SEMANTIC ANALYSIS BASED TEXT CLUSTERING BY THE FUSION OF BISECTING K-MEANS AND UPGMA ALGORITHM

G. Loshma¹ and Nagaratna P. Hedge²
¹Jawaharlal Nehru Technological University, Hyderabad, India
²Vasavi College of Engineering, Hyderabad, India
E-Mail: loshmagunisetti15@gmail.com

ABSTRACT

Owing to the fastest data growth, this era can be claimed as the era of zettabytes. An effective mechanism is the need of this hour, to manage all the available data efficiently. Clustering is a technique to group relevant documents together. This work takes the semantics into account and clusters the document with the hybrid of bisecting k-means and UPGMA algorithm. The semantic analysis is made possible by the inclusion of wordnet, which is a lexical database. The outcome of this algorithm is more accurate, as the clusters are meaningful. The performance of the proposed algorithm is evaluated with respect to precision, recall, F-measure, accuracy and misclassification rate. The experimental results of the proposed work are satisfactory.

Keywords: clustering, bisecting k-means, UPGMA, wordnet.

INTRODUCTION

The Internet handles several EBs (exabytes) of data every single month. Thus, this era can be claimed as the era of ZBs (zettabytes). A single EB and ZB are equivalent to 1000000 TBs (Terabytes) and 1000 exabytes, respectively. Thus, an effective mechanism is the need of this hour, to manage all the available data efficiently. Most of the present data are in textual format. Another problem to be addressed is the efficient data retrieval. The data retrieval speed is directly proportional to the growth of data. Delayed data retrieval is not tolerable in this fast paced world. Speedy data retrieval is possible only when the available data is organised properly.

Clustering is an efficient technique that aims at organising the similar data together in clusters. Thus, the required data can simply be retrieved from a vast collection of data in a shorter span of time. The main goal of a text clustering algorithm is to group relevant data together into clusters. The so formed clusters are easily controllable and results in faster data retrieval. An efficient clustering algorithm paves way for effective information retrieval with the least response time and thus improves the Quality of Service (QoS).

Several text clustering algorithms are presented in literature. However, most of the clustering algorithms do not deal with the semantic information of the text. For instance, Bag of Words (BoW) is the approach that does not consider the semantic relationship of the words. The cluster formation is more accurate when the clustering algorithm takes semantic relationship into account. This sort of clustering algorithm takes the textual meaning of the text into account, in order to cluster the text. The resultant clusters are meaningful because the clusters are formed on the basis of the literal meaning of the words [1-11]. Thus, the clustering efficiency can be improved.

This paper presents a semantic based clustering algorithm by incorporating wordnet and feature based semantic similarity measure. Wordnet is the most popular English thesaurus, which lists out the semantic relationship between terms. This work employs a feature based semantic similarity measure, which relies on the semantic relationship between the terms. Finally, the so formed clusters are labelled for easier recognition. Labelling is not a big deal in this work, as the semantic relationship between the terms is the central theme of this work.

The rest of the paper is systematized as follows. Section 2 presents the review of literature. The proposed algorithm is explained in the section 3. The performance of the proposed algorithm is analysed and the evaluation results are discussed in section 4. Finally, the concluding remarks are presented in section 5.

Review of literature

This section reviews the related clustering algorithms in the literature. Text clustering is the most researchable topic and several works are already present in the literature. This section aims at analysing the related works.

Wordnet

Wordnet is one of the largest thesauruses of English language. It connects all the terms to relevant terms, with respect to their meaning. It contains synonyms and the relationship of terms. Wordnet 2.1 consists of 1, 55,327 words in 1, 17,597 senses. Synset is a technical term of wordnet, which aggregates nouns, verbs, adjectives and adverbs to form synonym set. This lexical database is employed for text clustering applications to improve the accuracy based on semantics.
K-Means algorithm

K-means algorithm is the popular clustering algorithm, and this term was first introduced by James Macqueen in the year of 1967 [12]. The standard algorithm was presented by Stuart Lloyd in the year 1957. K-means algorithm is easy to implement and thus many clustering problems employed k-means algorithm.

This algorithm consumes minimal time for execution [13-16]. The pitfall of this algorithm is its dependency on cluster point [17-20]. The steps involved in k-means algorithm are presented below.

1. Choose k initial centre points;
2. Allocate all the points to the nearest centre point;
3. Recalculate the centre point of every cluster;
4. Repeat steps 2 and 3 until the centre point remains the same.

The initial centre points are needed to be chosen and then all the points are allotted to the nearest centre point. The centre point of all the clusters is calculated again and this step is repeated until there is no change in the centre points.

Bisecting k-means algorithm

Bisecting k-means algorithm is the improved version of k-means algorithm. This algorithm iterates by selecting a cluster and follows a principle to divide the cluster. This process gets over as soon as the required count of clusters is attained or when the whole hierarchical tree is formed. The standard bisecting k-means algorithm is presented below.

Input: Dataset, Iteration count (it), required clusters;
Output: K clusters
1. Decompose a random cluster;
2. Compute bi-clusters;
3. Repeat step 2 until it;
4. Consider one of the bi-cluster that generates a high quality cluster;
5. Repeat steps 1 to 4, until the required cluster count is reached.

The clustering quality of this algorithm depends on the selected stopping criteria. This can be achieved by splitting the largest cluster and then to bisect the cluster by taking the cluster centroid into account.

UPGMA algorithm

UPGMA algorithm is the Unweighted Pair Group Method with Arithmetic Mean algorithm, which follows the bottom-up approach. This algorithm tends to construct a dendrogram by clubbing the two nearer clusters. The clustering process is achieved by the exploitation of distance or similarity matrix.

Proposed approach

This section presents the proposed text clustering algorithm. The proposed text clustering algorithm has its underlying roots on feature based semantic similarity measure, which yields the fruits of accuracy. The entire work is compartmentalized into phases such as pre-processing; feature based semantic similarity, clustering operation and cluster labelling. All these phases are described as follows.

Pre-processing

Pre-processing is the most important step in any sort of application, as it makes the data more suitable for further process. In this work, this step expedites the clustering process and enhances the quality of the algorithm. This phase eliminates stop words and stems the terms available in the document and represents the text document in a suitable format.

The initial step aims at eliminating or removing the words which has no meaning on its own. For instance, conjunctions, prepositions, articles and pronouns are meaningless by themselves. These words are called as stop words and are removed from the documents, in order to save memory and time. The sample stop words are listed in Table-1.
Table-1. List of stop words.

<table>
<thead>
<tr>
<th>List of stop words</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>an</td>
<td>The</td>
</tr>
<tr>
<td>Above</td>
<td>Across</td>
<td>Afore</td>
</tr>
<tr>
<td>against</td>
<td>Along</td>
<td>Aside</td>
</tr>
<tr>
<td>Among</td>
<td>Except</td>
<td>Include</td>
</tr>
<tr>
<td>Out</td>
<td>Despite</td>
<td>During</td>
</tr>
<tr>
<td>Beyond</td>
<td>From</td>
<td>Into</td>
</tr>
<tr>
<td>To</td>
<td>Out</td>
<td>Until</td>
</tr>
<tr>
<td>With</td>
<td>Than</td>
<td>Through</td>
</tr>
<tr>
<td>And</td>
<td>But</td>
<td>Because</td>
</tr>
<tr>
<td>So</td>
<td>Or</td>
<td>Yet</td>
</tr>
<tr>
<td>You</td>
<td>My</td>
<td>Me</td>
</tr>
<tr>
<td>She</td>
<td>Who</td>
<td>Myself</td>
</tr>
</tbody>
</table>

The second step of term stemming is triggered after the completion of stop words removal. The task of stemming aims at clipping the words by removing ‘ing’, ‘ed’, ‘es’, ‘s’ and so on. This phase removes stop words and stems the available words, so as to save memory and execution time.

**Text document representation**

The text documents are represented in such a way that the documents are represented as vectors. Two documents are claimed to be similar if those documents have high degree of correlation between them. All the documents are organised as vectors in the vector space as matrix. The term weights of all documents are given by

\[ doc_i = wt_{t1}, wt_{t2}, \ldots wt_{th} \]  

(1)

Where \( doc_i \) is the specific document, \( wt_{ti} \) is the weight of first term in the \( i^{th} \) document, \( wt_{hi} \) is the weight of the \( h^{th} \) term in the \( i^{th} \) document.

\[ Vdoc_i = \{wt_{1,i}, wt_{2,i}, \ldots wt_{h,i}\} \]  

(2)

Equation 2 notifies the vector space model of the documents. \( wt_{1,i}, wt_{2,i} \) are the term weights of the documents and are computed by

\[ wt_{hi} = tf_h \times IDF \]  

(3)

\[ IDF = \log \left( \frac{Doc}{docf_h} \right) \]  

(4)

\( tf_h \) is the occurrence frequency of \( h \) in the \( i^{th} \) document, \( docf_h \) is the total count of documents that possesses the term \( h \). \( Doc \) is the total number of documents in the dataset. The weight of the document is fixed on the basis of the importance of term. However, the above equations from 1 to 4 focus on the occurrence frequency of the terms alone.

This work formulates the vector space model by taking the semantic of the term into account and is presented below.

**Semantic similarity**

The semantic similarity between terms is computed by the incorporation of wordnet [21-25]. Wordnet is a lexical database which accumulates the terms called as synsets. The semantical relationship between terms is calculated by taking the semantic correlation between the terms. Every word is checked for the semantic relationship of another word in wordnet.

Let \( a_{h1,h2} \) is the semantic relationship between two terms \( wd_1 \) and \( wd_2 \). In case, if \( wd_2 \) is present in the synset of \( wd_1 \), then \( a_{h1,h2} \) is set to 1; otherwise \( a_{h1,h2} \) is set to 0 and is represented in (5).

\[ \begin{aligned} 
wd_2 \in wd_1 & \quad a_{h1,h2} = 1 \\
wd_2 \notin wd_1 & \quad a_{h1,h2} = 0 
\end{aligned} \]  

(5)

The weight \( wd_{ij1} \) of term \( t_{i1} \) in document \( doc_x \) is given by (6).

\[ wd_{ij1} = wd_{ij1} + \sum_{h2 \neq h1} a_{h1,h2} \cdot wd_{ij2} \]  

(6)

By this way, the semantic relationship between every pair of terms is computed. This is followed by the computation of similarity measure. This work exploits the cosine similarity between the documents and is presented in (7, 8).
\[
\text{Sim}(\text{Doc}_a, \text{Doc}_b) = \text{cosine}(\text{Doc}_a, \text{Doc}_b) \tag{7}
\]

\[
\text{cosine}(\text{Doc}_a, \text{Doc}_b) = \frac{\sum_{i=1}^{n} w_{id_1i} w_{id_2i}}{\sqrt{\sum_{i=1}^{n} w_{id_1i}^2 \cdot \sum_{i=1}^{n} w_{id_2i}^2}} \tag{8}
\]

Thus, the semantic similarity between the terms and the documents are found out. This step is followed by the process of clustering.

**Clustering algorithm**

This work exploits the fusion of bisecting k-means algorithm and UPGMA algorithm. Thus, the proposed work inherits the merits of both the algorithms. Bisecting k-means algorithm follows top-down approach, whereas UPGMA algorithm utilizes the bottom-up approach. This paves way for the generation of refined clustering results. The algorithm is presented below.

1. Initialize the necessary attributes;
2. Pre-process the documents;
3. Create document clusters by bisecting k-means;
4. Compute centroids of the clusters;
5. Pass the computed cluster centroid to UPGMA algorithm;
6. Refine the cluster centroids;
7. Produce k count of clusters;

The outcome of the above presented algorithm is the refined clusters of documents. The related documents are clustered together, thus the documents present in a cluster are highly correlated. On the other hand, the documents in the different clusters show lesser degree of similarity. By this way, the text documents are clustered effectively.

**Cluster labelling**

In this step, the degree of recurrence of all terms in all documents of every cluster is found out. This is followed by arranging the terms in descending order with respect to the degree of recurrence. The term with greatest occurrence is picked up and the cluster is labelled with that term. This is given by

\[
\text{lb}l = \text{Clt}(\text{Doc}(\text{Term}(\text{Doc}_1, \text{Doc}_2, \text{Doc}_3, \ldots, \text{Doc}_n))) \tag{9}
\]

Where cluster is denoted by \(\text{Clt}\), \(\text{Doc}\) is the documents present in the cluster, \(\text{Term}\) is the terms present in the cluster, \(\text{DoR}\) is the degree of recurrence of all terms present in the document.

This is followed by the arrangement of terms in descending order with respect to the degree of recurrence in the entire cluster. The term which is ranked first is declared as the label for the corresponding cluster. The main objective of cluster labelling is to enhance the readability. Any user can come to a conclusion about the substance of the cluster, at a streak. Thus, the intention of this work to cluster the similar documents and to label the cluster with meaningful term is achieved, successfully.

**Experimental analysis**

This section evaluates the performance of the proposed algorithm in terms of precision rate, recall rate, F-measure, accuracy and misclassification rate. The proposed work is compared with the outcome of k-means, bisecting k-means and UPGMA algorithms. The proposed work which is based on semantic analysis proves accurate results.

The dataset being exploited for evaluating the performance of the proposed work is Reuters-21578 R8, which has got 8 classes [26]. On the whole, the dataset contains 7674 documents, which consists of 5485 training documents and 2189 testing documents.

**Precision rate**

Precision rate is the total number of documents whose actual label is \(x\), but misclassified with label \(y\).

\[
P_{\text{rate}} = \frac{\text{doc}_{xy}}{\text{doc}_y} \times 100 \tag{10}
\]

Where \(\text{doc}_{xy}\) is the total number of documents with actual label \(x\), but wrongly classified as \(y\). \(\text{doc}_y\) is the documents which as correctly labelled as \(y\). Thus, a clustering algorithm works well with greater precision rates.
From the experimental results, it is obvious that the proposed work shows greater precision rate with 98.5%. Thus, the documents are clustered in a better way.

**Recall rate**

Recall rate is the total number of documents whose actual label is $x$, but misclassified with label $y$.

$$R_{rate} = \frac{doc_{xy}}{doc_x} \times 100$$

(11)

Where $doc_{xy}$ is the total number of documents with actual label $x$, but wrongly classified as $y$. $doc_x$ is the documents which as correctly labelled as $x$. Thus, a clustering algorithm works well with greater recall rates.

The experimental results show that the recall rate of the proposed work is 99.2%, which is comparatively greater than other algorithms.

**F-measure**

F-measure is computed by taking precision and recall rate into account. F-measure of a cluster and a class is given by

$$F(cls, cltr) = \frac{2 \times p_{rate} \times r_{rate}}{p_{rate} + r_{rate}} \times 100$$

(12)
Accuracy rate

The accuracy rate of the algorithm is determined by the sum of correctly clustered documents and the correctly rejected documents (as they are not relevant) to the total number of clustered documents.

\[ acc = \frac{ccd + crd}{total\ clustered\ documents} \]  \hspace{1cm} (13)

Misclassification rate

Misclassification rate is the rate of wrong clustering of documents. The misclassification rate must relatively be low and is calculated by

\[ mis_{rate} = 1 - acc \]  \hspace{1cm} (14)

Thus, the misclassification rate of the proposed work is the least, which when compared with all the other algorithms. Thus, the power of fusion of bisecting k-means and UPGMA algorithm along with semantic analysis is proven by the experimental results.

CONCLUSIONS

This paper presents a semantic approach based hybrid clustering algorithm. The main goal of clustering is to group related documents together in a cluster. This work employs wordnet to perform semantic analysis. The
cluster formation is accurate, as the clusters are framed semantically. Bisecting k-means and UPGMA algorithms are employed for the process of clustering. Finally, the clusters are labelled on the basis of term weight. The performance of the proposed algorithm is evaluated and compared with several existing algorithms. The experimental results prove the efficiency of the proposed algorithm.

REFERENCES


the association for computational linguistics. pp. 268-275.


