document organization. In natural language processing and information retrieval documents are mainly represented using bag-of-words model. This model represents a text (such as a document or a sentence) as bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity. This model is also known as Vector Space Model (VSM).

VSM has been successfully applied in many areas. However, representing text documents using this model makes sense from a point of view of computational efficiency; it can be imprecise because word order can be very important. Taking relationships between words, it is possible to extent bag-of-words model. Such relationships have been dealt with using ontologies, graphs, topic modeling and information extraction techniques.

It is possible to use compression distances for identifying relationships between words (text structure). With the help of data compression such distances give a measure of similarity between two objects. This means that they can give a similarity measure between two texts from texts themselves. Compression methods are intended for compressing natural language text and other data with a similar sequential structure. However, these methods can achieve some compression on almost any kind of data. Compression is useful because it helps to reduce resource usage, transmission capacity or data storage space. Text structure gives important information on a text. But unmodified texts contain words, that not relating to the matter at hand. This makes the comparison of the text difficult. In order to improve text comparisons, various distortion techniques are applied. This is mainly with the aim of helping the compressor to obtain more reliable similarities in a compression-based clustering scenario. The distortion techniques remove irrelevant information's while preserving both text structure and relevant information.

This work applies distortion techniques with a different goal, mainly used as a tool that allows us to discover the structural characteristics of data sets. That is to discover their texture. The analysis is carried out to discover data set nature can be divided into different parts. First, this work applies some preprocessing technique including stop word removal and distortions. For word removal two approaches are there. The first method uses a generic fixed stop-word list and the other in which this list is generated from the collection itself. The first approach is better in terms of maintaining the most relevant information of the documents. That is, the replaced words are not specific enough. The second method generates the stop-words list from the documents itself, obtaining a more aggressive word removal. Following preprocessing, compression similarity measures are analyzed on the bases of two different compression algorithms. Further analysis is carried out based on this NCD measures.

II. LITERATURE REVIEW

In previous works [1] mainly four distortion methods are used, which includes OO, RPA, RPE, and RPPW. Followed by this preprocessing compression algorithms are applied. In [3] LZMA compression algorithms are used. In order to obtain better results Bzip2 and PPMD compression algorithms are also used. The NCD measure obtained from compressors
are given as the input of compLearn toolkit. In the CompLearn Toolkit, the output of the clustering algorithm is represented as a dendrogram. The procedure carried out to measure the error of a dendrogram is as follows. First, the pairwise distances between the documents that should be clustered together are added. Second, after calculating this addition, the addition that corresponds to errorless clustering is subtracted from the total quantity obtained in the first step.

III. SYSTEM DESCRIPTION

A. Preprocessing

Several distortion methods that modify text structure from less to more are used in this work. After applying distortion technique, distortion is carried out, that is randomly permuting different parts of the distorted texts. One of the distortion method presented here is Original Order (OO) [5] distortion. Here no random permuting is carried out after replacing the words with asterisks. The names of the other distortion techniques are given below:

Randomly Permuting asterisks (PRA): After substituting the words using asterisks, the strings of asterisks are randomly sorted or permuted. Here the remaining words are maintained in their actual positions, while the removed words are not. This method is mainly to identify whether the structure of the contextual information is relevant or not. It is very important that each string of asterisks is treated as a whole. That is, if a word such as “cat” is replaced by *** these asterisks always remain together. This method is created to evaluate the importance of the structure of the replaced words.

Randomly Permuting Remaining Words (RPRW): After replacing the words using asterisks, the remaining words are randomly permuted. In this method the structure of the asterisks is maintained, while the structure of the remaining words is not. This method is created to evaluate the importance of the structure of the remaining words.

Randomly Permuting Everything (RPE): After replacing the words using asterisks, both remaining words and the strings of asterisks are randomly sorted. This method is to evaluate the effect of the loss of the structure in the clustering outputs. Here also the strings of asterisks are randomly sorted as a whole too. In original order distortion only asterisks replacement is there, no sorting is performed. Here the removed words are selected with the help of BNC. Here also the strings of asterisks are randomly permuted as a whole too. This method is similar to the bag-of-words model and it is created as a control experiment.

B. Compression Algorithms

Many compression techniques have been used since the emergence of data compression as a research area, from primitive procedures to sophisticated algorithms that achieve very high compression values. In this work, data compression is used as the tool, to measure similarity between two documents.

Similarity normally assumes that two objects \( x \) and \( y \) are similar if the blocks of \( x \) are in \( y \) and vice versa. If this happens then it to describe object \( x \) by making reference to the blocks belonging to \( y \), thus the description of \( x \) will be very simple using the description of \( y \). This is what a compressor does to code the concatenated \( xy \) combination: a search for information shared by both sequences in order to reduce the redundancy of the whole sequence. This was studied by giving rise to the concept of Normalized Compression Distance (NCD) [2], [4] mathematical formulation is as follows:

\[
NCD(x, y) = \frac{\max\{C(xy) - C(x), C(yx) - C(y)\}}{\max\{C(x), C(y)\}}
\]

Where \( C \) is a compression algorithm, \( C(x) \) is the size of the \( C \)-compressed version of \( x \), \( C(xy) \) is the \( C \) compressed size of the concatenation of \( x \) and \( y \), and so on. In practice, the NCD is a non-negative number \( 0 \leq r \leq 1 + \epsilon \) representing how different two objects are. Smaller numbers represent more similar objects. The \( \epsilon \) in the upper bound is as a result of imperfections in compression techniques.

BZIP2 is a block-sorting compressor developed by Julian Seward. It compresses data using Run Length Encoding (RLE), the Burrows-Wheeler Transform (BWT), the Move-To-Front (MTF) transform and Huffman coding. The BZIP2 algorithm reads the input stream block by block. Each block is compressed separately as one string. The length of the blocks is between 100 and 900 KB. The compressor uses the BWT to convert frequently-recurring character sequences into strings of similar letters, and then it applies Move-To-Front transform and Huffman coding. The important phase is the BWT. It permutes the order of the characters of the string being transformed with the aim of bringing repetitions of the characters closer. This is very useful for compression, since there are techniques such as MTF and RLE that work very well when the input string contains runs of repeated characters.

Statistical compressors are based on creating statistical models of the text. The model assigns probabilities to the input symbols, and then they are coded based on these probabilities. Static or dynamic models are there, depending on whether the probabilities are fixed or dynamic, the latter are more suitable because they adapt to the particularities of the
data presented in the file being compressed. One of the important statistical compressors is PPM algorithm; here the name stands for Prediction with Partial string Matching. PPM was developed by Clearly and Witten with extensions and an implementation by Moffat. It is a finite-context statistical modeling method that can be viewed as blending together several fixed-order context models to predict the next character present in the input sequence. Probabilities for each context in the model are calculated from frequency counts which are updated adaptively. PPM uses sophisticated data structures and it usually achieves the best performance of any real compressor although it is also usually the slowest and most memory intensive. Many variants of the PPM algorithm have been implemented: PPMA, PPMB, PPMP, PPMX, PPMZ, etc.

C. NCD-Based Clustering

In this work similarity between preprocessed contents are measured with the help of NCD [6] by applying various compression algorithms as explained above. If the analyzed data set includes 5 documents then 4x 4 matrix is formed, indicating similarity between all pairs of documents. Figure 1 shows an 8 x 8 distance matrix, which shows NCD values of 9 documents from IMDB data set. Position [0] [0] indicate NCD between document 1 and document 1. Similarly [0] [1] indicate NCD between document 1 and document 2.

In order to calculate errors we require actual and error pairwise distances. From the errorless dendrogram it is possible to calculate actual pairwise distance between nodes. The distances between two nodes are defined as the minimum number of internal nodes needed to go from one to the other. For example, in Figure 2 the distance between the nodes with label theMatrix.2 and theMatrix.3 would be one, since both nodes are connected to the same internal node. In order to calculate actual pairwise distance add all the pairwise distances between nodes starting with the same string, i.e., add all the pairwise distances between the documents that should be clustered together. For example, in Figure 2 there are three nodes whose label starts with theMatrix., they correspond to the three plot summary of famous movie series “The Matrix”. Therefore, add the distance between theMatrix.1 and theMatrix.2, between theMatrix.2 and theMatrix.3, and between theMatrix.1 and theMatrix.3. Repeat this procedure with all the nodes, thus obtain a certain total quantity. These values are actual pairwise distances of IMBD clusters.

In this work error pairwise distances are calculated from similarity matrix itself. For example: if we want to calculate error pair distance between document 1 and document 5, NCD value corresponding to [0] [4] (NCD between document 1 and document 5) is 0.906. For document 1 there exist 9 possible NCD values, all these values are compared with the NCD value corresponding to [0] [4], which we want to reach. If a particular NCD value is lesser than that of NCD of document which we want to reach then it is considered as a node to pass”. That is node to pass in order to reach from one document to another. Finally counts of all node to pass” are taken as the error pair distance between the documents. Here error pair distances between to calculate errors. First, the error pairwise distances between the documents that should be clustered together are added. Second, after calculating this addition, the addition corresponds to errorless clustering is subtracted from the total quantity obtained in the first step. The Table 1 shows actual pairwise distance obtained from errorless dendrogram and error pairwise distance calculated from the NCD matrix. Subtracting actual and error pair distance and summing those values give error value; here the error value is 38.

Errorless dendrogram for Book and UCI-KDD datasets are given in figure 3.3 and 3.4. The Silhouette coefficient gives a measure of the quality of the clusters obtained in a clustering scenario. For each datum i, the following measures are calculated:

- \(a(i)\): which is the average dissimilarity of \(i\) with all other data within the same cluster.
- \(b(i)\): which is the lowest average dissimilarity of \(i\) to any other cluster in which \(i\) is not a member.

\[
s(i) = \frac{b(i) - a(i)}{\max \{a(i), b(i)\}}
\]

The Silhouette coefficient is usually calculated using cosine or euclidean distances. However, as mentioned previously, in this work the application domain based on dendrograms. This has been done by considering the distance between two datum (dendrogram leaves) as the minimum number of internal nodes needed to go from one to the other. The Dendrogram Silhouette Coefficient (DSC) using the following formula:

\[
DSC = \frac{1}{N} \sum s(i)
\]
IV. EXPERIMENTAL DETAILS AND ANALYSIS

A. Data Set
Books dataset: Fourteen classical books from universal literature, to be clustered by author.

IV. EXPERIMENTS

A. Data sets
- UCI-KDD dataset: Sixteen messages from a newsgroup, to be clustered by topic.
- MedlinePlus dataset: Twelve documents from the MedlinePlus repository, to be clustered by topic.
- IMDB dataset: Fourteen plots of movies from the Internet Movie Data Base IMDB- to be clustered by saga.
- Books dataset: Fourteen classical books from universal literature are clustered.

B. Experimental Result

Interesting conclusions can be drawn from the experiments. First, losing the structure of texts makes the clustering results get worse as the amount of removed words increases. Second, losing the remaining words structure makes the clustering results get worse when the texts contain a lot of remaining words and a few of asterisks. Third, losing every structure, both behaviors are observed at the same time, that is the clustering errors are worse for small and large numbers of removed words.

For all the analyzed databases, the clustering error becomes worse when the structure of the document changes. From the experimental results it is clear that OOD produces lesser error when comparing it with other. In almost all cases this behavior is true.

Figure 3 shows experimental results of IMBD data set using BZIP2 compression algorithm. Results of IMBD data set using PPMD compression algorithm is shown in figure 4. Similar ranges of results are produced by MedlinePlus data set also. In all these graphs, error difference varies from 0.1 to 0.3 ranges. The remaining two data sets analyzed in this work are Book and UCI-KDD. Analyzing obtained result, error differences are always above 0.3. Nature or texture of
the data set is one of the very important factor that affect clustering results. In this work various distortion methods are applied over preprocessed data. This is mainly for identifying the texture of data sets. In each of these distortions the structure of the documents are changed. Changing the structure always produce errors in clustering.

**TABLE I**

<table>
<thead>
<tr>
<th>CLUSTER</th>
<th>NODES</th>
<th>ERROR DIST</th>
<th>ACTUAL DIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>IJ</td>
<td>IJ.RaidersOfTheLostArk, IJ.TempleOfTheDoom</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>IJ.RaidersOfTheLostArk, IJ.TheLastCrusade</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>IJ.TheLastCrusade, IJ.TempleOfTheDoom</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>PC</td>
<td>piratesOfTheCari..1, piratesOfTheCari..2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>piratesOfTheCari..2, piratesOfTheCari..3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>piratesOfTheCari..1, piratesOfTheCari..3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>ISSW</td>
<td>tarWars.4,starWars.5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>tarWars.5,starWars.6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>tarWars.4,starWars.6</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>TM</td>
<td>theMatrix.1,theMatrix.2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>theMatrix.1,theMatrix.2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>theMatrix.2,theMatrix.3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TMY</td>
<td>theMummy.2,theMummy.1</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

But it is more specific for some data sets, i.e more for more for structurally oriented data sets,

This occurs because the structure of the texts that comprise these data sets is highly representative to cluster them. Losing text structure does not affect the clustering behavior so strongly in the Medline and the IMDB data sets. This is due to the fact that the remaining words are the key factor to cluster the documents belonging to these data sets. When the NCD captures the structure of similar documents, text structure is very important. These are the Books and the UCI-KDD data sets. Fourteen classical books from universal literature is presented in Book data set. UCI-KDD represents sixteen messages from a newsgroup. Here the experiments show that the structures of the documents in these data sets are very important while considering clustering scenario.

Two different compression algorithms are used here. One is block based compression technique, BZIP2 compression and the other is PPMD compression. BZIP2 algorithm reads the input stream block by block and each block is compressed separately as one string. PPMD is a statistical compressor which is based on context. Considering the factor run time PPMD compression algorithm is much faster than that of BZIP2 compression. PPMD always produce NCD matrix with in short time even if there exist different frequency level. Figure 5 shows a comparison of execution time for both compressions.

**Fig.3:** IMDB with Bzip2 compression

**Fig.4:** IMDB with PPMD compression
V. CONCLUSION AND FUTURE WORKS

Discovering the nature of data sets can help us manage texts more efficiently. Thus, there can be data sets where representing texts applying a model that does not preserve text structure is preferable to applying a model that preserves it, and vice versa. This paper utilizes four distortion techniques with the purpose of finding out what the best way of managing a data set is. Changing the structure always produce errors in clustering. But it is more specific for some data sets, that is more for structurally oriented data sets. On the other hand if the data is not structurally oriented, errors will be there, but it always maintain small error range.

It is also possible to use other clustering algorithms such as K-means or SVM to test these approaches in larger data sets.

ACKNOWLEDGMENT

We thank computer science department of Amal Jyothi College of Engineering for providing us with relevant data. This work was supported as part of thesis project.

REFERENCES