What is Text Clustering?

What is Text Clustering?

Clustering methods are unsupervised algorithms that help in summarizing information from large text data by creating different clusters. This method is useful in terms of understanding what your dataset is mainly about and in what different categories, you can divide the context of the text located in the dataset.
Before I discuss clustering algorithms in detail, later on. Let us first understand the algorithm working and the findings that are required to feed the models later on.
In this article, I will be going to discuss the theoretical concepts of the K-Means algorithm and the practical implementation of the K-Means algorithm. As we dive more into the article, you will see some of the experiments that I have done just to improve the accuracy of the models. In the end, you will see the conclusions drawn after implementing this algorithm on my dataset and its overall performance in summarizing the dataset to the correct extent.
K-Means Clustering:
K-means clustering is a type of unsupervised learning method, which is used when we don't have labeled data as in our case, we have unlabeled data (means, without defined categories or groups). The goal of this algorithm is to find groups in the data, whereas the no. of groups is represented by the variable K. The data have been clustered on the basis of high similarity points together and low similarity points in the separate clusters.
Before we dive deeper into feeding the dataset to the K-Means algorithm, we have to first prepare the dataset.
The Practical Implementation:
Step-1: The first thing is to read the file and return the list using the below-provided function.

Step-2: Reading N-Grams:
The second step is to read the N-Grams that we have generated in the previous step of Collocations:

- After looking at the top 100 results produced in Collocation's step, I concluded that frequency, t-test & likelihood ratio test performs well after filtering & gives mostly similar results.
- Whereas, PMI & Chi-sq test gave similar & good results without even applying a filter. But I still applied a filter on both of the methods. Although these methods are giving good results but still considering some "let us" "last updated" like occurrences as meaningful.

- Applying filter might haven’t deleted such senseless occurrences, but yeah it has reduced their preference in the list.
- I have visualized each of the filtered methods in the previous article, and I get to see other than frequency, all the methods are giving meaningful & similar clustering results.
- The list of generated n-grams is mostly similar because only the order of preferences is changing in each method.
- So, Here I have used filtered likelihood bigrams & trigrams list which I have generated in the previous article.

The below function is written to read n-Grams (Bigrams &Trigrams) and then return the combined list of bigrams & trigrams.

Step-3: After reading nGrams, the next step is to split the list of n-Grams into tokens to use further.

Step-4: Functions to get n-Grams vectors.
This step is crucial as we have to create n-gram vectors so as to feed the model later on.

Step-5: Read glove vectors from the text file.

I have used pre-trained domain-based embeddings to generate words for vector mapping. You can use your domain-based embeddings if there is any otherwise you can use some other popular embeddings out there like Stanford's Glove, Google's Word2Vec, Elmo, and Flair.

The next step is to create a word to vector feature array using the below lines of code.

Create a new data frame with unique words and their frequency occurrences in the documents listed under the corpus.

At this point, our dataset is all prepared to execute further and to be able to feed to the K-Means algorithm. But before feeding this dataset to the model, we have to also take care of some limitations of the K-means algorithm which may result in poor accuracy of the model.
K-Means Limitations:

1. K-means Clustering algorithm is an unsupervised learning method that requires lots of fine-tuning and one should keep in mind its limitations and drawbacks.
2. As per my analysis, it doesn’t work well with small size datasets.
3. It doesn’t perform well on datasets which has uniform Distribution.

4. It needs to find some optimal value of k using some external methods before feeding it to the algorithm's parameter.

Improvements made to improve the accuracy of the model:

1. Take the Transpose:

- I have taken the transpose of my dataset and have applied further steps on the transpose because it is giving somewhat good separable clusters.

2. Scale the Dataset:

- Since we will be working on an unsupervised learning model and it works badly on low data.
- Hence, there is a need to scale the data before feeding it to the k-means algorithm.

3. Standardize the Dataset:

- In order to ensure internal consistency of the data means each data type will have the same content and format.
- Standardized values are useful for tracking data that isn't easy to compare otherwise.

4. Check uniformity of the Dataset:

- I have checked the uniformity of both datasets using the KL divergence test.
- I have checked the KL test just for having an idea of distributions, we are not comparing distributions here.
- The KL divergence is zero if two distributions are equal.
- The KL divergence is positive if two distributions are different.

To find the KL divergence of two datasets, the below function has been used.

The below code will show the visual difference between the distribution of data frame and normal distribution.

Comparison between Data Frame & Normal Distribution:

- I have got the exact similar distribution as above for Transpose and Normal Distribution with KL divergence score = 138.635.

Conclusion of applying Uniformity Distribution:

- I have first compared plots of each dataset with that of normal distribution because I was getting bell-shaped distributions for each dataset.
- From the above graph, one can easily conclude that the dataset is not distributed uniformly since we are getting some bell-shaped curves.
- So, we can apply the K-Means algorithm easily.

Once we have checked all the conditions and made the required improvements in the limitations whatsoever, we can move further to apply the K-Means algorithm on our prepared dataset.
Perform Clustering:
I have used the K-Means algorithm here to generate clusters.

1. K-Means Clustering
   K-means clustering is a type of unsupervised learning method, which is used when we don't have labeled data as in our case, we have unlabeled data (means, without defined categories or groups). The goal of this algorithm is to find groups in the data, whereas the no. of groups is represented by the variable K. The data have been clustered on the basis of high similarity points together and low similarity points in the separate clusters.

The practical working of this algorithm is such as it computes the centroids and iterates until it finds the optimal centroid. Further, the data points are assigned in such a manner that the sum of the squared distance between the data points and centroid would be the minimum.
Since K is the parameter that has to be defined at first, we need to compute the optimal value of the K.

To compute the optimal value of K, we have used the Elbow Method.
Elbow Method:
To select the optimal no. of clusters or value of parameter K, perform the below-provided function.

Conclusion after applying Elbow Method:

- As shown in the above figure, the knee of an elbow curve signifies the cut-off point to be considered as optimal no. of clusters to be chosen.
- So, the optimal no. of clusters is 2.

After performing K-means Clustering:
The Results:

- As the figure itself is evidentiary that K-means clustering results in outliers.
- According to the figure above, there can only be two separate clusters that can be obtained from the dataset provided.
- K-means Clustering is actually sensitive to outliers because the mean can easily be influenced by outliers.
- In order to improve the results of clustering, I have fine-tuned the parameters of the K-Means algorithm and now results are somewhat enhanced. See the below image for reference.

- There is a total of 4 separate clusters that can be obtained from the dataset provided.
- Now let us see the top feature clusters that are obtained.

The conclusion drawn on my dataset and further improvement:

- To check my dataset's distribution, I have applied the KL divergence method and it was not uniform. So, k-means can be applied.
- I found optimal values of k using the Elbow method.
- For better results, Scale the dataset and standardize it before feeding it to k-means.

Here is the complete code that you can refer to for a better understanding.
I hope this article would have solved your queries.