DTI 5125 Data Science Application

Assignment 2

Group 7:

Haowei He (300158003)
Thomas Courtney (8335223)
Justin Dalrymple(300100350)

1. Introduction

The assignment is to apply text clustering to 5 digital books downloaded from Gutenberg, which belong to 5 different genres and authors. With the different genres of each book and the different writing styles of each author, our purpose is to find the best clustering algorithm that can identify the true author labels through content partitions and analyze the performance of each clustering algorithm.

2. Data Section

2.1 Data Description & Feature Engineering

Create 200 partitions of each book and each partition has 150 words after data cleaning. There is no splitting for training or testing sets, instead the entire dataset is used for clustering models.

	booktext_words	booktext	authors	target
0	[storm, speak, sister, sister, troubl, actual,...	storm speak sister sister troubl actual withou...	a	0
1	[close, crape, courag, direct, consequ, behind...	close crape courag direct consequ behind match...	a	0
2	[write, charl, repli, hall, alacr, moment, eng...	write charl repli hall alacr moment engag plac...	a	0
3	[counti, find, boast, ann, excus, mari, sinc, ...	counti find boast ann excus mari sinc upon i d...	a	0
4	[attempt, fellow, miss, ill, scrupl, wentworth...	attempt fellow miss ill scrupl wentworth she o...	a	0
...	...	...	...	...
995	[first, could, govern, everyth, made, one, dep...	first could govern everyth made one deprec lea...	e	4
996	[fli, put, consider, street, syme, smile, brok...	fli put consider street syme smile broken i in...	e	4
997	[gentleman, shudder, vast, pocket, marqui, lik...	gentleman shudder vast pocket marqui like pock...	e	4
998	[pool, would, abrupt, you, exil, along, we, ma...	pool would abrupt you exil along we man fieri ...	e	4
999	[shake, leav, dark, old, mob, gentleman, comed...	shake leav dark old mob gentleman comedi europ...	e	4

1000 rows × 4 columns

Table 1 - Dataset After Cleaning and Processing

According to the requirement of input data in the clustering model, we need to convert textual data to numeric. Therefore, three feature engineering models were chosen for training the models which are Bag of words, TFidf, and LDA.
For the LDA linear discriminant method, we implement a second transform based on BOW. Then we also do the second transformation and apply the clustering methods for further analysis on these models.

3. Clustering methods

3.1 K-means

Before applying clustering, we first define to normalize the methods and then we determine the optimal clusters from the three models(Fig 1.1/1.2/1.3). The goal of K-Means is to check for similar data and cluster them together while trying to separate each cluster apart as far as possible.

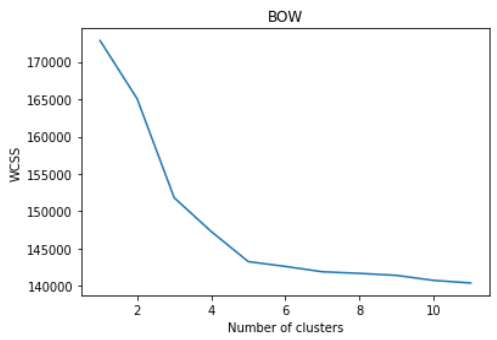

Fig 1.1 K-means clustering (BOW)

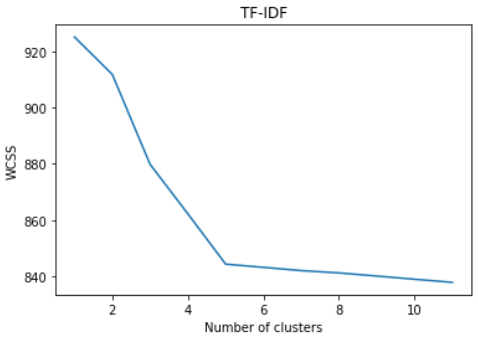

Fig 1.2 K-means clustering(TF-IDF)

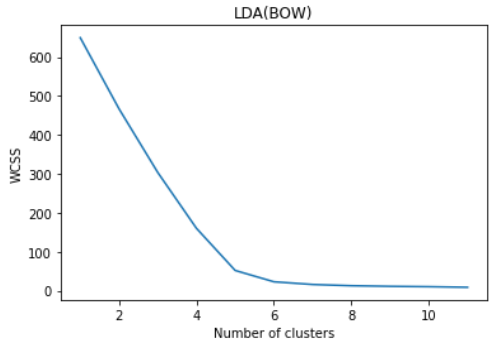

Fig 1.3 K-means clustering (LDA)

As seen in the graphs, the optimal number of clusters is five. This naturally matches the number of texts and authors selected. This also holds true for all feature models.Then implement the elbow test

to evaluate the outcomes by calculating Kappa and silhouette, and do the adjusted rand score cover one of those two.(Fig 1,4)

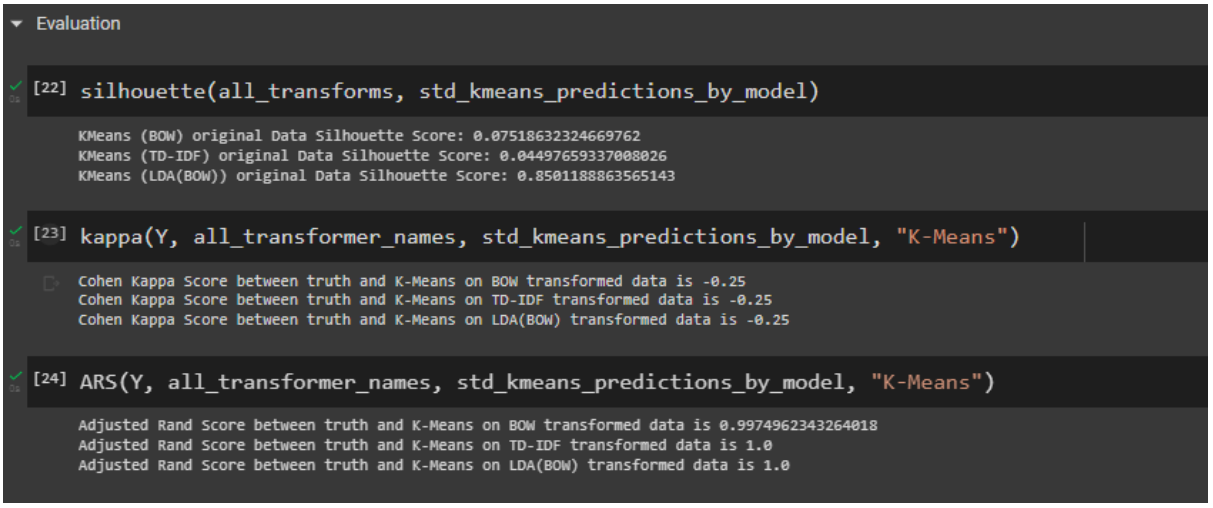

Fig 1.4 K-means clustering Evaluations

We can figure out from the outcome that LDA and TF-IDF have similar scores and LDA is slightly better.

We then use scaled features to K-Means. The goal of scaled features with K-Means is to check where the knee lands at while trying to separate each cluster apart as far as possible. Scaling using this technique doesn't prove to increase the clustering abilities. This is most likely due to the nature of the data itself where scaling isn't as effective (Fig 1.5/1.6/1.7)

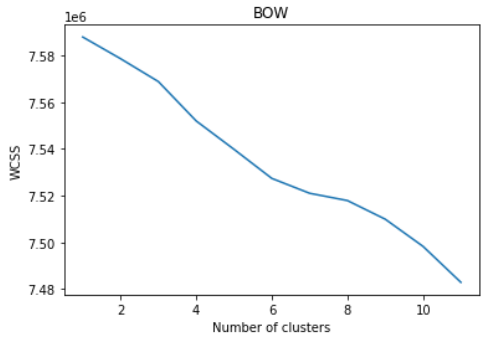

Fig 1.5 Scaled Features (BOW)

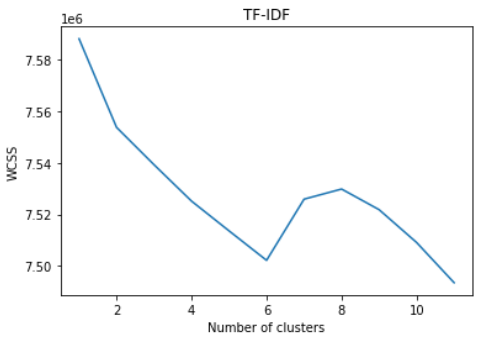

Fig 1.6 Scaled Features (TF-IDF)

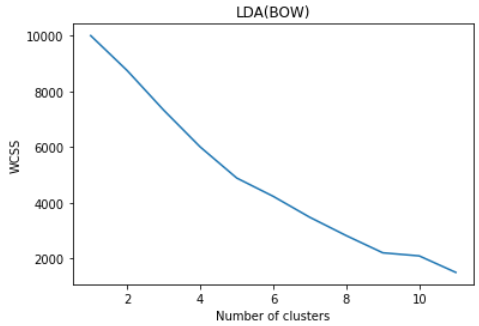

Fig 1.7 Scaled Features (LDA)

Evaluate the outcome again(Fig 1.8)

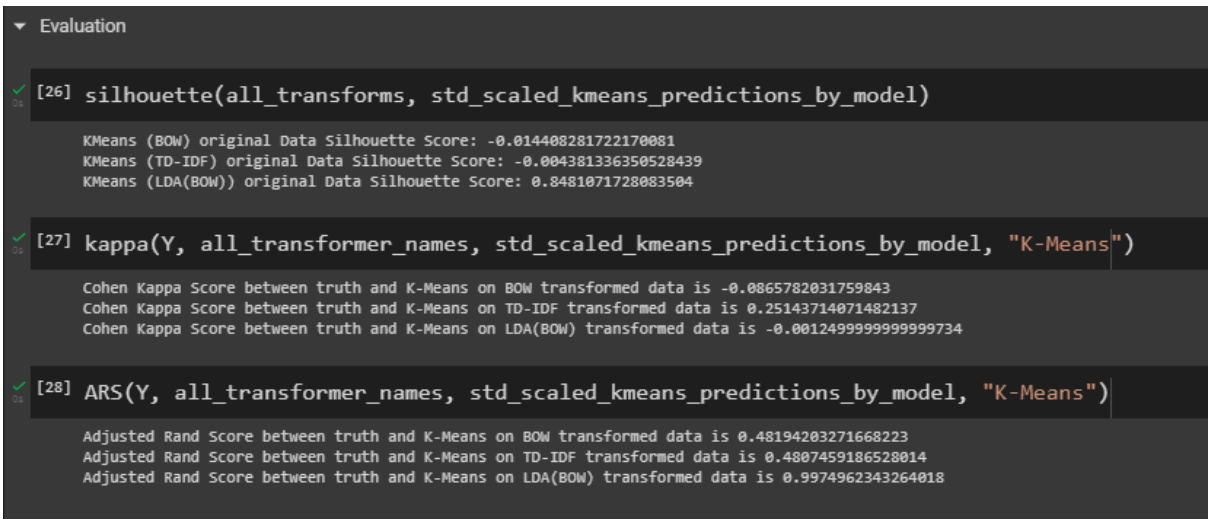

Fig 1.8 Scaled Feature K-Means Evaluation

From these evaluations, LDA has the optimal outcome and it is much better than TF-IDF and BOW. And BOW as well has the worst scores.

Next we apply SVD dimensionality reduction with n_components = 4 to view the optimal features number(Fig 1.8/1.9/2.0)

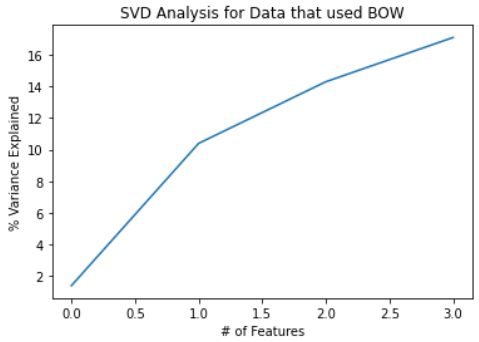

Fig 1.9 SVD K-Means (BOW)

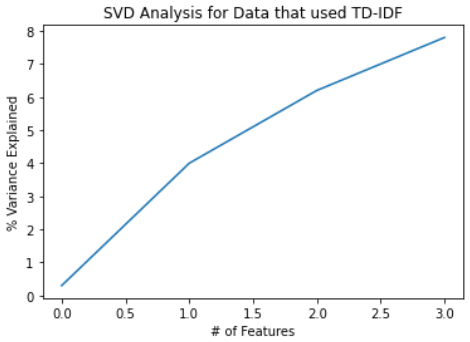

Fig 1.10 SVD K-Means (TF-IDF)

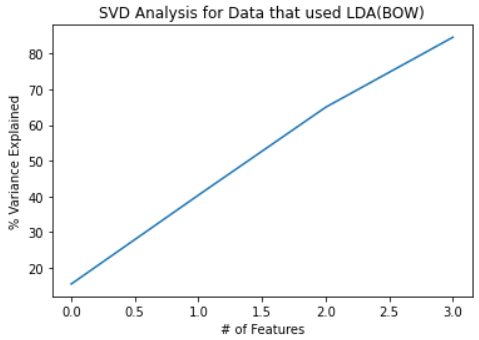

Fig 1.11 SVD K-Means (LDA)

The features outlined by the reduction mechanism make sense, since we are only looking at one feature. Here, the feature reduction to 2 has resulted in 3 clusters which is better than the original method(Fig 1.12/1.13/1.14)

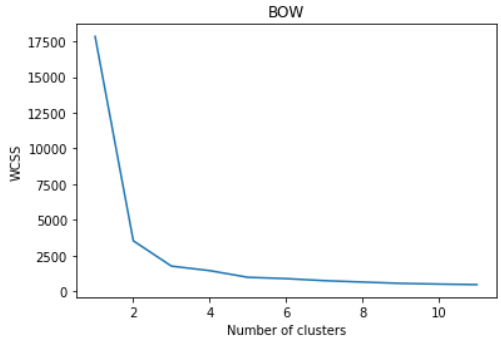

Fig 1.12 Reduction Outcome K-Means (BOW)

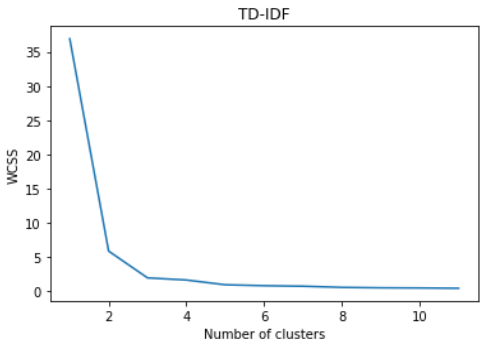

Fig 1.13 Reduction Outcome K-Means (TF-IDF)

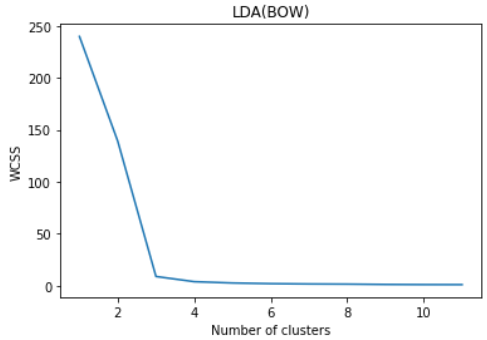

Fig 1.14 Reduction Outcome K-Means (LDA)

Again, we evaluate the outcome(Fig 1.15)

```
Evaluation

silhouette(all_transforms, std_svd_kmeans_predictions_by_model)

KMeans (BOW) original Data Silhouette Score: 0.061171761337624586
KMeans (TD-IDF) original Data Silhouette Score: 0.03320426832587965
KMeans (LDA(BOW)) original Data Silhouette Score: 0.5434361187362585

[34] kappa(Y, all_transformer_names, std_svd_kmeans_predictions_by_model, "K-Means")

Cohen Kappa Score between truth and K-Means on BOW transformed data is 0.11499999999999999
Cohen Kappa Score between truth and K-Means on TD-IDF transformed data is 0.25
Cohen Kappa Score between truth and K-Means on LDA(BOW) transformed data is -0.25

ARS(Y, all_transformer_names, std_svd_kmeans_predictions_by_model, "K-Means")

Adjusted Rand Score between truth and K-Means on BOW transformed data is 0.4435762155574122
Adjusted Rand Score between truth and K-Means on TD-IDF transformed data is 0.47308599109931404
Adjusted Rand Score between truth and K-Means on LDA(BOW) transformed data is 0.4817568735950199
```

Fig 1.15 Reduction K-Means Evaluations (LDA)

TF-IDF has the highest kappa score while LDA has the highest silhouette and ARS scores.

To have a better view of the method, try another reduction method(TSNE).In TSNE, each data point is mapped to the corresponding probability distribution by mapping transformation. Specifically, in high-dimensional space, Gaussian distribution is used to convert distance into probability distribution, and in low-dimensional space, long tail distribution is used to convert distance into probability distribution, so that the middle and low distances in high-dimensional space can have a larger distance after mapping, which can avoid paying too much attention to local features and ignoring global features during dimension reduction.(Fig 1.16/1.17/1.19)

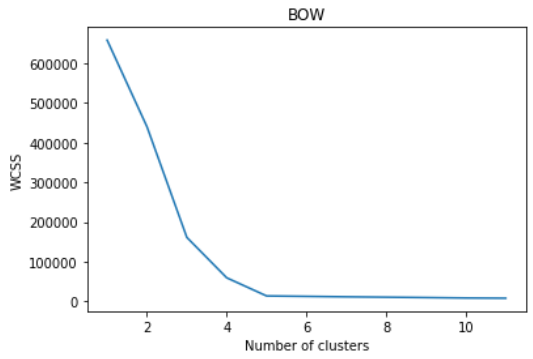

Fig 1.16 TSNE K-Means (BOW)

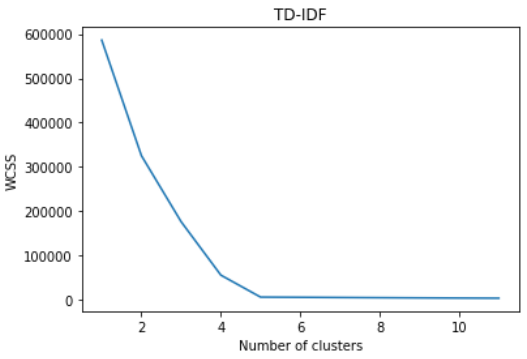

Fig 1.17 TSNE K-Means (TF-IDF)

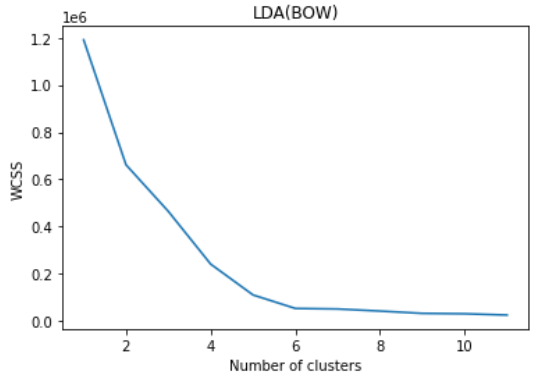

Fig 1.18 TSNE K-Means (LDA)

Interestingly, in this case, though the knees report generally 5 clusters, the BOW model reports the upper knee at 4. Evaluate again(Fig 1.19)

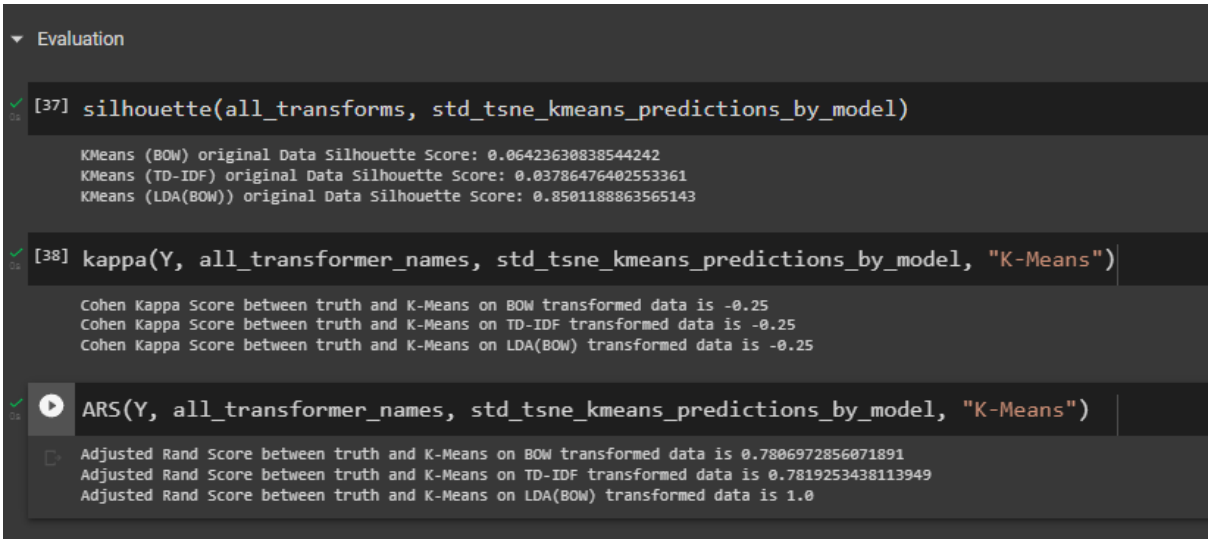

Fig 1.19 TSNE K-Means Evaluations

The outcome shows that LDA is the best while BOW is the worst.

3.1.1 Error Analysis

After all, perform error-analysis by using the top 10 frequent words for standard K-Means and top 5 for std K-Means. KMeans clustering with LDA has the highest silhouette score and ARS score. Thus, we want to focus on KMeans clustering to identify what are the characteristics of the instance records that generate different scores in evaluation after applying different feature engineering methods. Here we plot the top 10 words in 5 clusters for KMeans using BoW and TFiDF respectively.

Fig 1.20 to Fig 1.29 show the top 10 frequent words for standard K-Means.

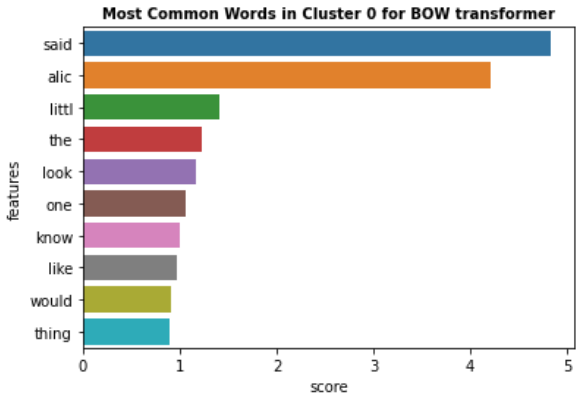

Fig 1.20

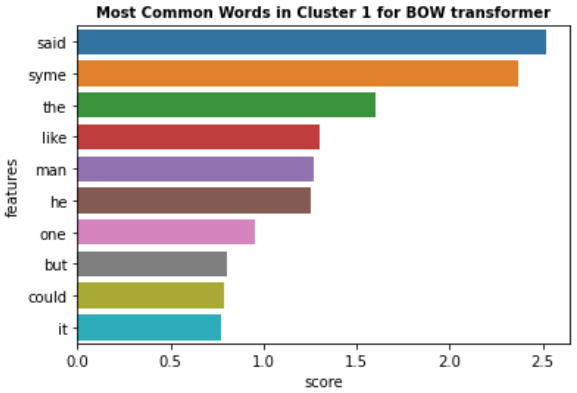

Fig 1.21

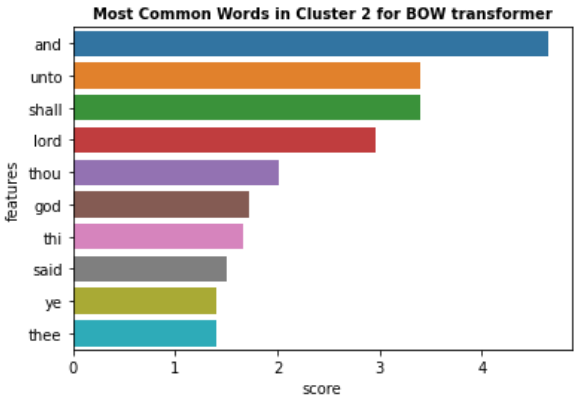

Fig 1.22

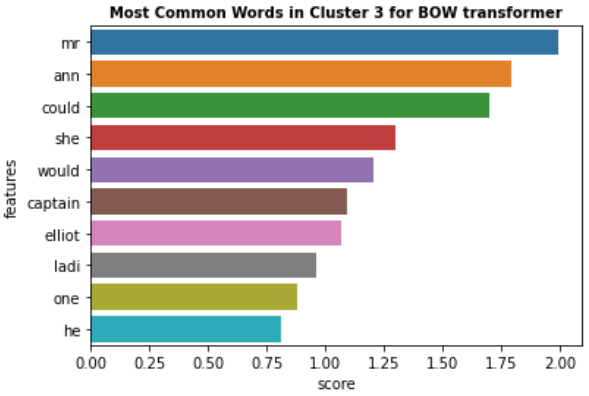

Fig 1.23

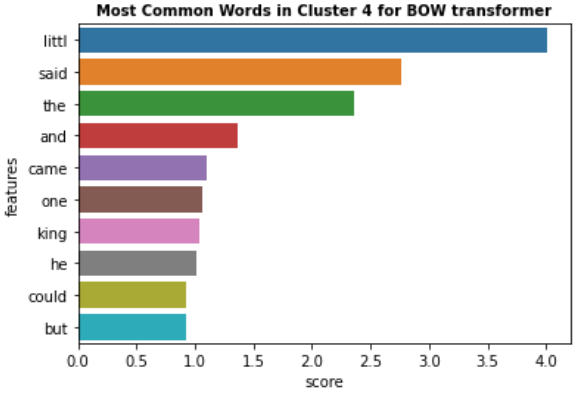

Fig 1.24

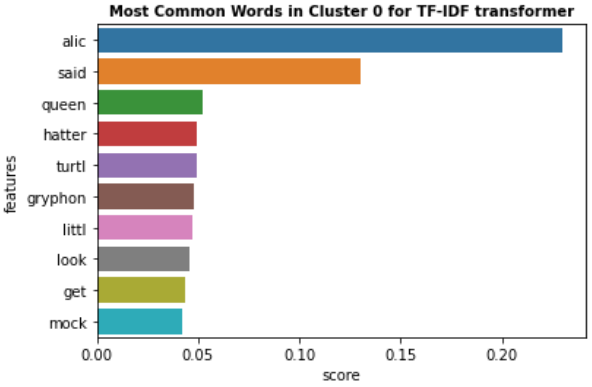

Fig 1.25

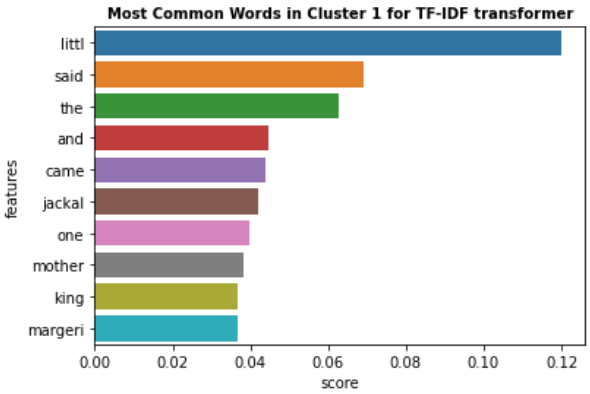

Fig 1.26

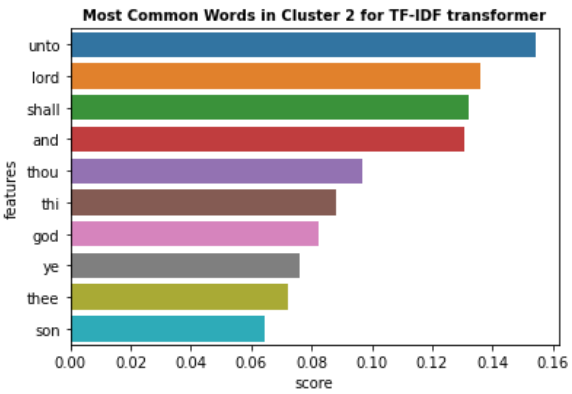

Fig 1.27

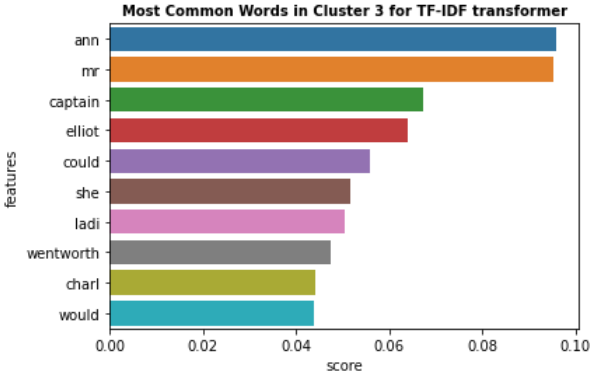

Fig 1.28

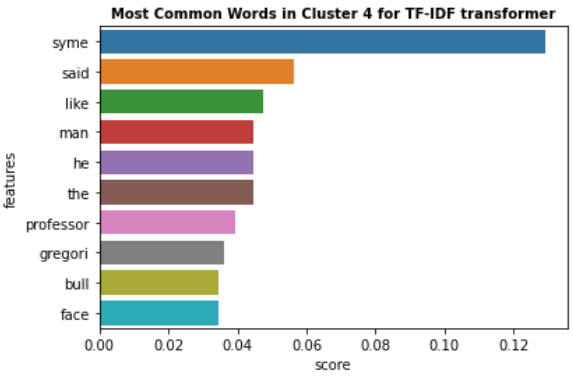

Fig 1.29

Fig 1.30 to Fig 1.39 show the top 5 frequent words for standard K-Means

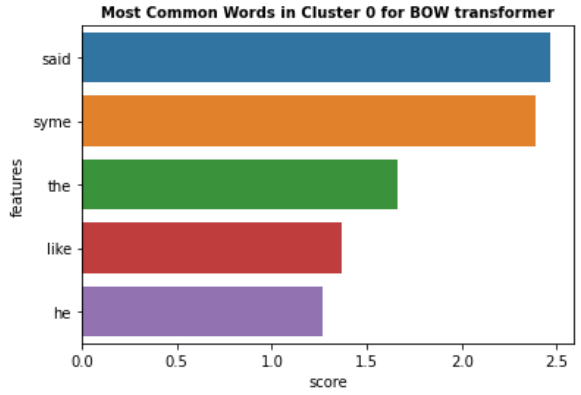

Fig 1.30

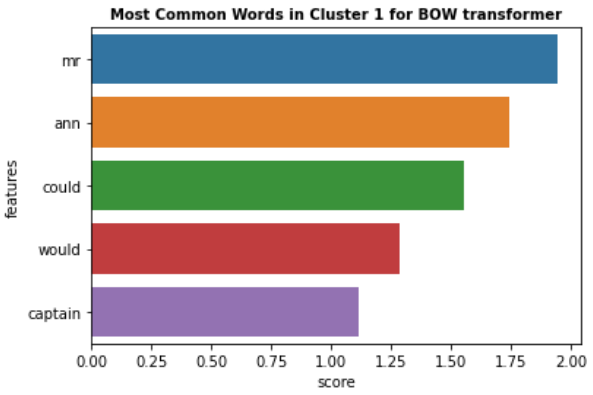

Fig 1.31

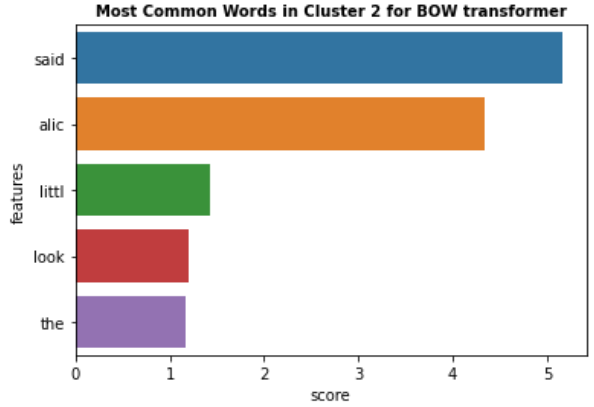

Fig 1.32

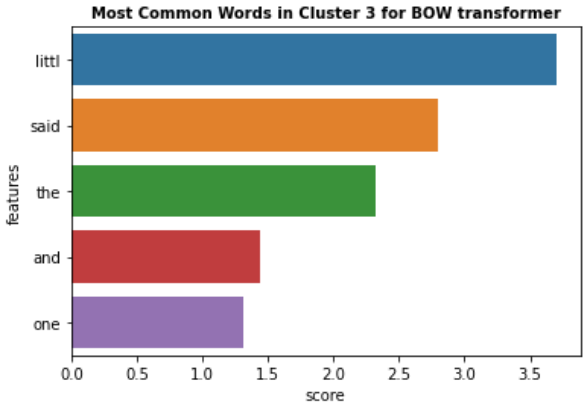

Fig 1.33

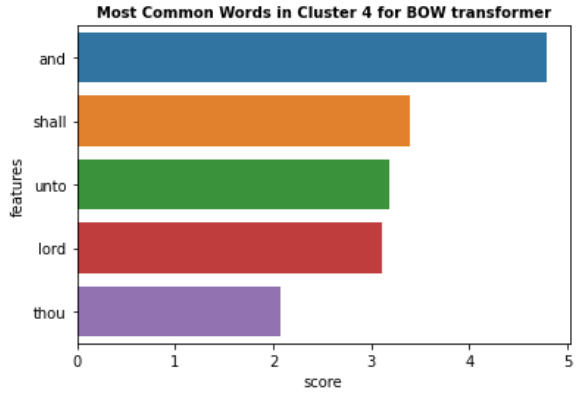

Fig 1.34

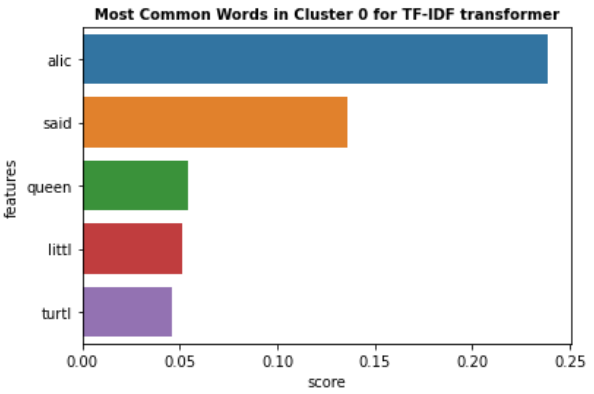

Fig 1.35

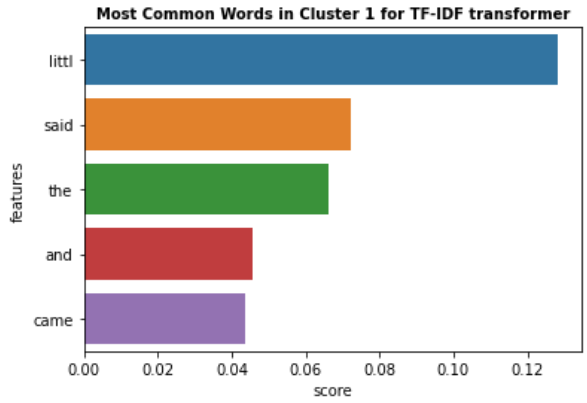

Fig 1.36

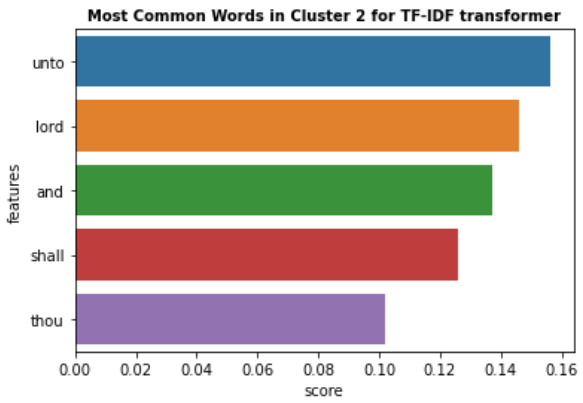

Fig 1.37

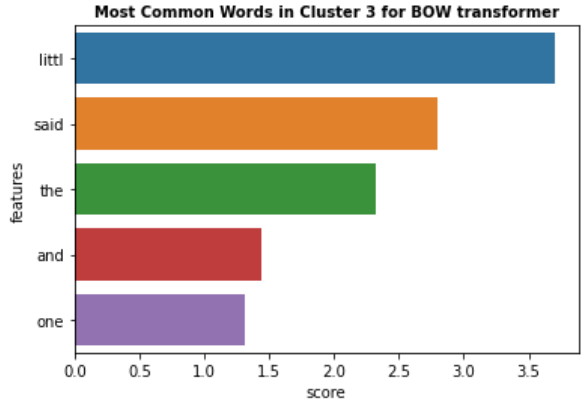

Fig 1.38

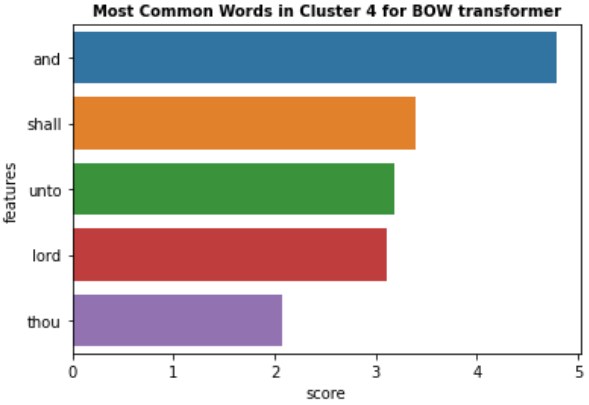

Fig 1.39

3.1.2 K-means Conclusion

The K-means clustering algorithm is relatively simple and highly efficient. It's also very flexible because it can easily be adjusted with different reduction methods. However, the optimal number of clusters must be decided before executing the analysis. When we are using the K-means algorithm, it will randomly select the starting point to develop a cluster so the result can vary even if we used the same data and the same function which also influences the final result.

3.2 EM Clustering

3.2.1 BIC & AIC

Similar to K-Means, we first use the BIC & AIC method to identify the number of clusters and both BIC & AIC are meant to be minimized in the chart. Ideally, the two measures will usually pick the same number for clusters, but in case they reflect different trends then BIC more favors simple models than AIC since the penalty term is larger in BIC than in AIC. For now we have already known there are 5 true labels, thus such methods of BIC and AIC are just for reference and analysis.

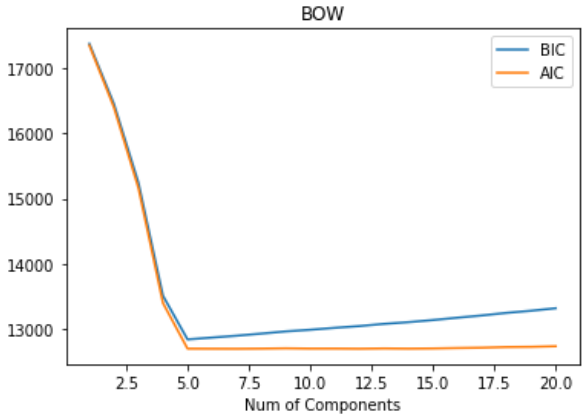

Fig. 2.1 BIC & AIC (BOW)

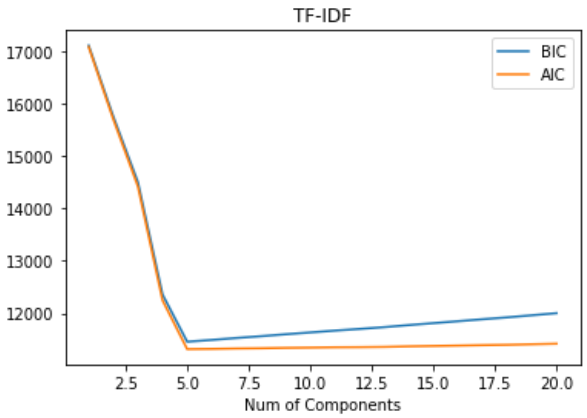

Fig. 2.2 BIC & AIC (TFiDF)

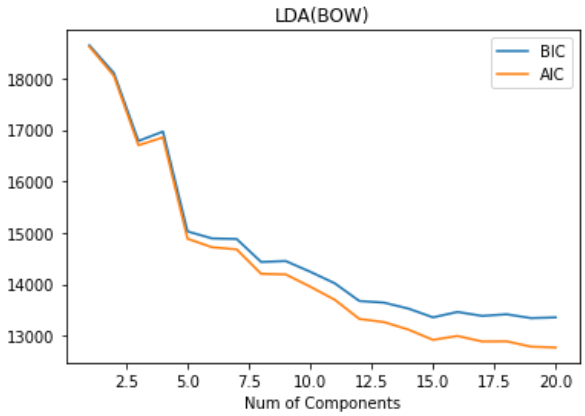

Fig. 2.3 BIC & AIC (LDA)

As seen in the graphs, the optimal number of clusters is also five. Then we process analysis on standard data and TSNE data.

3.2.2 Evaluations

Evaluate the standard data, the scores are shown below(Fig 2.4)

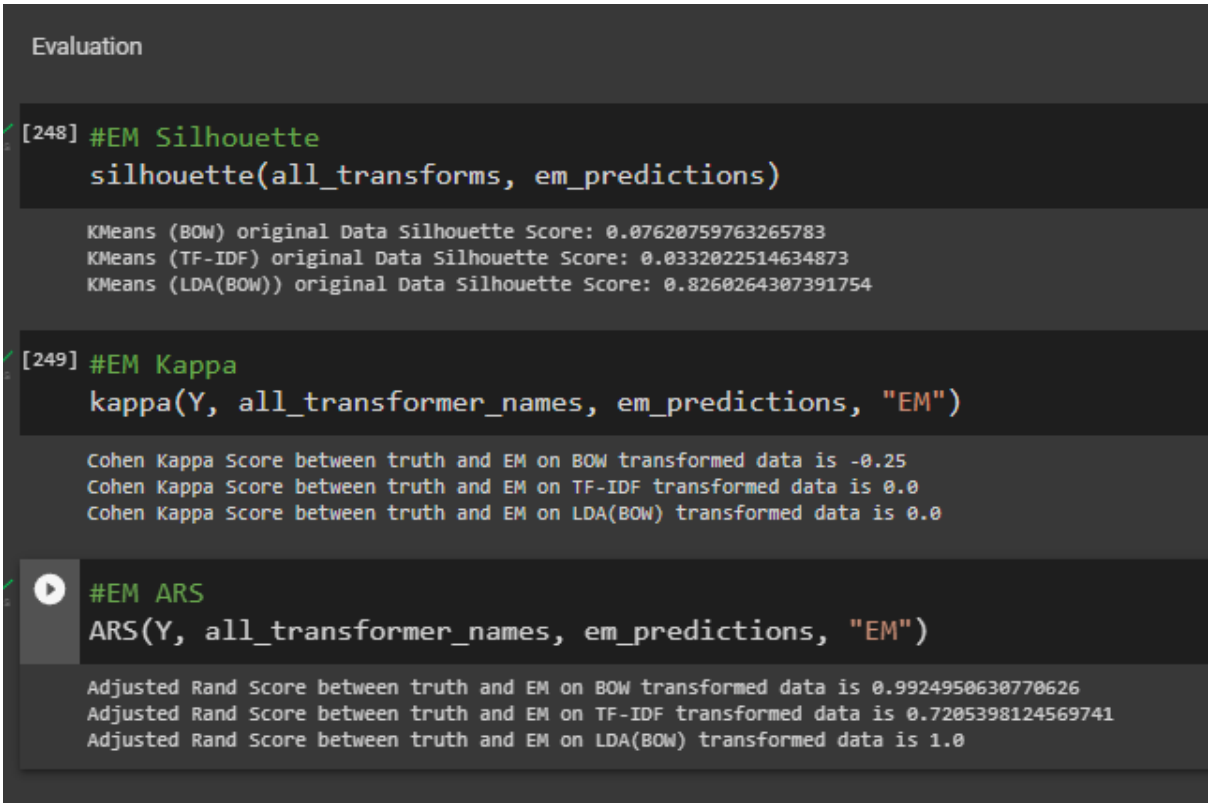

Fig. 2.4 Evaluations std-data

The evaluations show that LDA is the best model and the BOW is the worst.
Then perform the evaluations on TSNE data again and see how it goes(Fig 2.5)

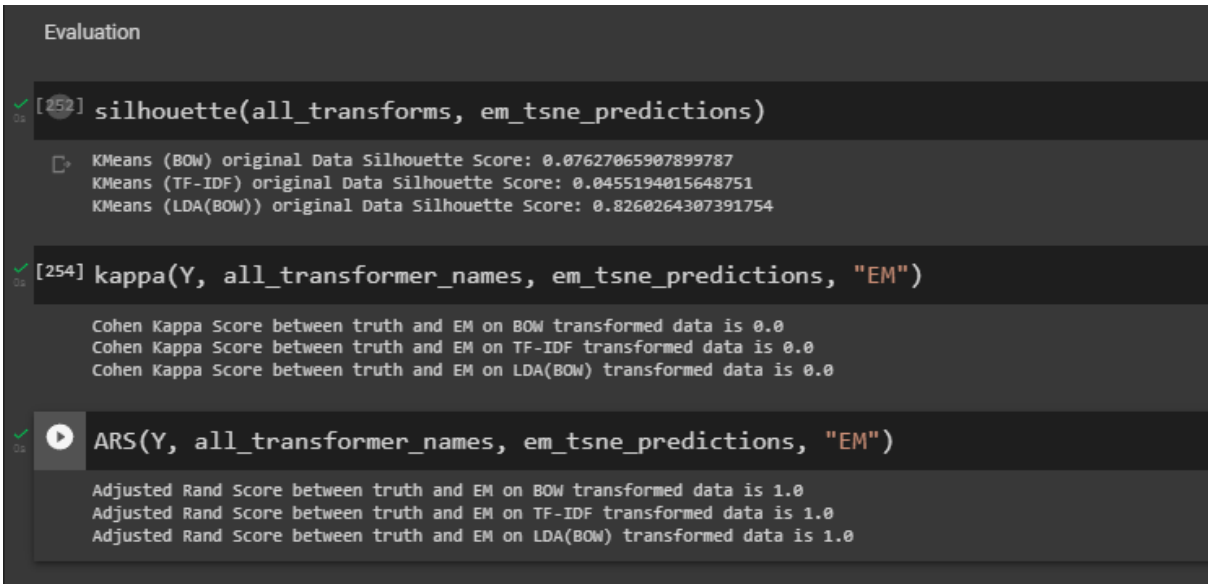

Fig. 2.5 Evaluations tsne-data

LDA in the evaluations is proved the best again and TF-IDF the worst.

3.2.3 Error Analysis

Now we perform the error analysis by plotting the top 10 frequent words for standard data and top 10 for TSNE data.
Fig 2.6 to Fig 2.15 show the top 10 frequent words for standard data.

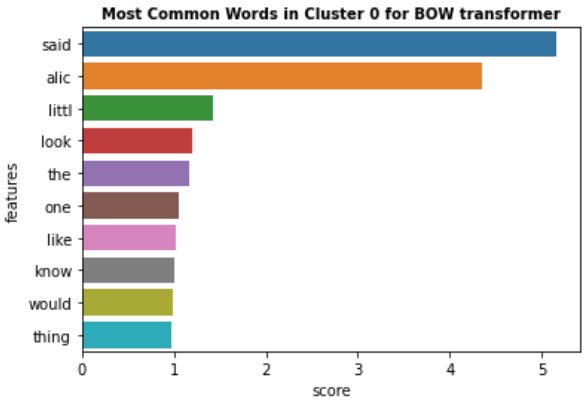

Fig 2.6

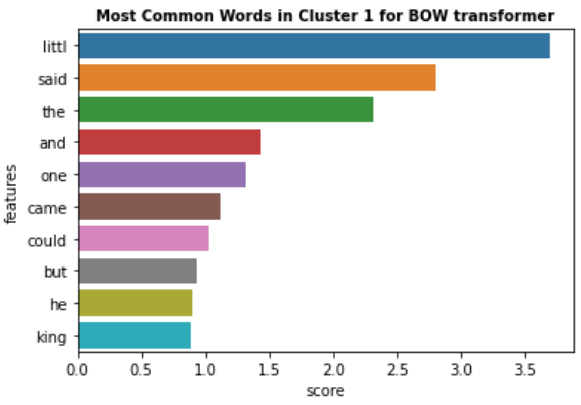

Fig 2.7

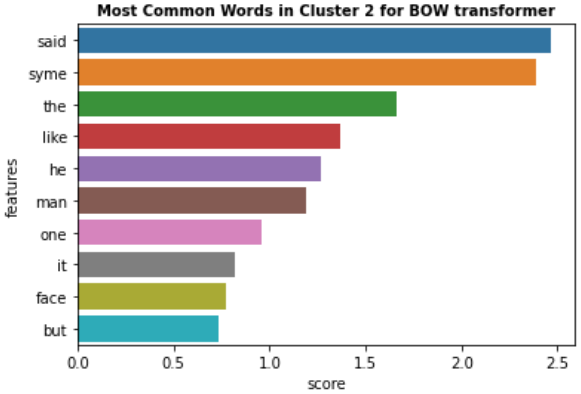

Fig 2.8

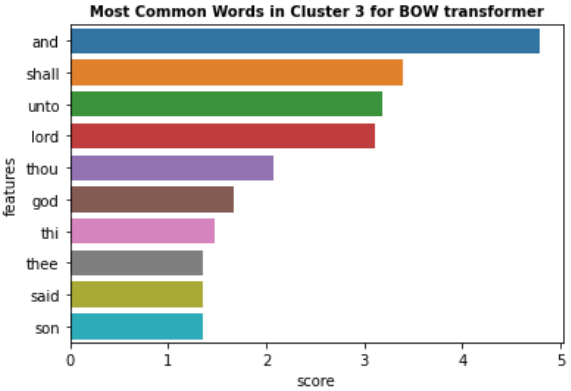

Fig 2.9

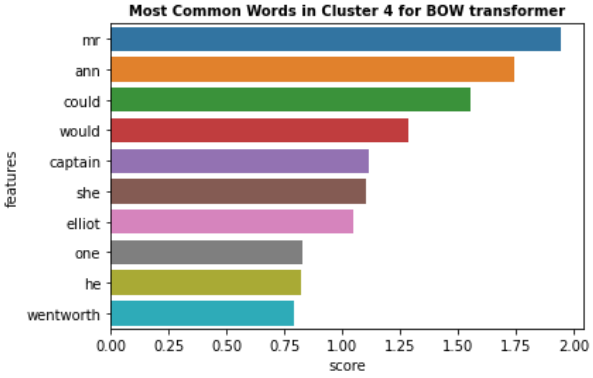

Fig 2.10

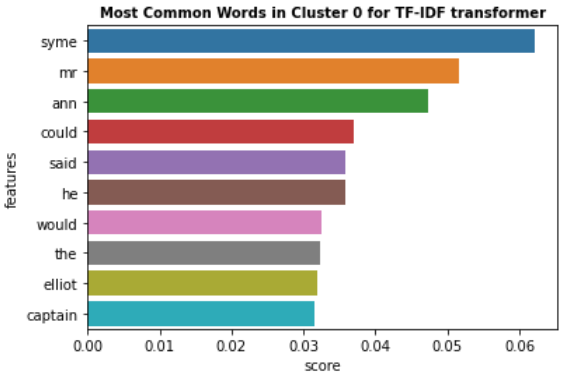

Fig 2.11

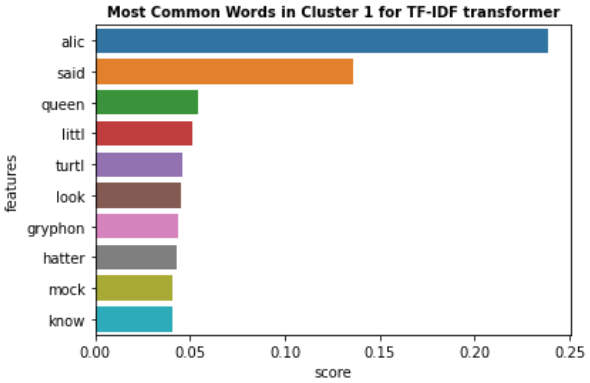

Fig 2.12

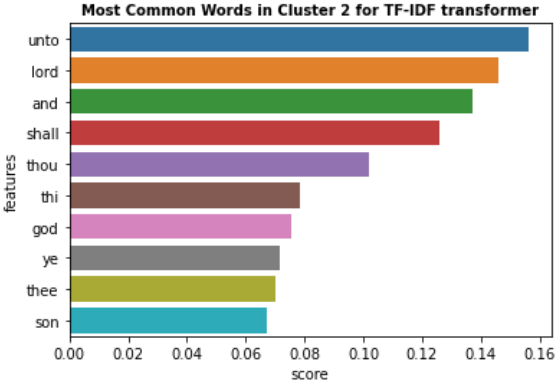

Fig 2.13

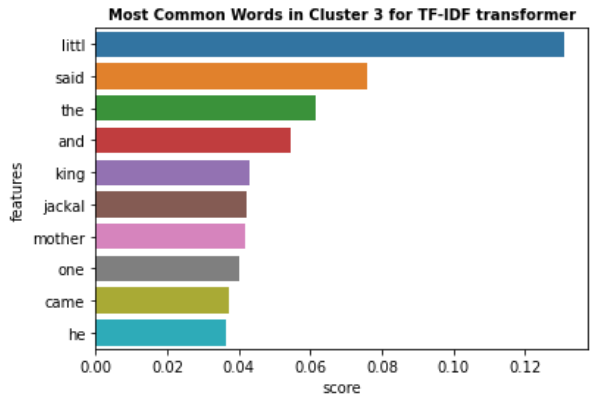

Fig 2.14

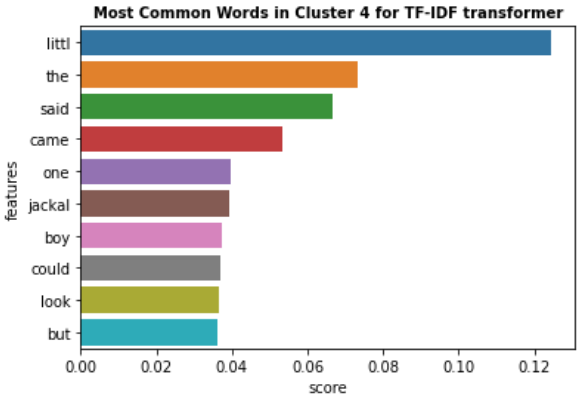

Fig 2.15

Fig 2.16 to Fig 2.25 show the top 10 frequent words for TSNE data.

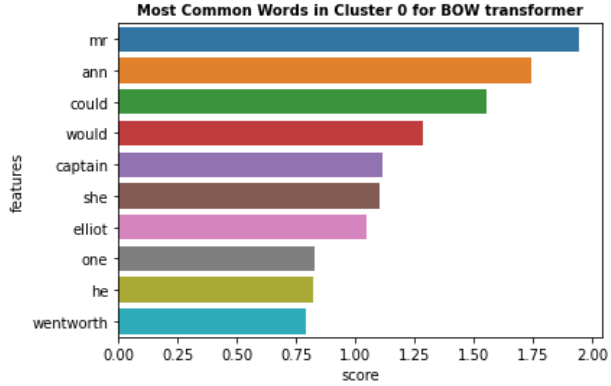

Fig 2.16

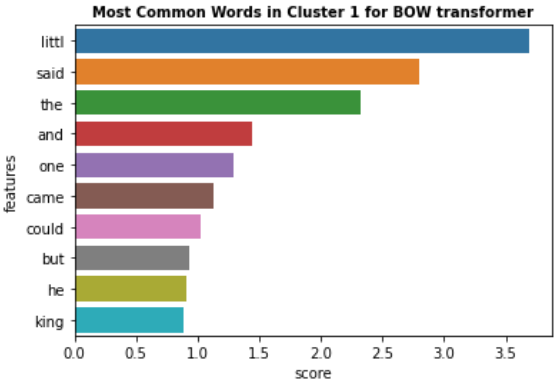

Fig 2.17

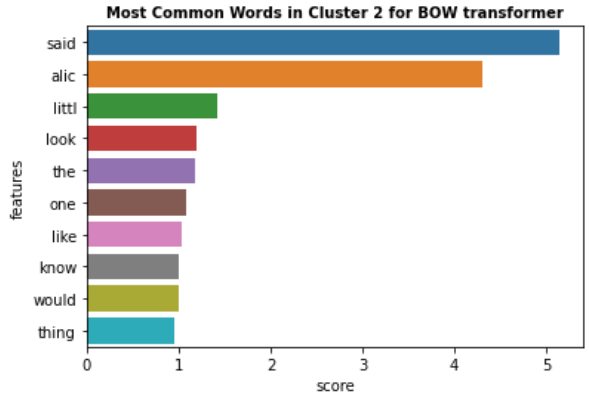

Fig 2.18

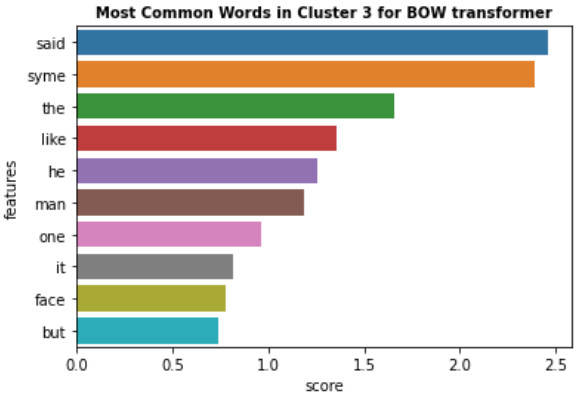

Fig 2.19

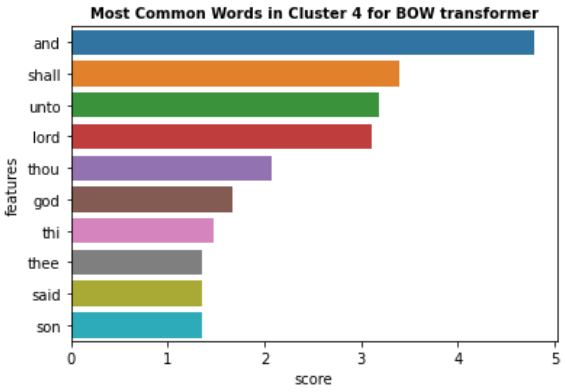

Fig 2.20

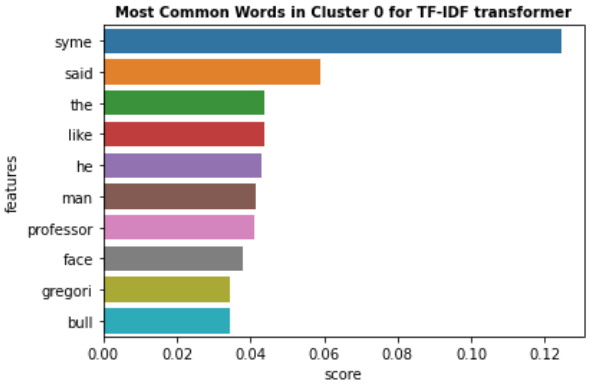

Fig 2.21

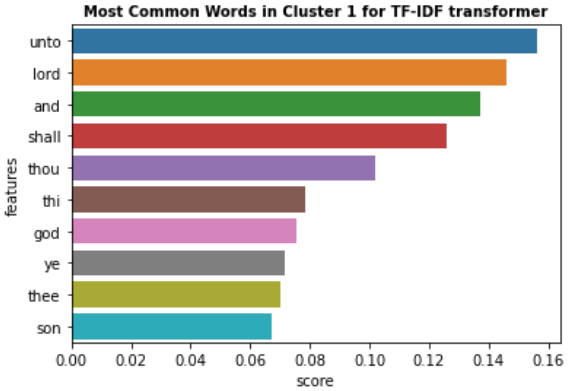

Fig 2.22

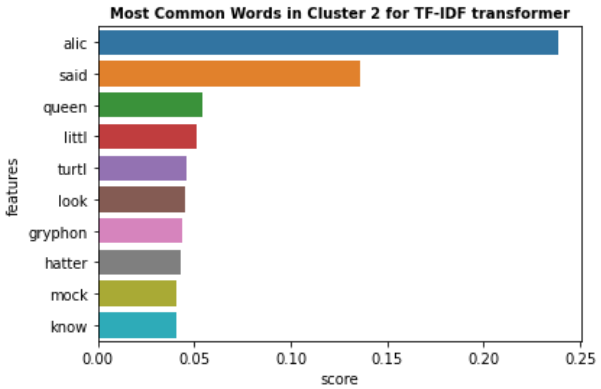

Fig 2.23

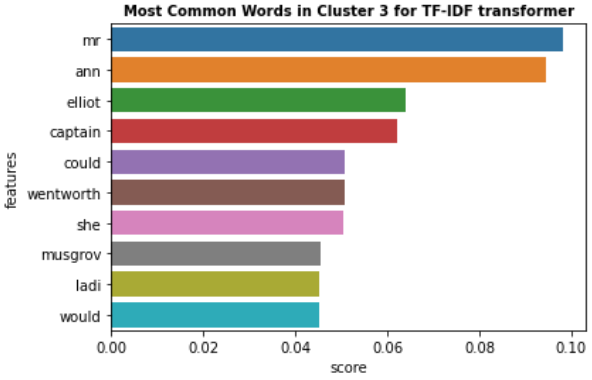

Fig 2.24

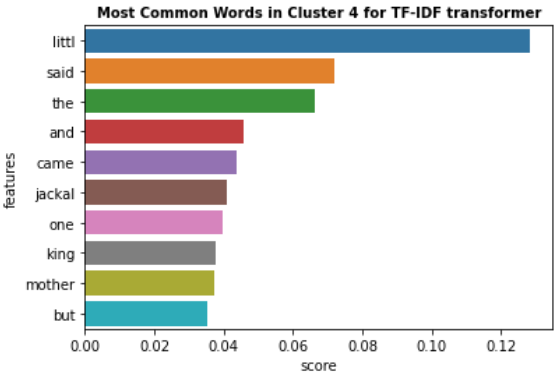

Fig 2.25

3.2.4 EM Conclusions

The EM clustering algorithm consists of two basic steps, E-step and M-step. Thus the performance of the machine is almost guaranteed to be improved after every iteration step. However, it had extremely slow convergence. Considering the text dataset is high-dimensional and PC may hardly implement it without the process of reducing dimensions using t-SNE. Besides, EM uses initialization parameters to start the first E-step. Therefore, initialization with some insights through cooperating with other algorithms can increase the EM overall performance.

3.3. Agglomerative clustering

3.3.1 Agglomerative Clustering Dendrogram

The two agglomerative clustering dendrograms with different feature engineering were drawn to show the whole process of clustering, the membership of the class so that we can know which individual cases are grouped into a cluster from a visual level. Therefore, the dendrogram is a good reference to identify the number of clusters.

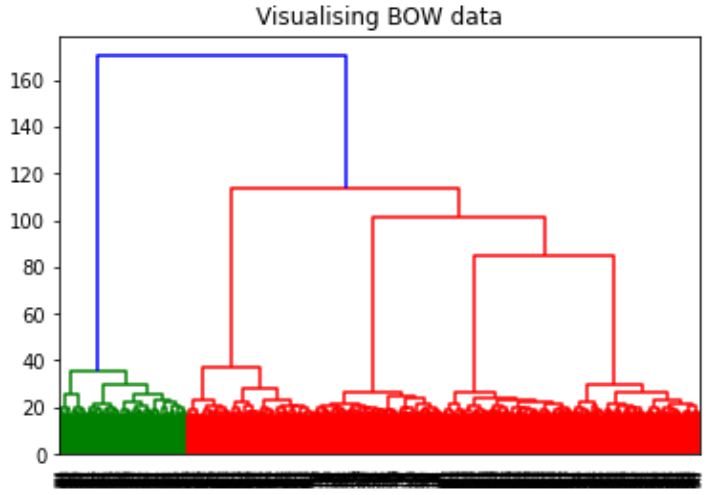

Fig 3.1 dendrogram(BOW)

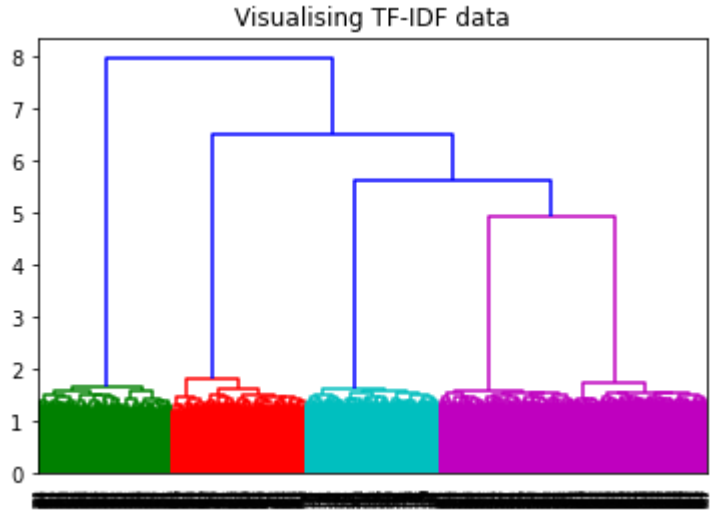

Fig 3.2 dendrogram(D2V)

3.3.2 Three Different Linkages

We first compared single and ward linkages. Different linkages with different feature engineering are shown in the below figures(Fig 3.3, 3.4, 3.5, 3.6,3.7,3.8).

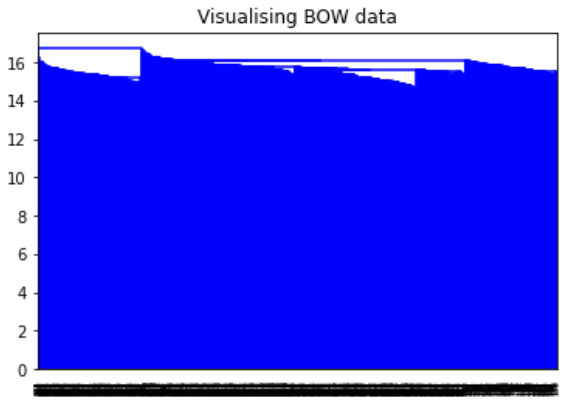

Fig. 3.3 Single Linkage (BOW)

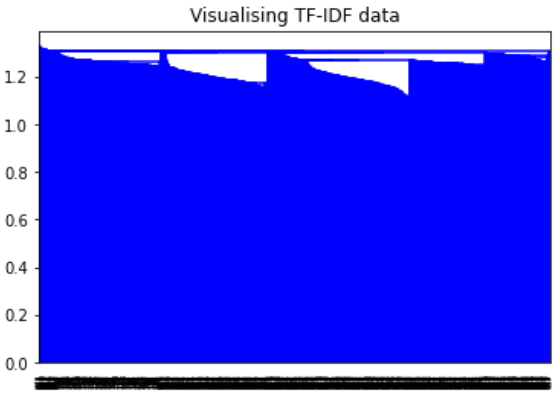

Fig. 3.4 Single Linkage (TF-IDF)

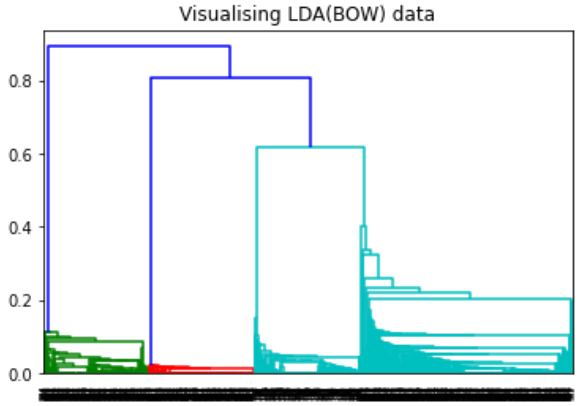

Fig. 3.5 Single Linkage (LDA)

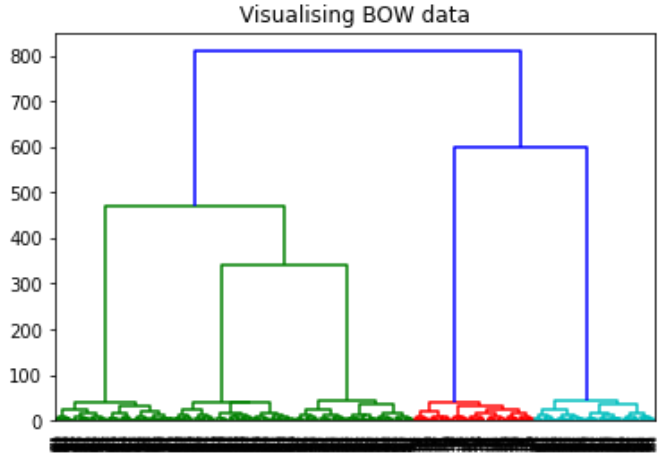

Fig. 3.6 Ward Linkage (BOW)

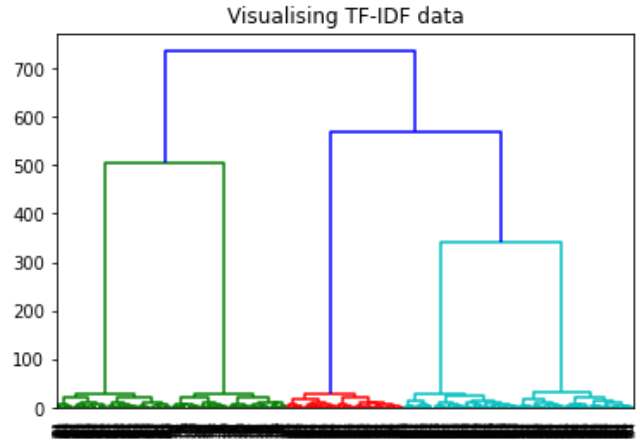

Fig. 3.7 Ward Linkage (TF-IDF)

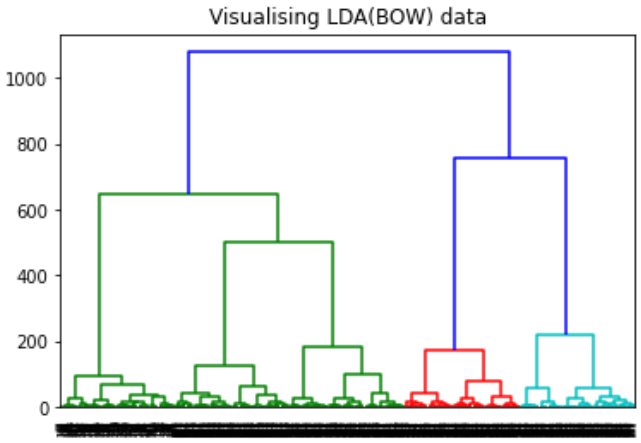

Fig. 3.8 Ward Linkage (LDA)

The single linkage doesn't seem to group the data in a clean fashion, we guess it happens based on the meaning of single linkage: "A drawback of this method is that it tends to produce long thin clusters in which nearby elements of the same cluster have small distances, but elements at opposite ends of a cluster may be much farther from each other than two elements of other clusters. This may lead to difficulties in defining classes that could usefully subdivide the data"[1]

3.3.3 Agglomerative Evaluations

Evaluate the methods and models with respects of std-data (Fig 3.9)

1

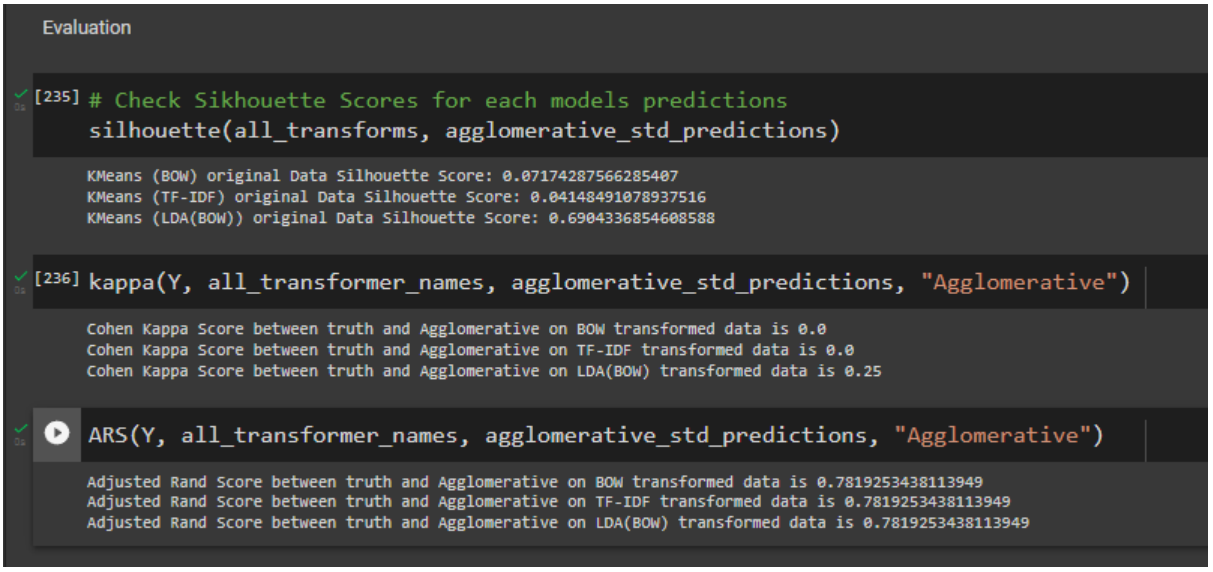

Fig. 3.9 Evaluations std-data

The consequences show that the LDA and BOW have highest ARS scores while BOW has the highest silhouette score but the LDA has the highest Kappa score. Based on the meaning of Kappa, LDA could be seen as the best model.

Evaluate the agglomerative tsne-data(Fig 3.10)

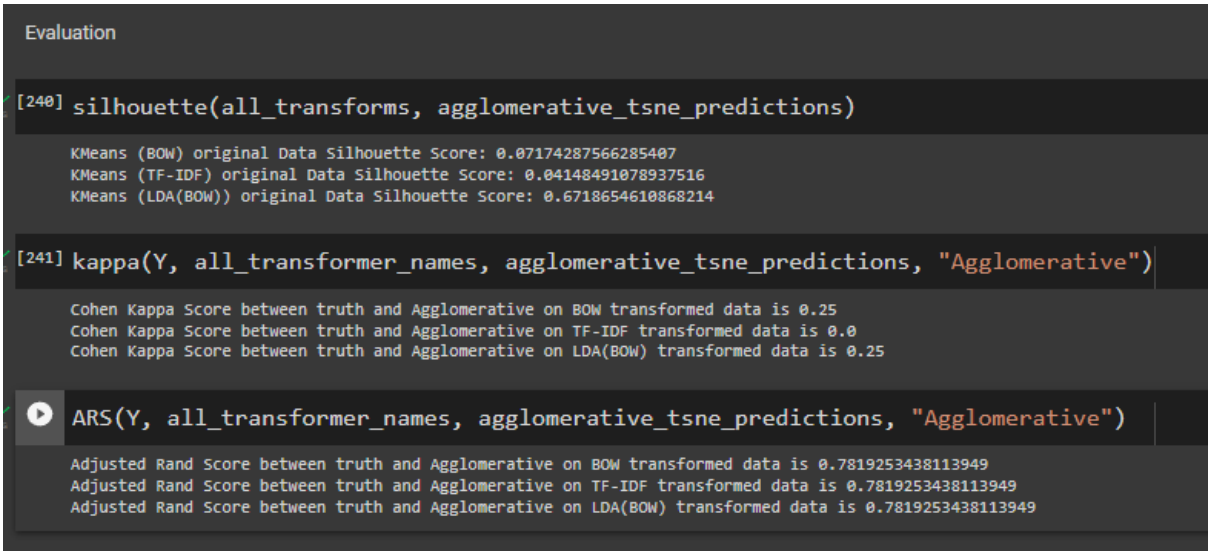

Fig. 3.10 Evaluations tsne-data

The results show that the LDA and BOW have highest ARS scores while BOW has the highest silhouette score, BOW is the best model.

3.3.4 Error Analysis

After all, perform error-analysis by using the top 10 frequent words for standard-data and top 10 for tsne-data.

Fig 3.11 to Fig 3.18 show the top 10 frequent words for standard data.

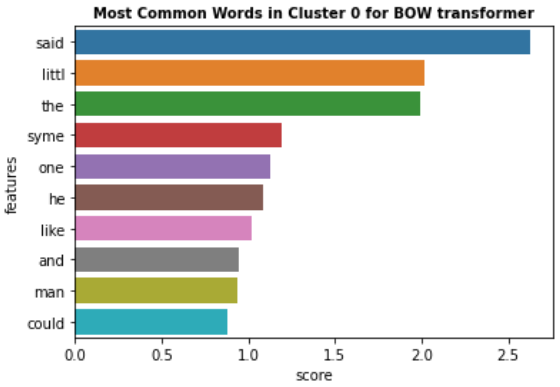

Fig. 3.11

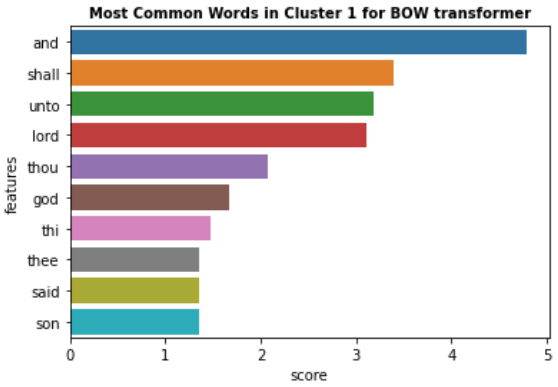

Fig. 3.12

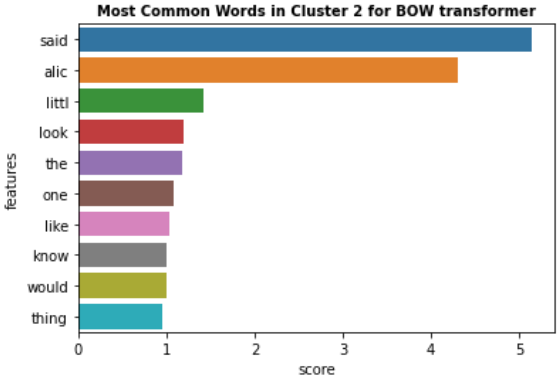

Fig. 3.13

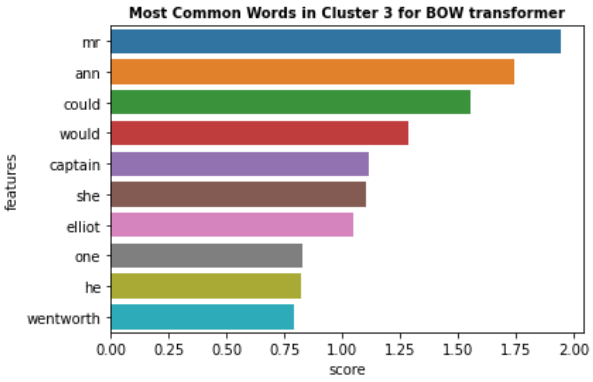

Fig. 3.14

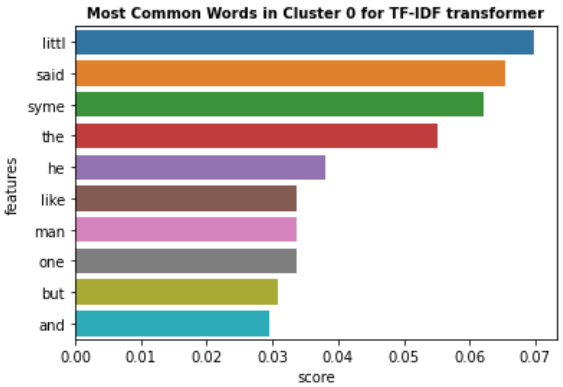

Fig. 3.15

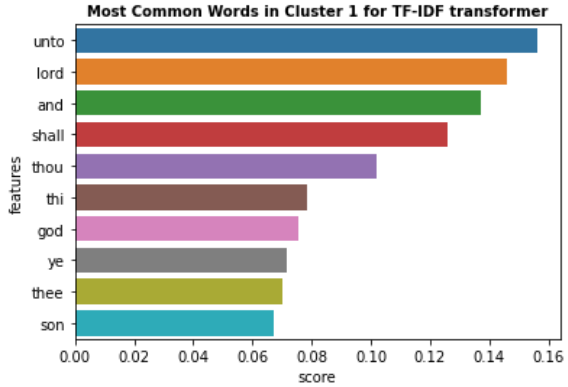

Fig. 3.16

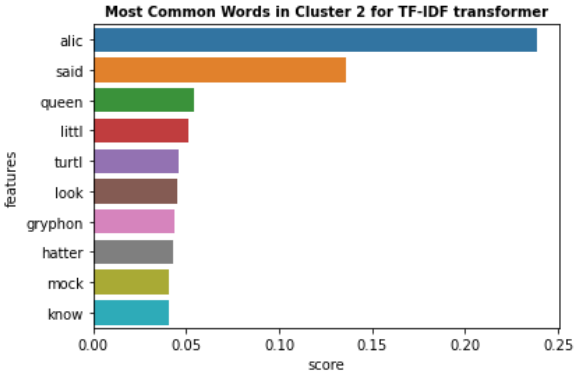

Fig. 3.17

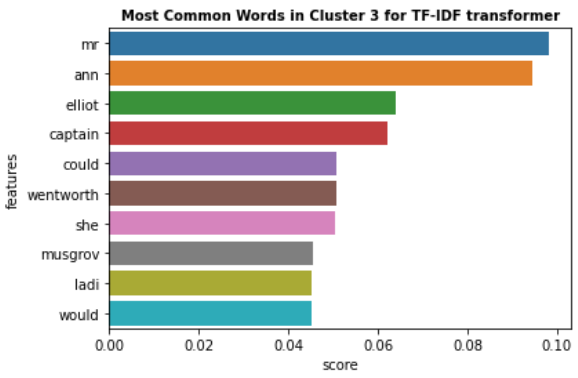

Fig. 3.18

Fig 3.19 to Fig 3.26 show the top 10 frequent word collocations for TSNE data.

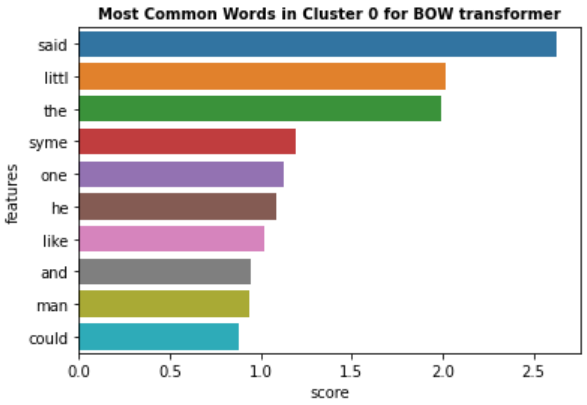

Fig. 3.19

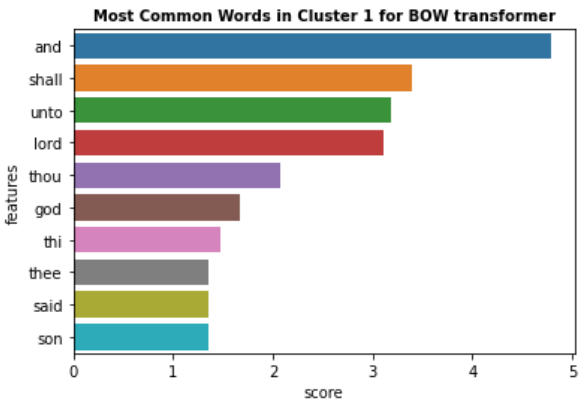

Fig. 3.20

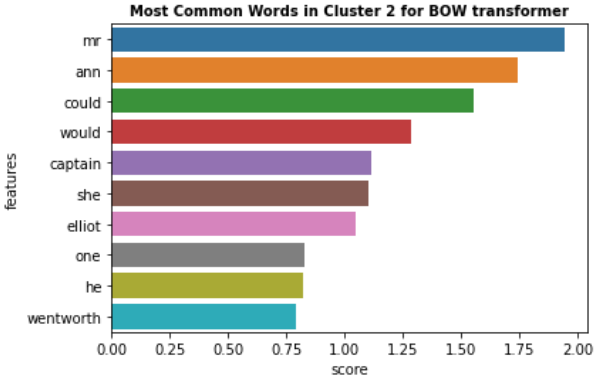

Fig. 3.21

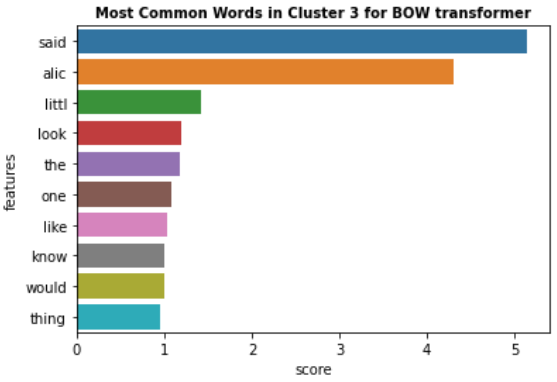

Fig. 3.22

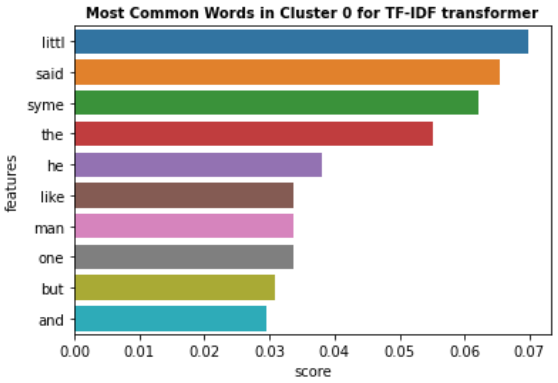

Fig. 3.23

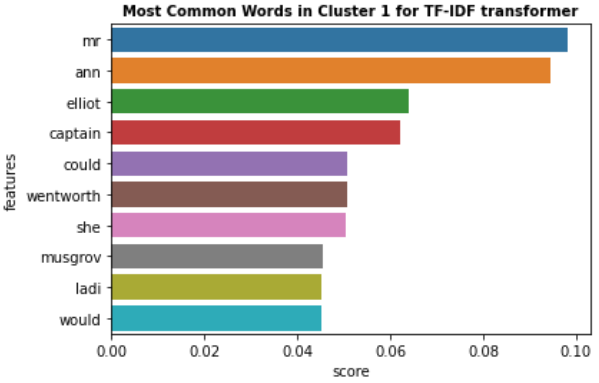

Fig. 3.24

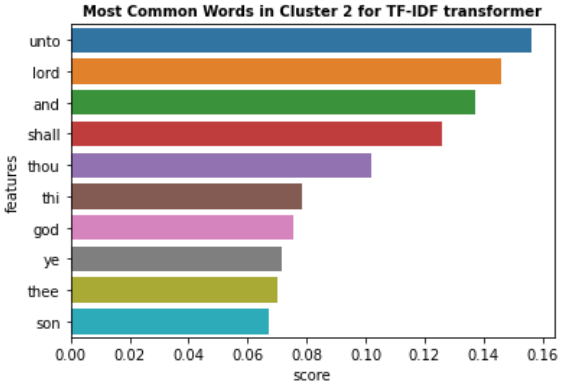

Fig. 3.25

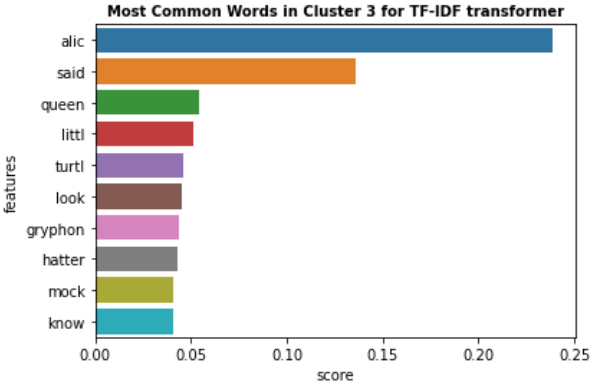

Fig. 3.26

3.3.5 Agglomerative Conclusions

The agglomerative clustering algorithm is easy to implement, and the number of clusters doesn't need to be predetermined. Second, the hierarchical relationship of classes can be discovered in the agglomerative clustering dendrogram. However, the time complexity and computational complexity are too high. Moreover, noise and outliers have a great impact on the clustering result. If the dataset is huge and has many outliers, the agglomerative clustering algorithm is not a good choice.

4. Conclusion

After several tests on different models and methods, we now come to the conclusion that the best clustering algorithm is KMeans after reducing dimensions. For feature engineering, the most accurate is LDA and the least accurate is BoW. And the KMeans method with LDA has the highest overall performance. For future improvement, we realize that the clustering calculation process is time-consuming and it would be helpful to have more efficient algorithms to reduce the run time.

Reference

1.. Single-linkage clustering - Wikipedia