2023

DSA_202101_5

Text Clustering Report

Assignment 2

Team Members (Group5):

1. Aya Metwally
2. Ahmed Nasser
3. Esraa Fayad
4. Abdelrahman Ali

Table of Contents

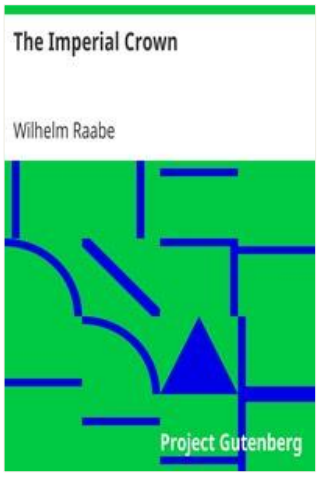
Introduction

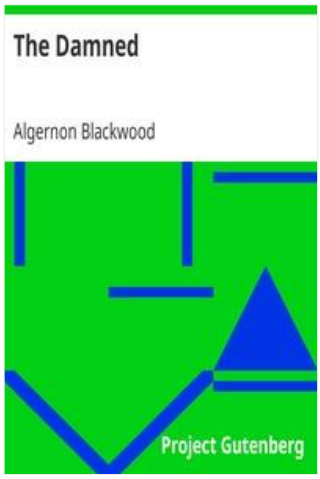
This project uses the example of five different digital Gutenberg books with different authors and genres. We train a machine learning model clustering that can generate 5 clusters, one cluster for every book. The final goal is to cluster the text using unsupervised techniques.

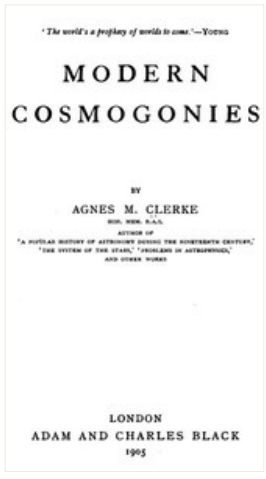
We used different algorithms like: K-means, Expectation Maximization (EM) and Hierarchal algorithm with different text vectorizers such as, Bag of Words, TF-IDF, Latent Dirichlet Allocation (LDA) and Word embedding (Word2Vec) and compare results of Kappa and Silhouette score for every algorithm with each text transformation technique.

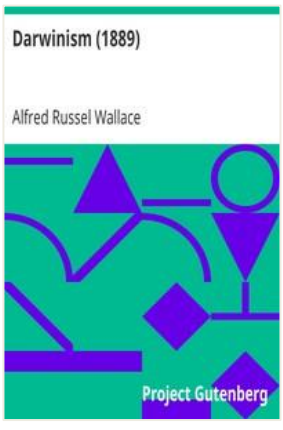
Data Preparing & Preprocessing

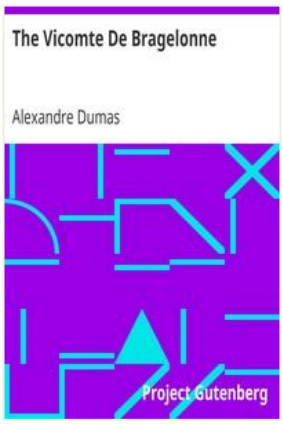
We choose different genre for five distinctive authors. The books are:











The authors are:

1. Wilhelm Raabe
2. Alfred Russel Wallace
3. Agnes Mary Clerke
4. Alexandre Dumas, Père
5. Algernon Blackwood

We prepared the data by randomly extracting 200 representative documents from each book source input. Then, we created a 150-word record for each document and labelled them as: 'Alfred Russel Wallace': 'a', 'Agnes Mary Clerke': 'b', 'Algernon Blackwood': 'c', 'Wilhelm Raabe': 'd', 'Alexandre Dumas, Père': 'e'

Next, we preprocessed the data by removing stopwords and unnecessary characters.

We also label authors as {0: 'Wilhelm Raabe', 1: 'Alfred Russel Wallace', 2: 'Agnes Mary Clerke', 3: 'Alexandre Dumas, Père', 4: 'Algernon Blackwood'}

	sentence	author	label
0	give second opportunity receive work electroni...	Wilhelm Raabe	d
1	men prague flee press steep path u stretch tir...	Wilhelm Raabe	d
2	need day accepted invitation promise lively ev...	Wilhelm Raabe	d
3	street nuremberg cousin friend grossen wait ga...	Wilhelm Raabe	d
4	word foolish weak tug heartstrings strong man ...	Wilhelm Raabe	d
...	...	...	...
995	repeat former question world think voice becom...	Algernon Blackwood	c
996	arm instead see really want bill say quietly t...	Algernon Blackwood	c
997	druid druid stone still lie copse field tumuli...	Algernon Blackwood	c
998	felt already idea rather horrible shiver ugly ...	Algernon Blackwood	c
999	unpure thing stammer expression sister talent ...	Algernon Blackwood	c

1000 rows × 3 columns

Fig1: labellig authors as a, b, c, etc

Data Visualization

We plot a histogram to make ensure that every book have 200 partitions:

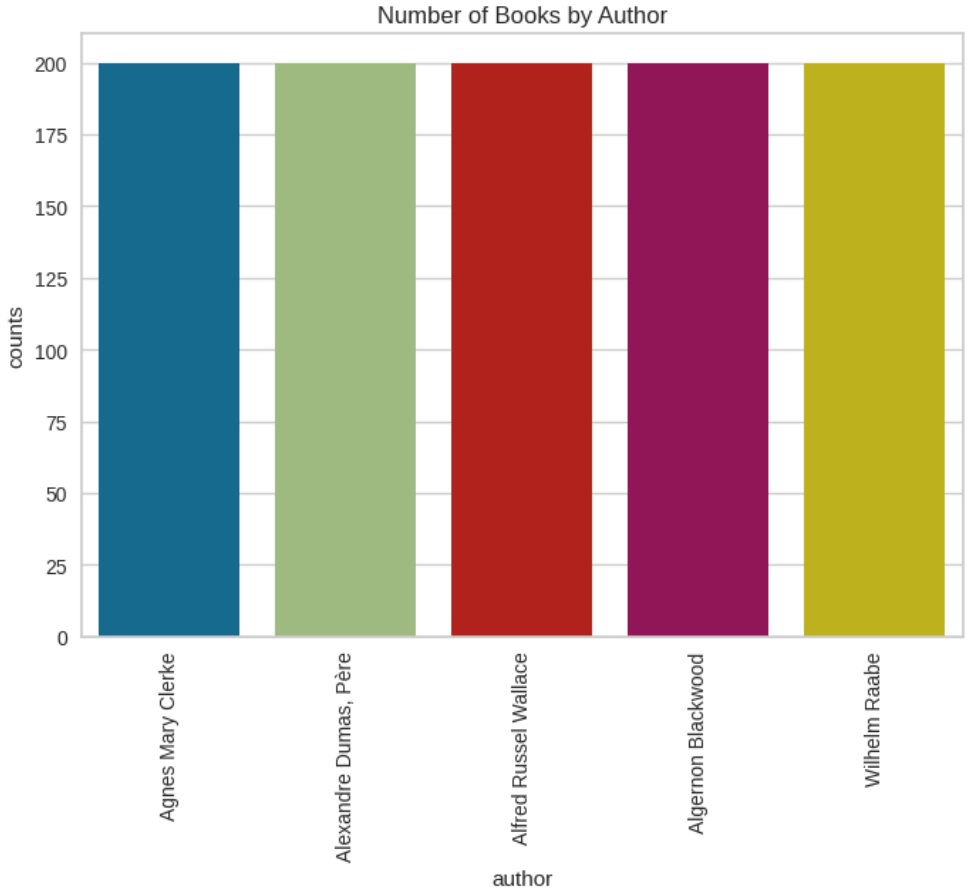


Fig2: Each author has 200 documents

We plot the length of characters for every document:

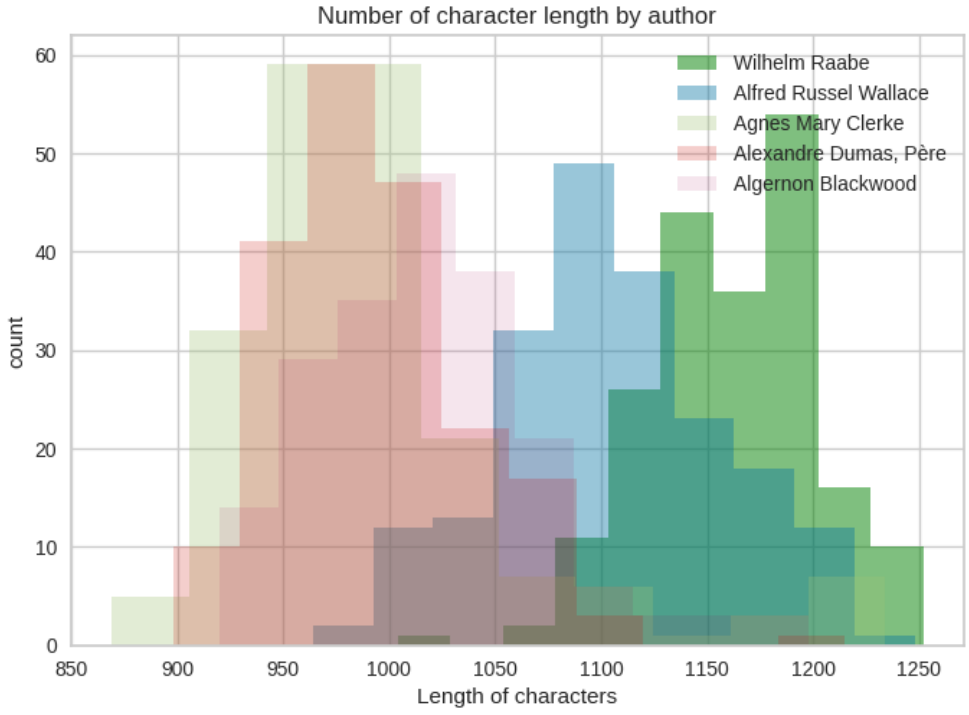


Fig3: Length of character by each author

We used the wordCloud to display the most common 100 words in each document:

Fig4: By author Wilhelm Raabe

Fig5: By author Alfred Russel

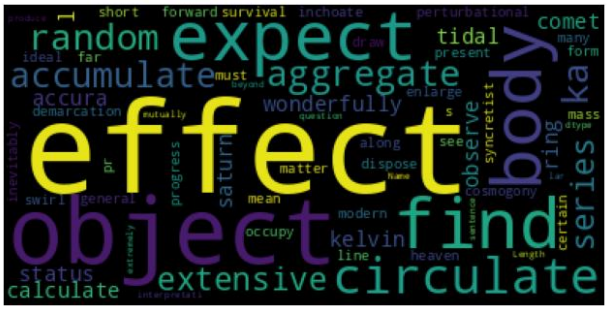


Fig6: By author Agnes Mary Clerke

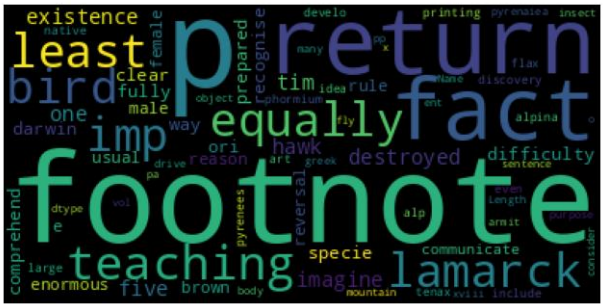


Fig7: By author Alexandre Dumas, Père

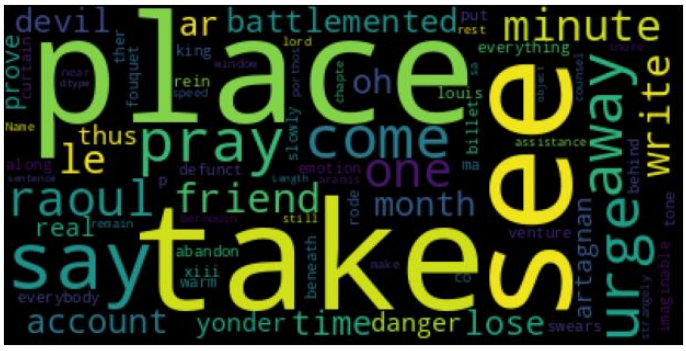
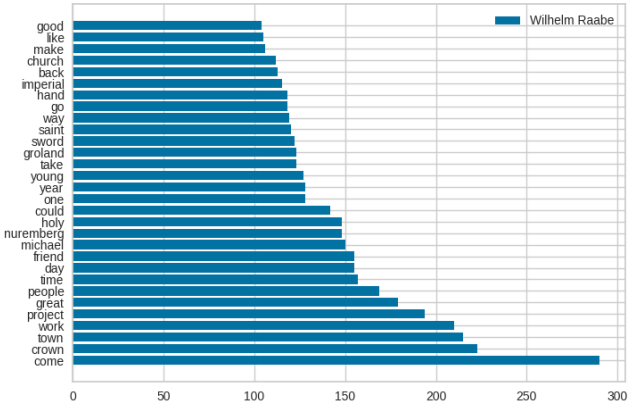
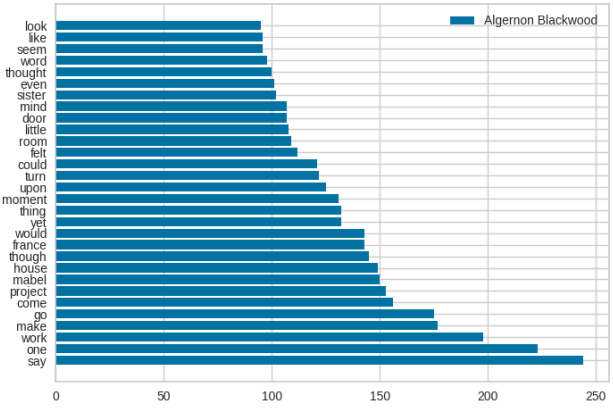


Fig8: common words by author Algernon Blackwood

We also used count vectorizer with ngram_range= (1,2) to Show the most frequent 30 words occurs in each book by author:





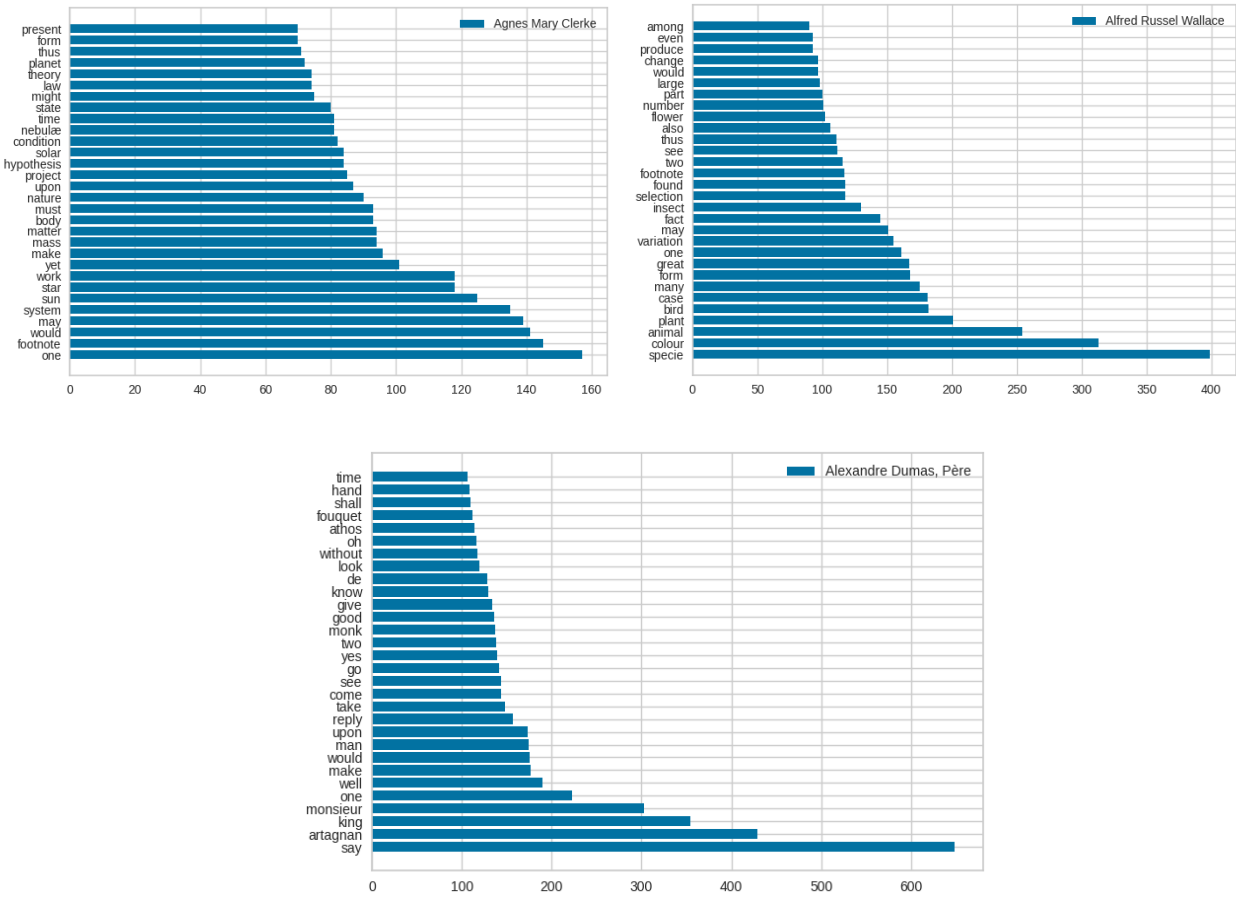


Fig9: top 30 words occurs in each book by author

Feature Engineering

For text transformation, we used 4 techniques: BOW, TF-IDF, LDA and Word2Vec. Each of these techniques has its own strengths and weaknesses, we apply each algorithm with every technique and compare results to choose the champion model.

1- Bag of word: A representation of descriptive text occurrences of words in the document.

	aachen	abandon	abash	abbe	abbey	abdon	abduction	abhor	abhors	abide	...	zool	zoological	zoologist	zoology	zoophyte	zöllner	æons	æsthetic	éclat	über
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

1000 rows × 11086 columns

Fig10: The transformation of BOW

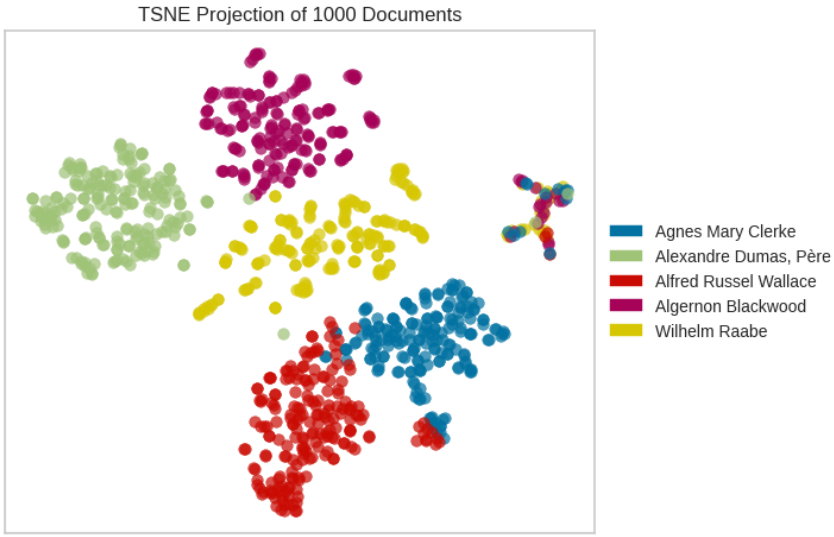


Fig11: The TSNE of BOW

2- Term Frequency-Inverse Document Frequency: Convert a collection of raw documents to a matrix of TF-IDF features

	aachen	abandon	abash	abbe	abbey	abdon	abduction	abhor	abhors	abide	...	zool	zoological	zoologist	zoology	zoophyte	zöllner	æons	æsthetic	éclat	über
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
996	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
997	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
998	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
999	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

1000 rows × 11086 columns

Fig12: The transformation of TF-IDF

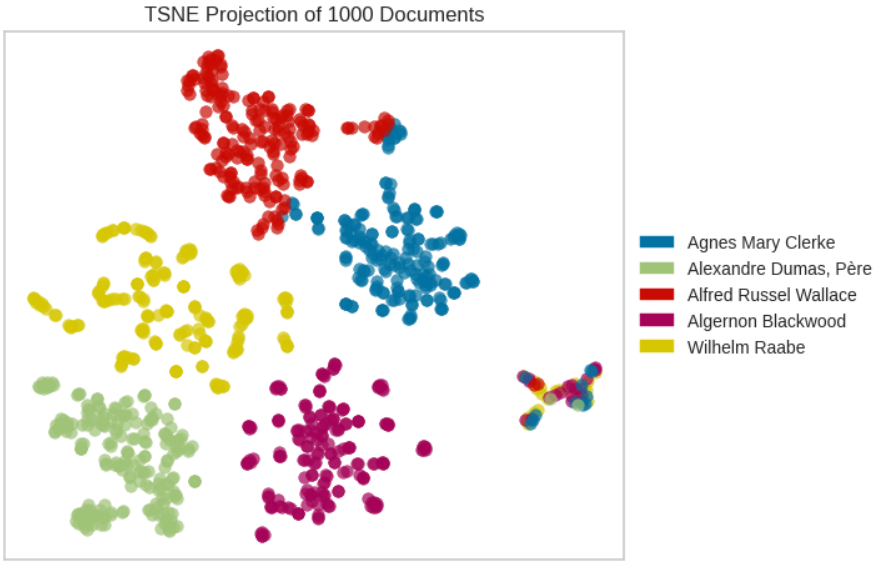


Fig13: The TSNE of TF-IDF

3- Latent Dirichlet Allocation (LDA): a text mining tool commonly used to discover hidden semantic structures within text bodies. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions.

	1	2	3	4	5	result
32	0.041215	0.106005	67.821861	0.122088	82.298851	5
66	0.041215	0.105999	15.278025	125.815125	8.915962	4
34	0.041215	4.630587	2.586063	0.122056	142.174133	5
283	0.041215	0.105991	40.361607	0.122014	109.711823	5
698	4.315674	133.915665	0.130702	4.575485	7.398367	2
628	0.041215	0.105982	150.070175	0.121966	0.143647	3
436	0.041215	0.105990	19.331919	130.747589	0.143656	4
564	0.041215	0.105981	150.093094	0.121981	0.143724	3
706	0.041215	0.105996	16.513929	119.045555	14.653898	4
200	1.229845	0.105988	136.471649	12.511062	0.143779	3

Fig14: 10 samples from LDA

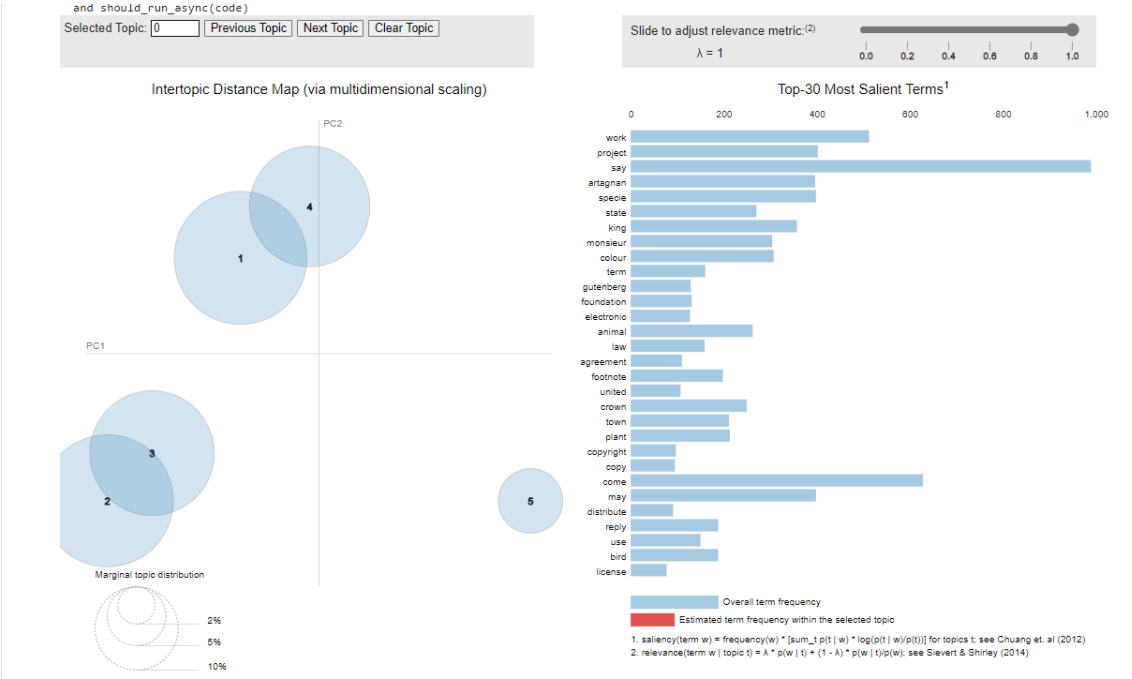


Fig15: LDA per topic model

4- Word Embedding (Word2Vec): consists of models for generating word embedding.

```
[ 0.01037092 -0.01678682 -0.07521727  0.00458486  0.2223188   0.00154048
 -0.12182119  0.32594147  0.07134296  0.09678854  0.10755903 -0.05733206
 -0.0668935   0.11390691 -0.23886973 -0.00097043  0.03980597 -0.15627016
 -0.01209488  0.30480072 -0.12780377 -0.04022014  0.11936542  0.07076063
  0.14959271  0.07513668 -0.12631875 -0.07085946 -0.07682614 -0.15998057
 -0.04901443  0.11070156 -0.0194509  -0.11388925 -0.05088202 -0.02877268
  0.2480938   0.10742459  0.22948514 -0.1826965  -0.04307009  0.02303656
 -0.14436541 -0.06625766  0.08619523  0.00740569  0.04585157  0.11219039
 -0.11459982  0.19330774 -0.09414732  0.04999454 -0.19432081 -0.05752908
 -0.06612711  0.1930504  -0.04758674  0.08155711 -0.02832458  0.00769949
 -0.10601158 -0.17930183 -0.19340406  0.05089835  0.13882987 -0.0852828
 -0.08320747 -0.23168932 -0.27455682 -0.13165131  0.15984206  0.13540949
  0.05261063 -0.36392695 -0.14242136 -0.0524765  -0.0579896   0.05002953
 -0.15611424  0.15988608  0.05474166 -0.25364986 -0.04586937  0.3203103
 -0.02720785  0.05547034 -0.10055982  0.08244028  0.18555823 -0.23594071
  0.1653683  -0.21503013 -0.01836672  0.01546138  0.12463352 -0.02173006
  0.0918555  -0.01866426 -0.08264756  0.19031495 -0.02286647  0.06640086
  0.04782354  0.03074789 -0.13924627 -0.15315796  0.23618019  0.12494505
 -0.2775405  -0.01513561 -0.14786395  0.14728178  0.1374951  -0.05549163
 -0.01973082  0.09457026  0.04505717 -0.01180082  0.10683841 -0.10192445
 -0.03772422  0.01120619  0.06045036 -0.08074109  0.12830651  0.04007745
 -0.04707736 -0.08822856 -0.13674282  0.01221876  0.12335434  0.33173248
 -0.05627306 -0.12807107  0.19772905  0.09733415 -0.3277545  -0.0341846
 -0.02466532  0.06852863  0.14631571  0.07004492  0.12189461  0.10522151
  0.26623338  0.09115966 -0.05490177 -0.0273051   0.12549077 -0.07311033]
```

Fig16: Vectorized prediction of Word2Vec

Dimensionality reduction: is the process of reducing the number of features (or dimensions) in a dataset while retaining as much information as possible.

This can be done for a variety of reasons, such as to reduce the complexity of a model, or to make it easier to visualize the data. There are several techniques for dimensionality reduction, including principal component analysis (PCA), and t-distributed Stochastic Neighbor Embedding (T-SNE). Each technique uses a different method to project the data onto a lower-dimensional space while preserving important information.

We used the data before and after transformation with PCA and T-SNE, and found that T-SNE technique is a good choice, so we used it. It is good for the computation time and good visualization. The champion model achieved high score while using T-SNE technique.

Clustering Algorithms

We train different models (K-Means, Expectation Maximization (EM), and Hierarchical algorithm) with each text transformation technique to Cluster the text and compare the results of Kappa and Silhouette score for them to choose the champion model.

- K-means with BOW:

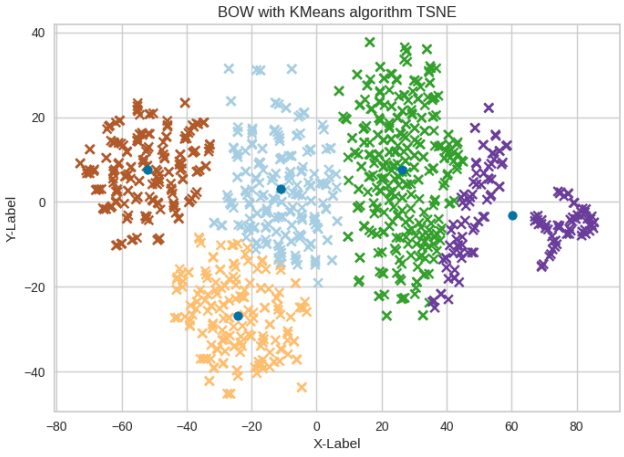


Fig17: BOW with K-means

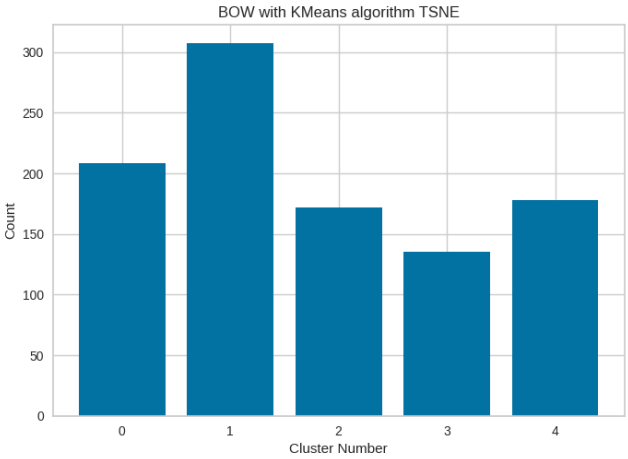


Fig18: Histogram BOW with K-means

- K-means with TF-IDF:

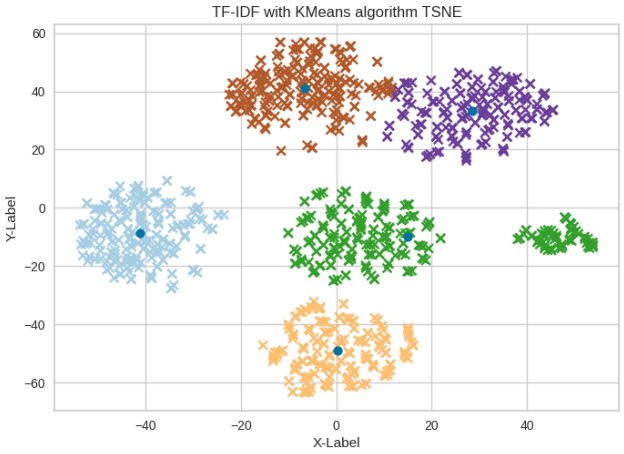


Fig19: TF-IDF with K-means

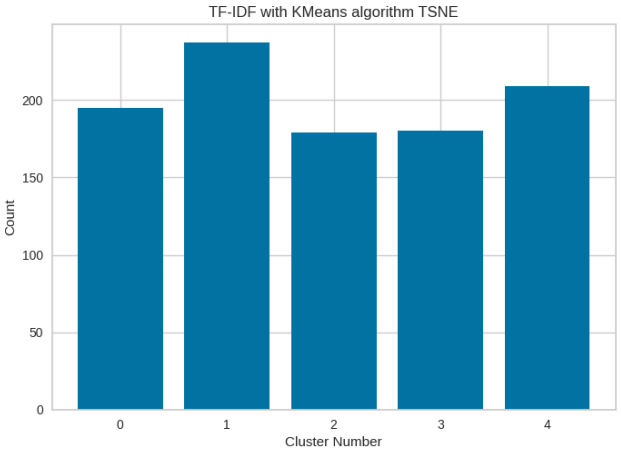


Fig20: Histogram TF-IDF with K-means

- K-means with LDA:

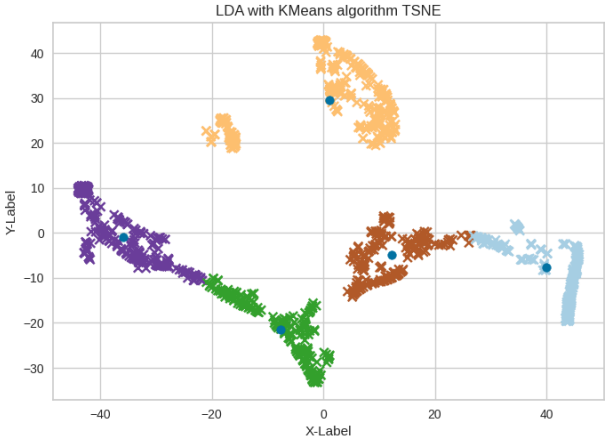


Fig21: LDA with K-means

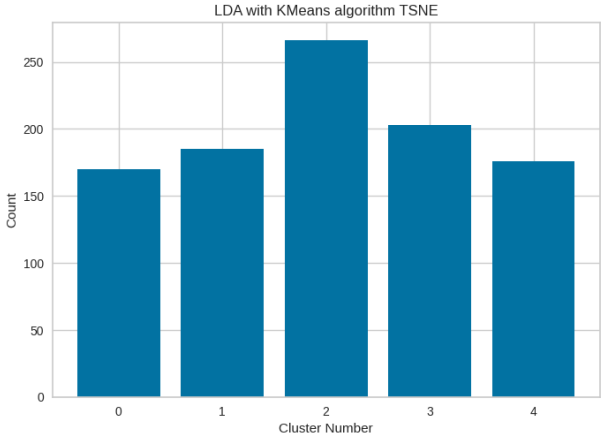


Fig22: Histogram LDA with K-means

- K-means with Word2Vec:

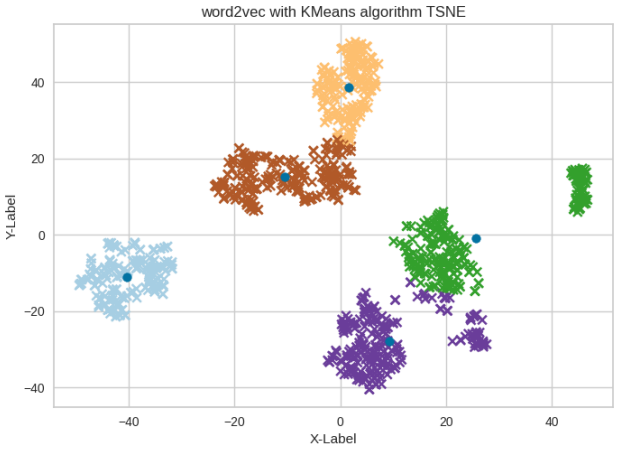


Fig23: Word2Vec with K-means

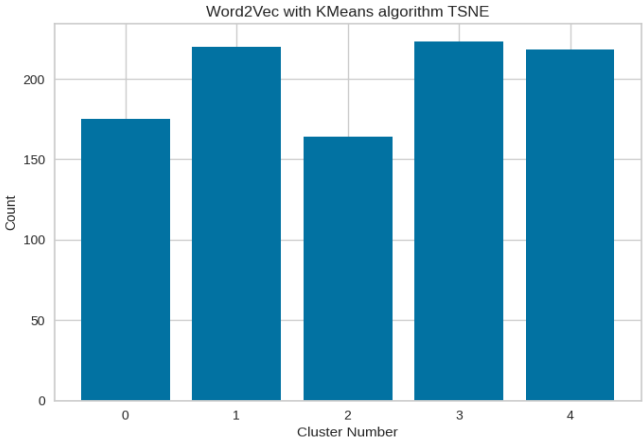


Fig24: Histogram Word2Vec with K-means

- Expectation Maximization with BOW:

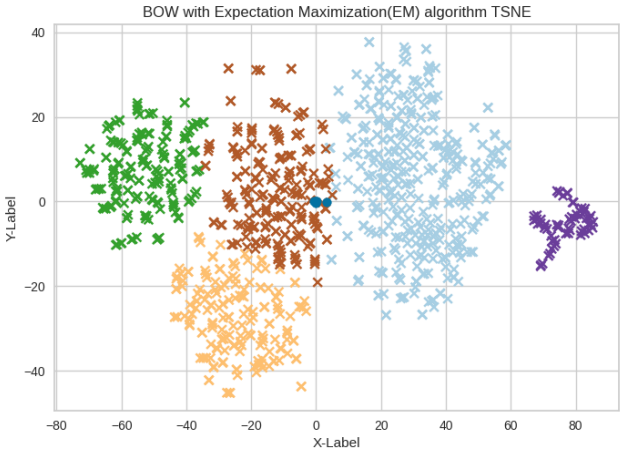


Fig25: BOW with EM

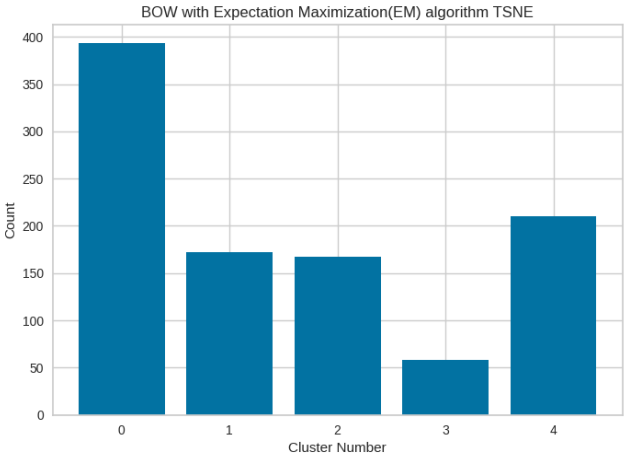


Fig26: Histogram BOW with EM

- Expectation Maximization with TF-IDF:

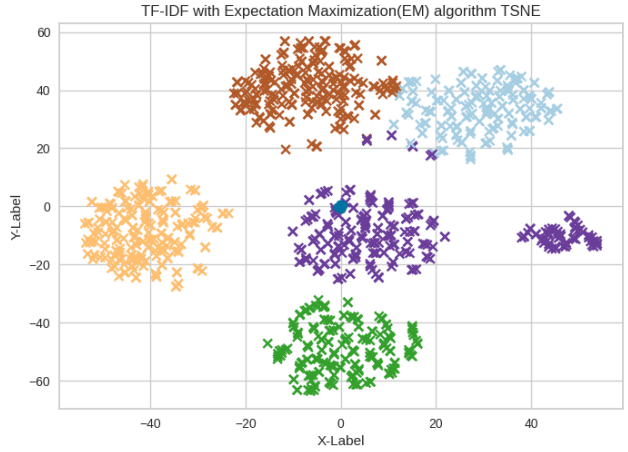


Fig27: TF-IDF with EM

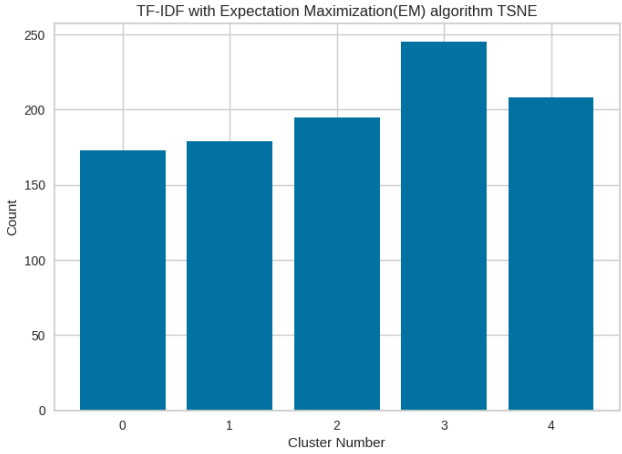


Fig28: Histogram TF-IDF with EM

- Expectation Maximization with LDA:

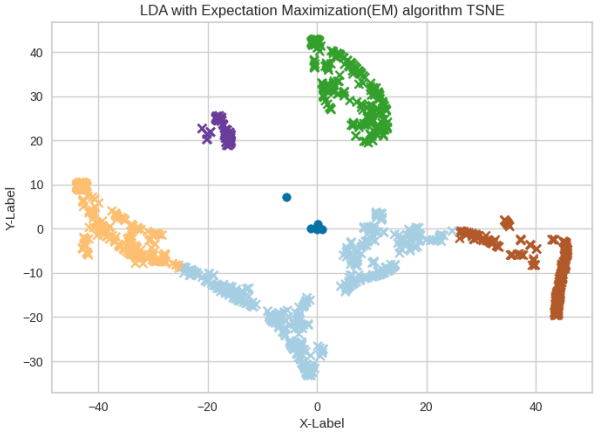


Fig29: LDA with EM

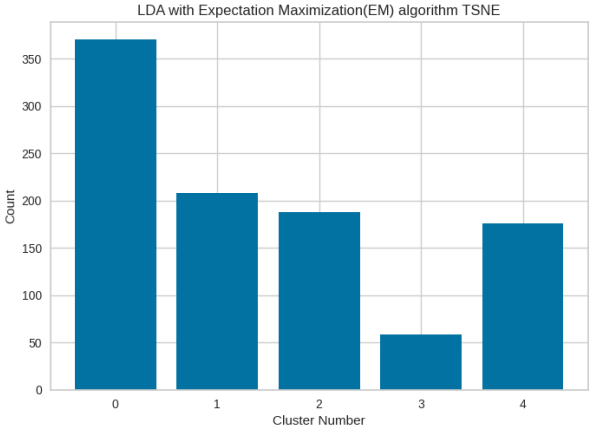


Fig30: Histogram LDA with EM

- Expectation Maximization with Word2Vec:

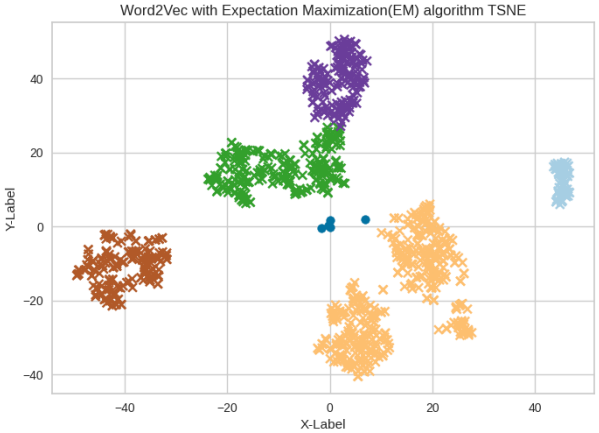


Fig31: Word2Vec with EM

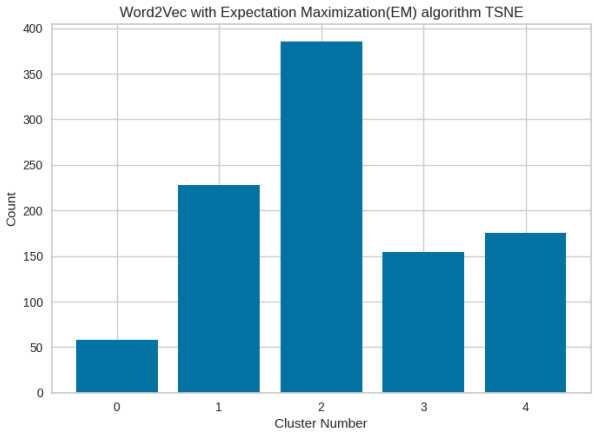


Fig32: Histogram Word2Vec with EM

- Hierarchical with BOW:

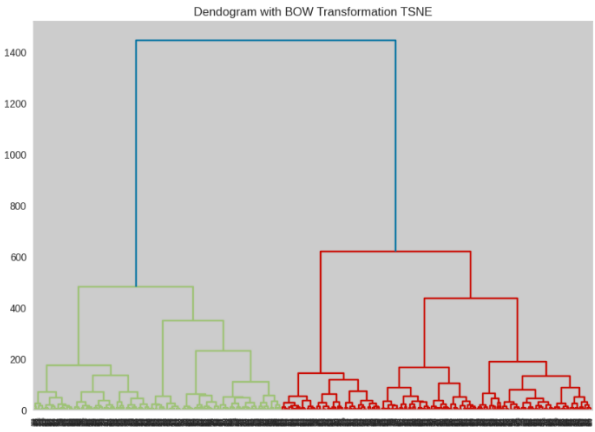


Fig33: Dendrogram with BOW for hierarchical

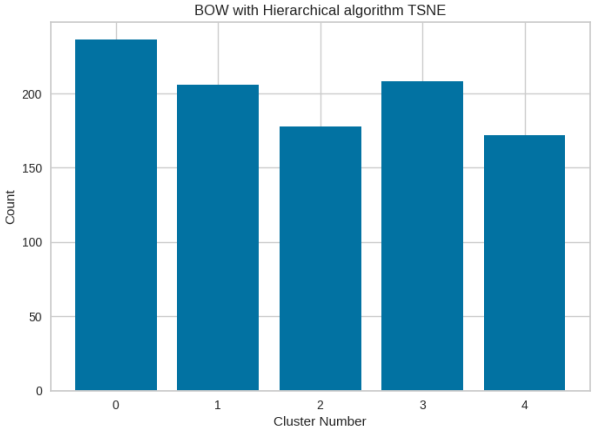


Fig34: Histogram BOW with hierarchical

- Hierarchical with TF-IDF:

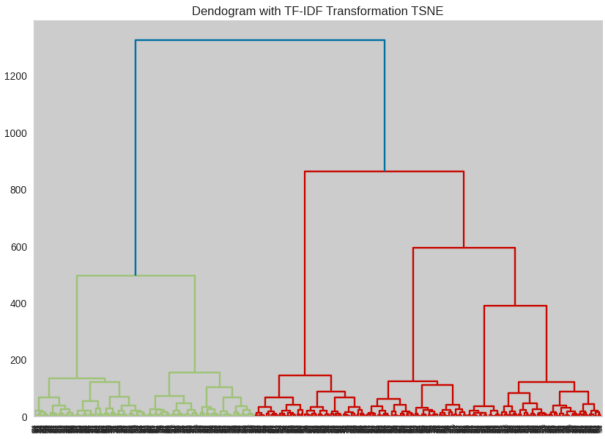


Fig35: Dendrogram with TF-IDF for hierarchical

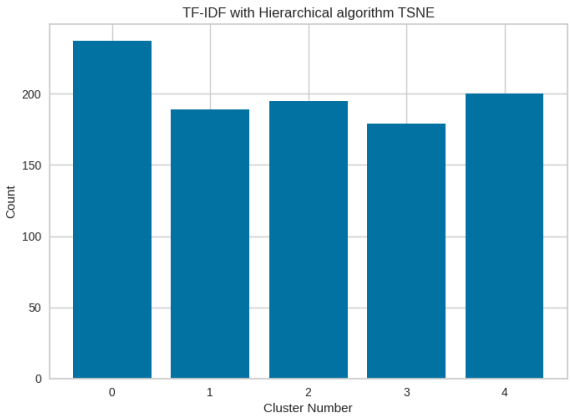


Fig36: Histogram TF-IDF with hierarchical

- Hierarchical with LDA:

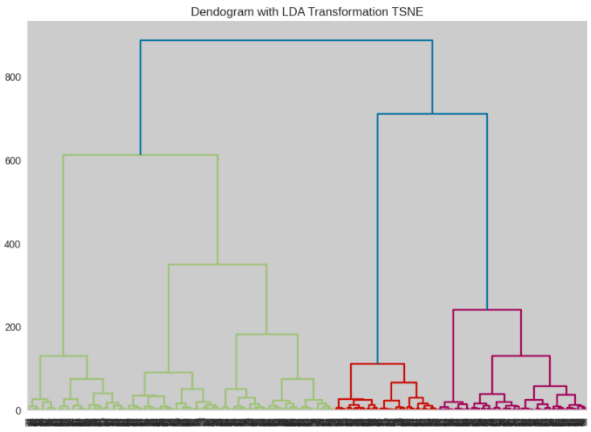


Fig37: Dendrogram with LDA for hierarchical

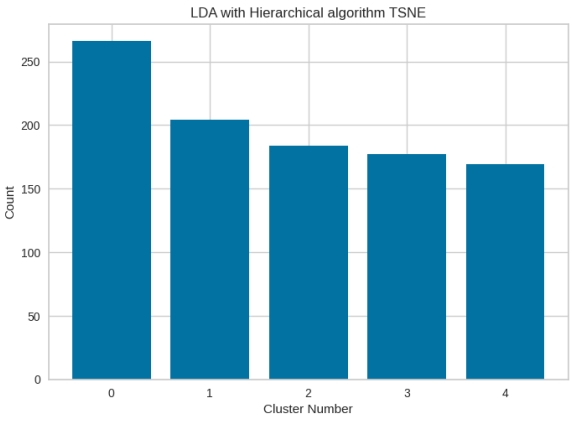


Fig38: Histogram LDA with hierarchical

- Hierarchical with Word2Vec:

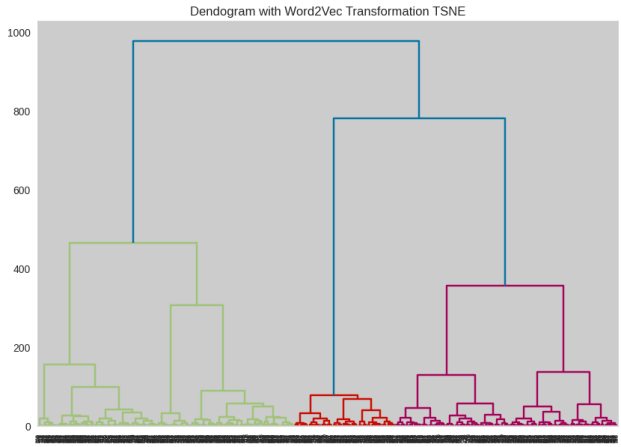


Fig39: Dendrogram with Word2Vec for hierarchical

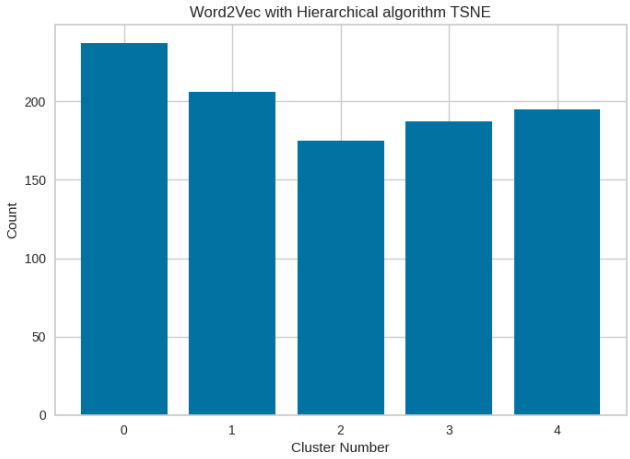


Fig40: Histogram Word2Vec with hierarchical

Evaluations

We calculated Kappa against true authors, Coherence and Silhouette.

1- Kappa

1.1- Kappa's Score with K-Means algorithm

K-means with BOW	K-means with TF-IDF	K-means with LDA	K-means with Word2Vec
0.73625	0.93625	0.87	0.84375

1.2- Kappa's Score with EM algorithm

EM with BOW	EM with TF-IDF	EM with LDA	EM with Word2Vec
0.65875	0.9275	0.7262500000000001	0.65875

1.3- Kappa's Score with Hierarchical algorithm

Hierarchical with BOW	Hierarchical with TF-IDF	Hierarchical with LDA	Hierarchical with Word2Vec
0.8225	0.9225	0.89	0.88375

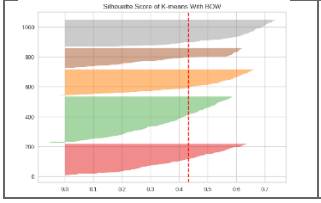
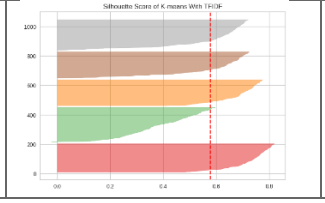
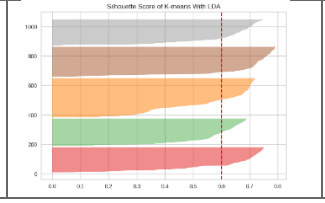
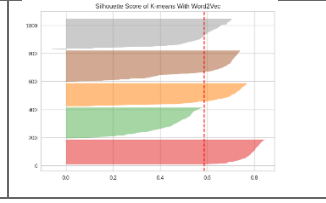
2- Coherence with LDA

Score using c_v: 0.5003(Higher value is better)

Score using u_mass: -2.0263 (lower value is better)

3- Silhouette

3.1- Silhouette's Score with K-Means algorithm

K-means with BOW	K-means with TF-IDF	K-means with LDA	K-means with Word2Vec
0.43264297	0.57847035	0.600884	0.58657855
			

3.2- Silhouette's Score with EM algorithm

EM with BOW	EM with TF-IDF	EM with LDA	EM with Word2Vec
0.44599622	0.5740476	0.57115	0. 61463964

3.3- Silhouette's Score with Hierarchical algorithm

Hierarchical with BOW	Hierarchical with TF-IDF	Hierarchical with LDA	Hierarchical with Word2Vec
0.41210836	0.5730236	0.5929133	0.57691103

The champion model:

Based on comparison between score of each model with every text transformation technique, we choose K-Means algorithm with TF-IDF Technique that has the highest score in Kappa evaluation with 0.938 and produces result which is the closest to the human labels.

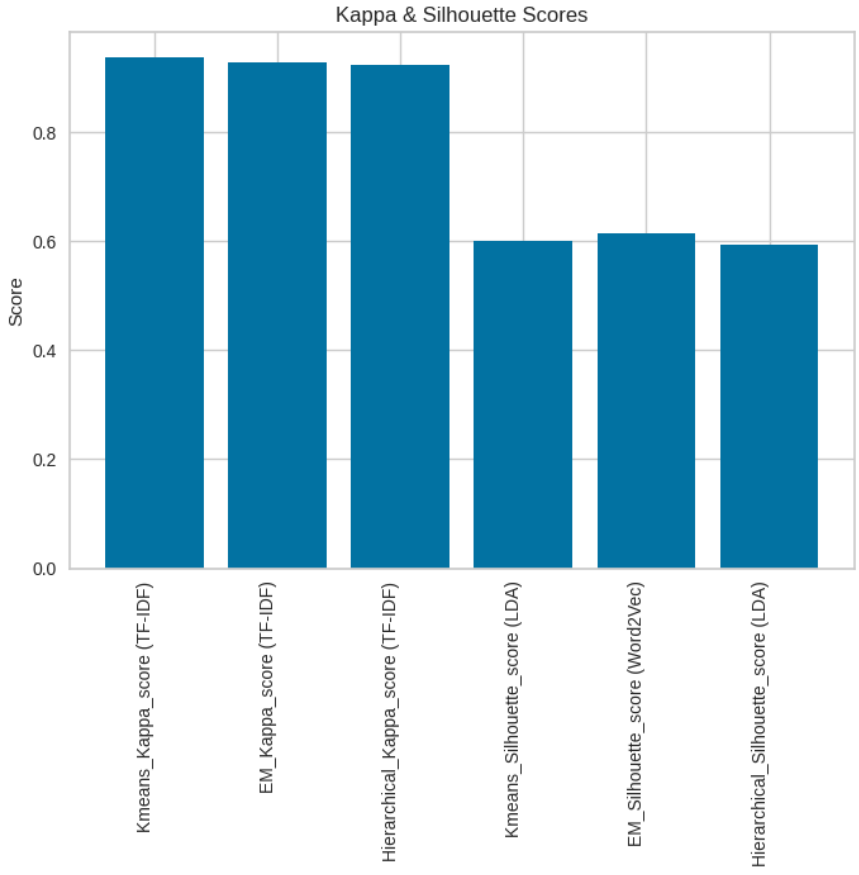
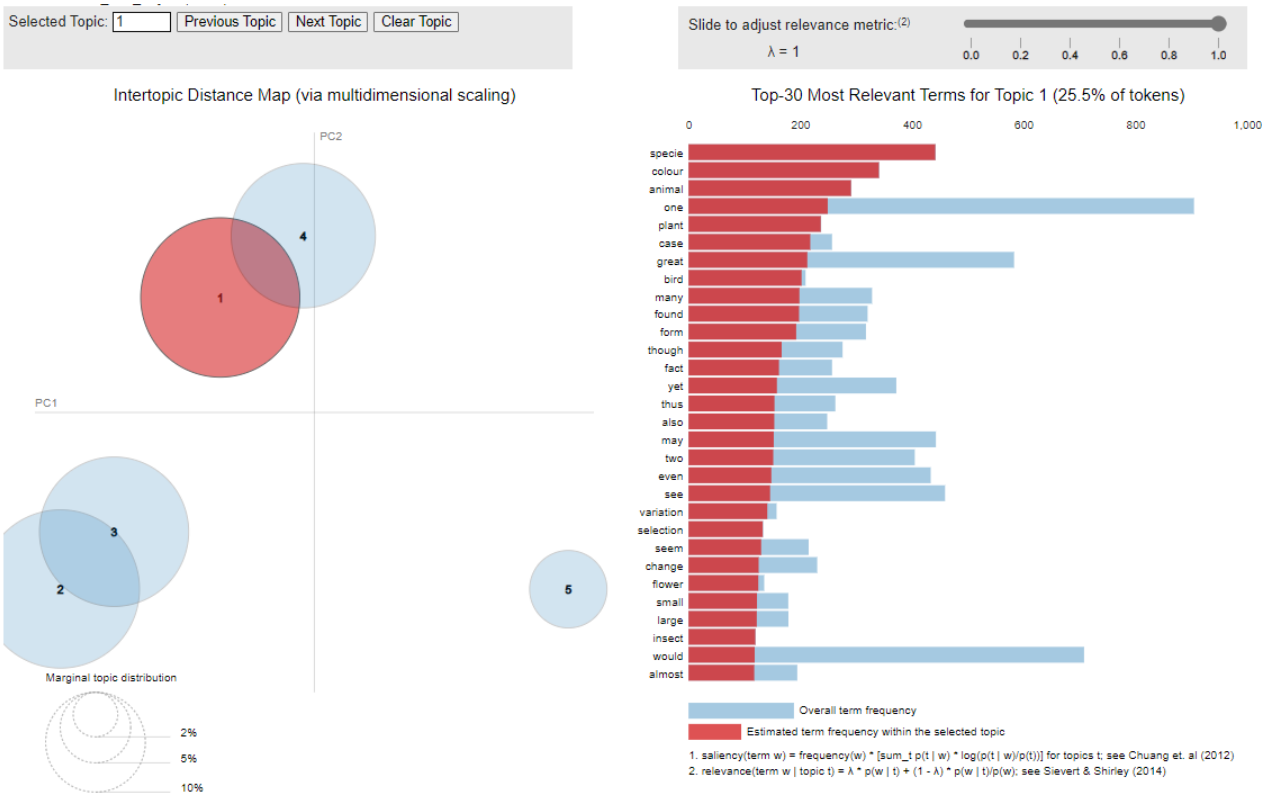
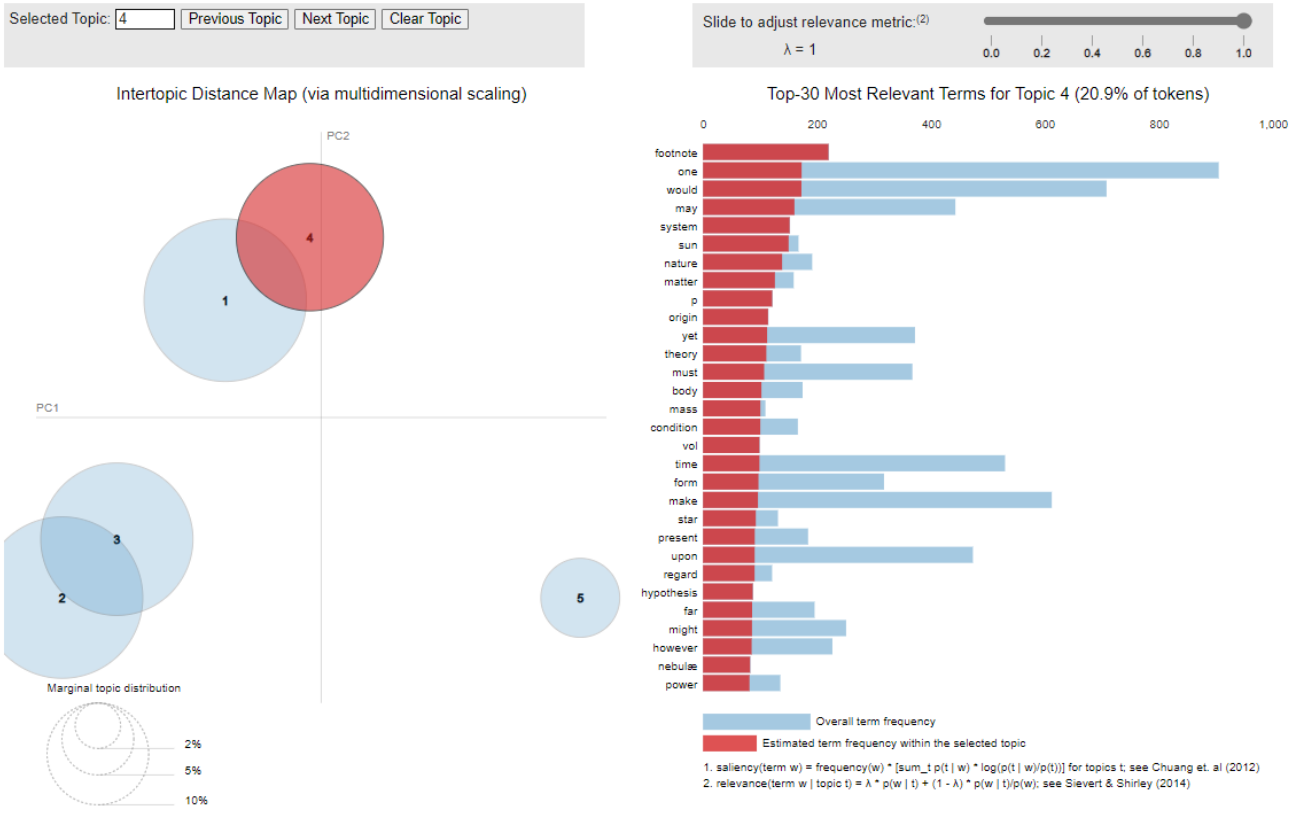


Fig41: maximum Kappa & Silhouette Scores

Error Analysis

As we noticed from LDA Visualization, there is interaction words in topics 1&4 and topics 2&3.





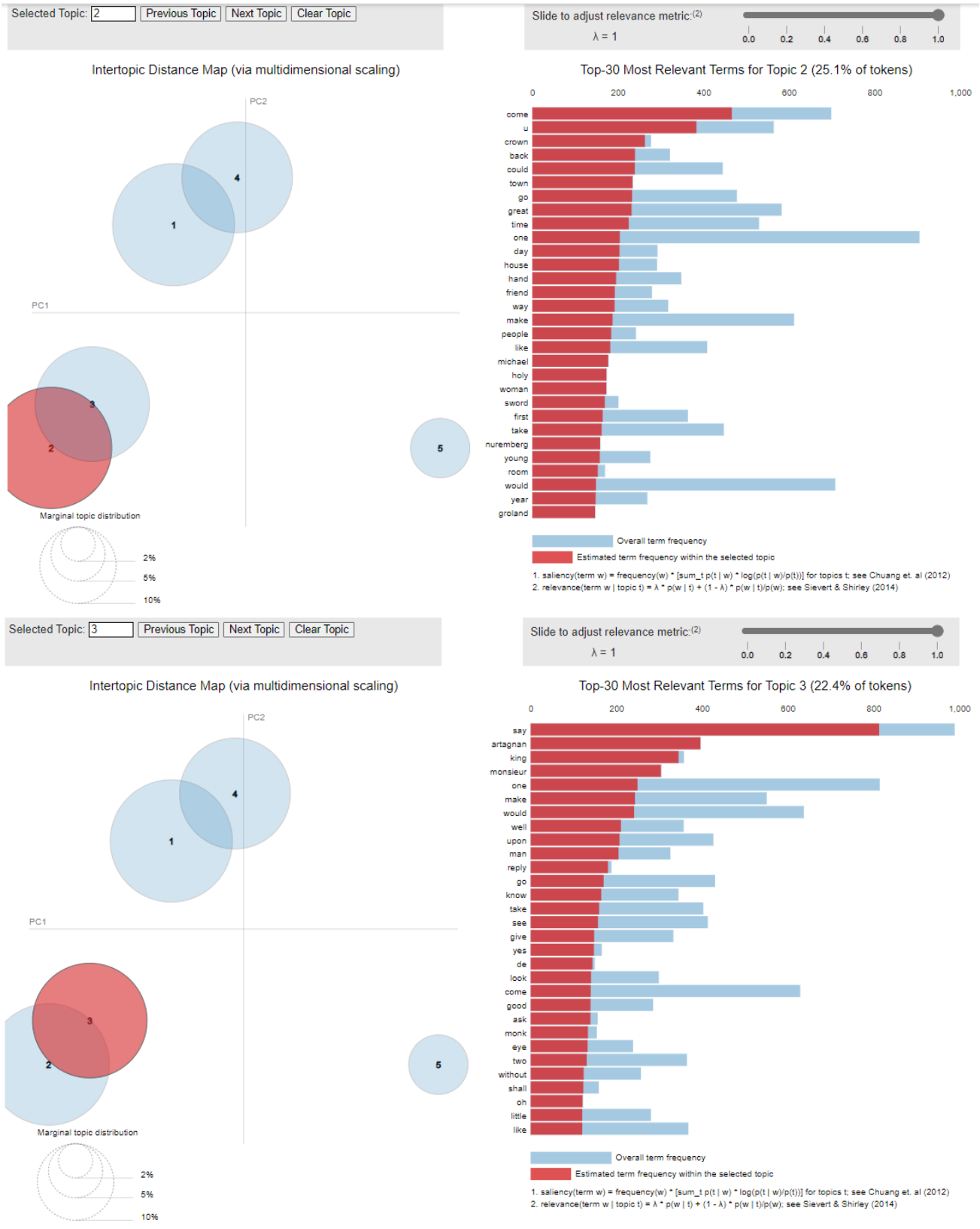


Fig42: LDA Visualization

We displayed the top 10 frequent words for each topic to identity what were the characteristics of the instance records that threw the machine off.

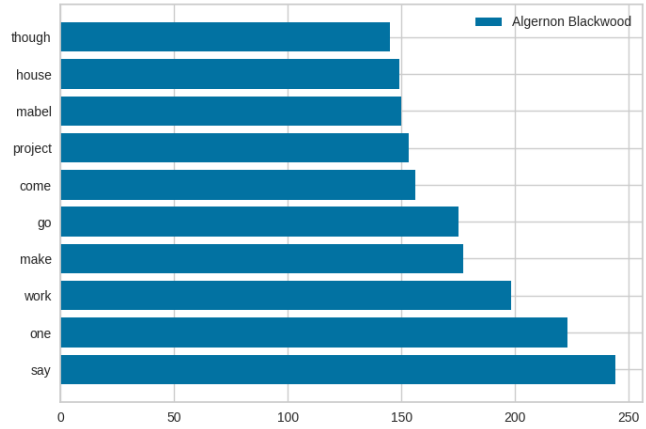


Fig43: Top 10 words in topic 1

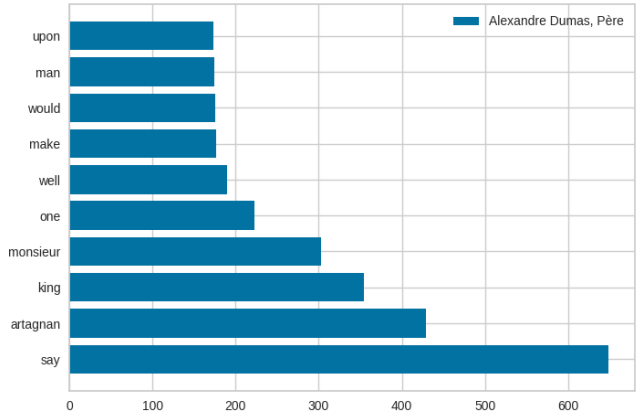


Fig44: Top 10 words in topic 4

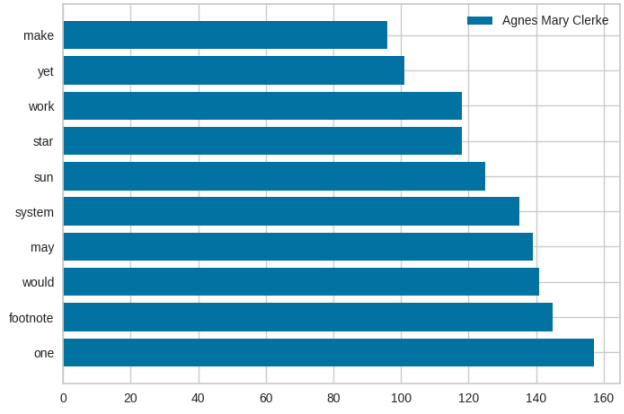


Fig45: Top 10 words in topic 2

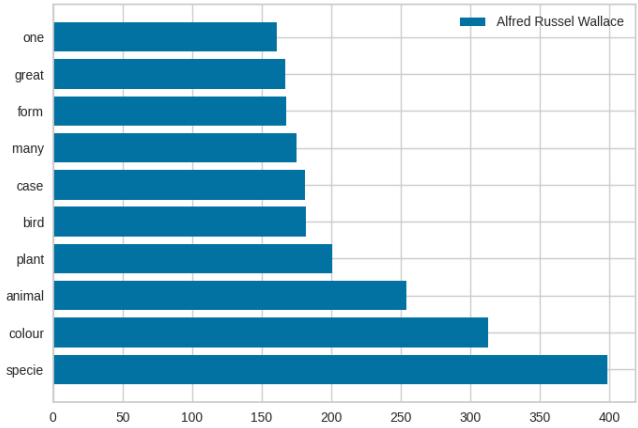


Fig46: Top 10 words in topic 3

As we in fig43 and fig44, the word say occurs in both topics 1&4, and the word one occurs commonly in both topics 2&3, and this make our algorithms may be confused.

We also provide our reasons for using TSNE instead of the PCA and the Data without any feature reductions. We train our models (K-means - EM - Hierarchical algorithm) on (BOW- TF-IDF -LDA- word Embedding).
In this section we took samples from the K-means (the Champion model). We used those (BOW- TF-IDF -LDA- word Embedding) transformations for all partitions of our books and in the first we didn't used any feature reductions or feature selection to see what will happen if put the data like raw and got these results.

- K-means with BOW (without transformation):

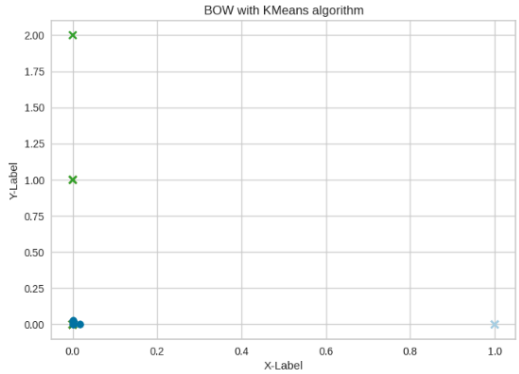


Fig47: Without transformation

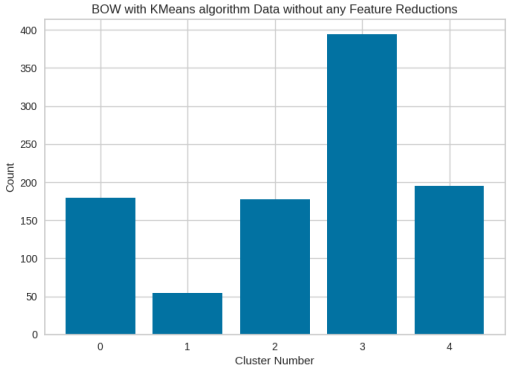


Fig48: Without transformation

The visualization of BOW didn't show anything, because the high dimensionality of data. And in the histogram K-means cluster 1 has only 50 words that is incorrect.

- K-means with TF-IDF (without transformation):

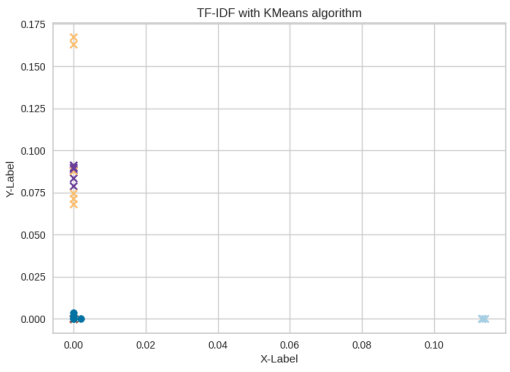


Fig49: Without transformation

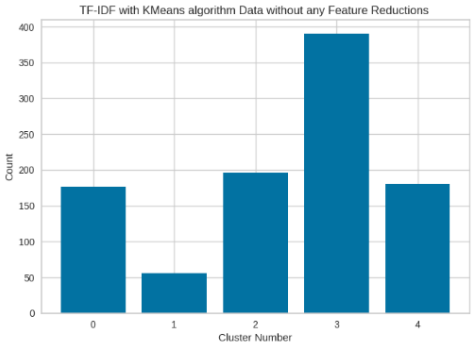


Fig50: Without transformation

- K-means with LDA (without transformation):

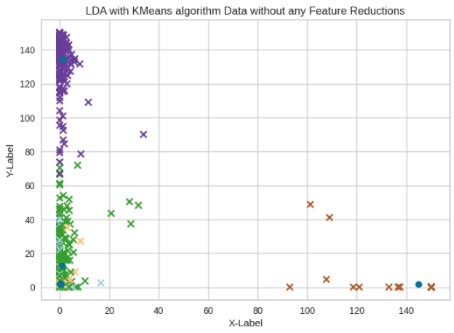


Fig51: Without transformation

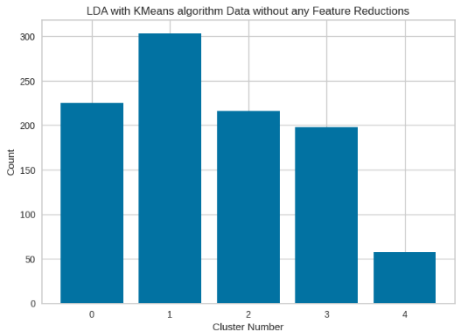


Fig52: Without transformation

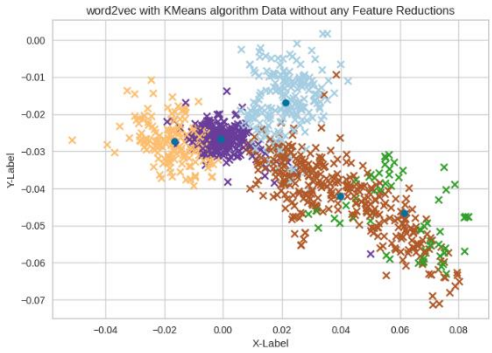


Fig52: Without transformation

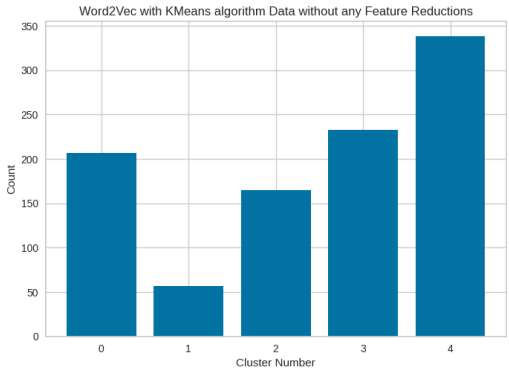


Fig53: Without transformation

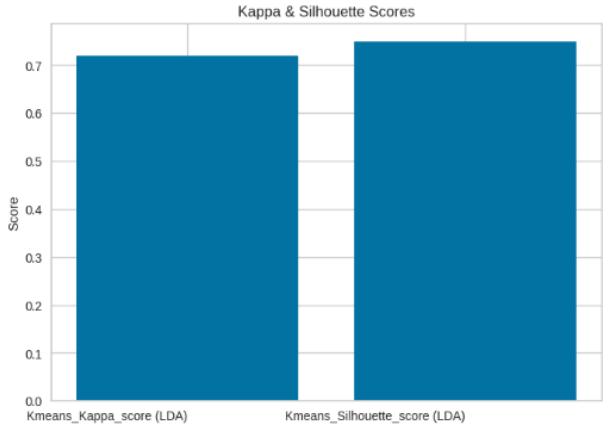


Fig54: scores of Kappa for K-means (Without transformation)

After we saw these results and all the cluster algorithms has similar form and low silhouette and kappa.

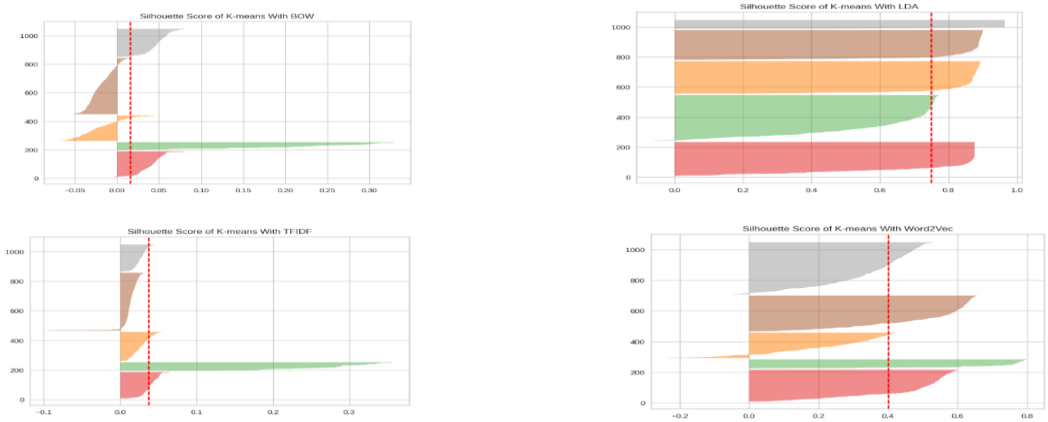


Fig55: scores of Silhouette for K-means (Without transformation)

- K-means with BOW (with PCA transformation):

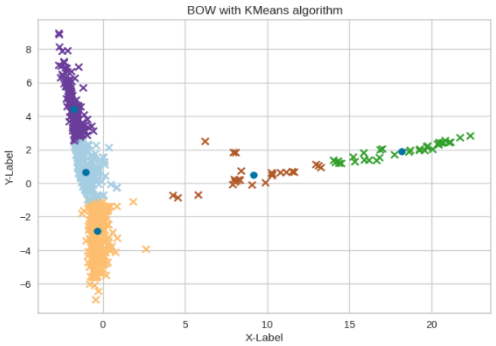


Fig56: With PCA

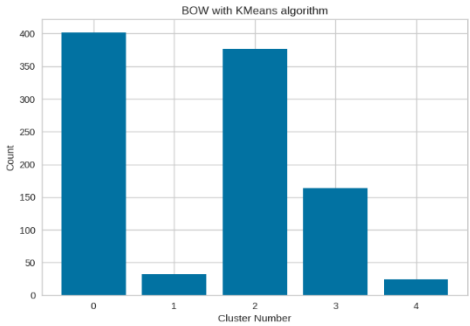


Fig57: With PCA

K-means cluster 1 and cluster 4 had fewer words about 20 words which seems incorrect.

- K-means with TF-IDF (with PCA transformation):

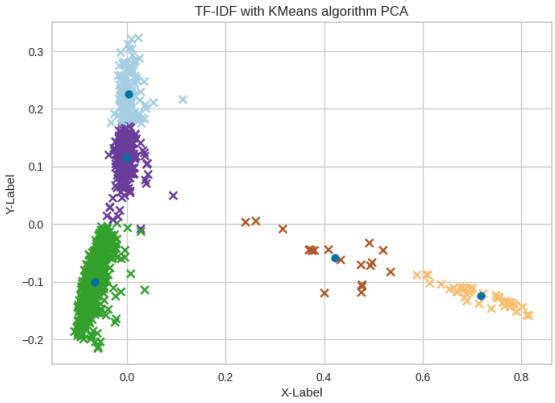


Fig58: With PCA

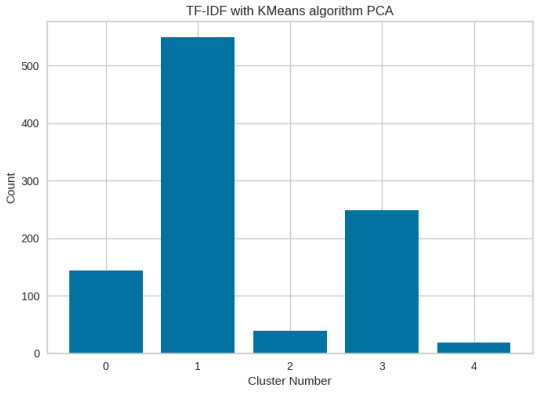


Fig59: With PCA

- K-means with LDA (with PCA transformation):

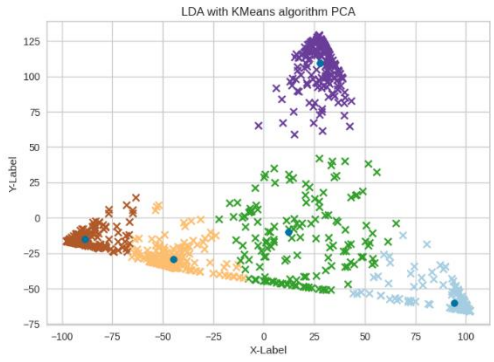


Fig60: With PCA

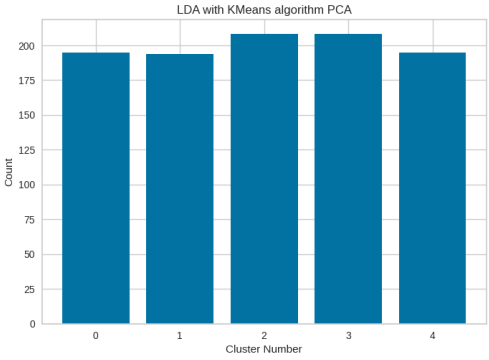


Fig61: With PCA

- K-means with Word2Vec (with PCA transformation):

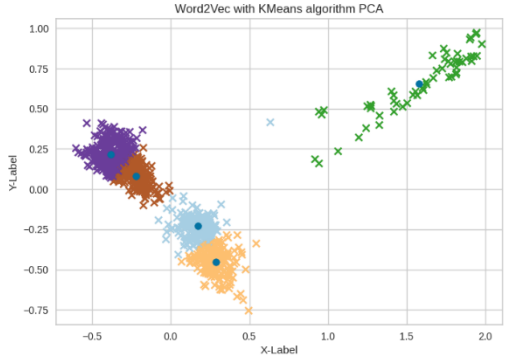


Fig62: With PCA

Fig63: With PCA

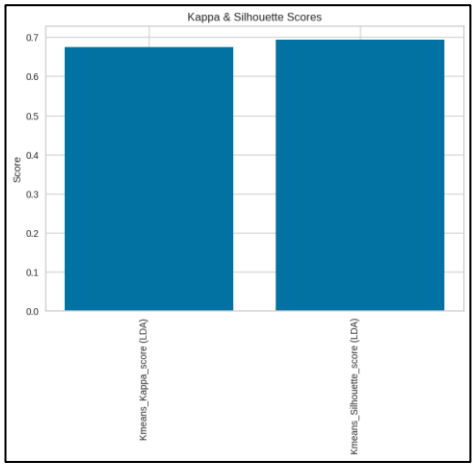


Fig64: scores of Kappa for K-means (With PCA transformation)

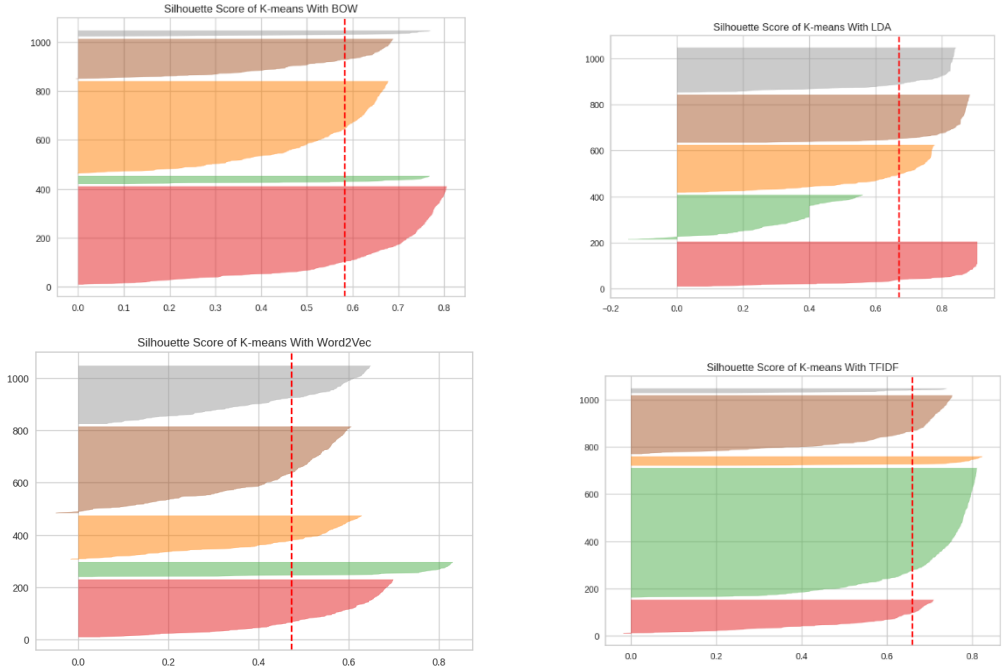


Fig65: scores of Silhouette for K-means (Without transformation)

These results were not bad and not good but in visualization was a little bad and we couldn't see clusters very well but when we compare this result from PCA with the form Data without any feature reductions, we could say there was an improvement by using the PCA but this not our target.

After using the TSNE, the result and the visualization gets better and we decided to use as the default feature reduction.

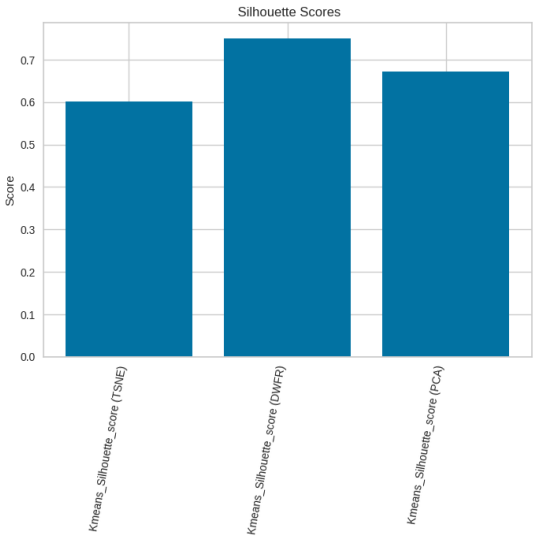


Fig66: scores of Silhouette

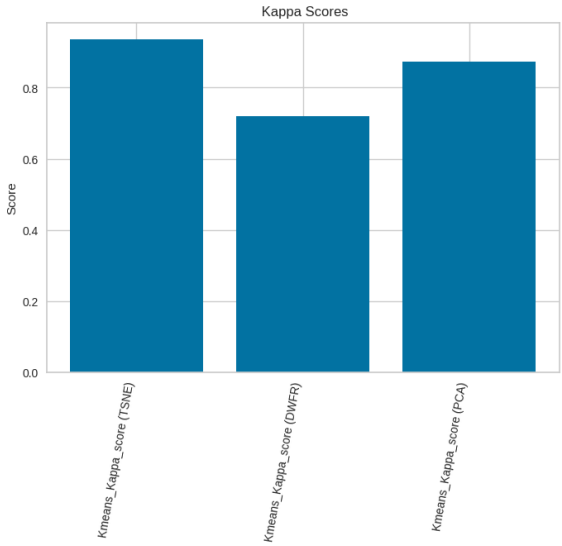


Fig67: scores of Kappa

The results from K-mean on TSNE, DWFR (Data without feature reduction) and PCA, the best average in two silhouette and kappa is the TSNE and this we choose it.

Conclusion

Our report shows that unsupervised learning algorithm like K-Means is effective for text clustering problem. Using text transformation, for example BOW, TF-IDF, LDA and Word2Vec can help to reduce the dimensionality of the data, remove noise and irrelevant words, and capture the semantic meaning of the data more accurately can be easily used by machine learning algorithms. Dimensionality reduction helps to simplify the model, reduce computation time, avoid overfitting, and improve model performance.

References

1- https://scikitlearn.org/stable/modules/generated/sklearn cluster.KMeans.html

2- https://scikitlearn.org/stable/modules/generated/sklearn.mixtur e.Gaussianixture.html

3- https://www.geeksforgeeks.org/dimensionality-reduction/

4- https://en.wikipedia.org/wiki/Topic_model

5- https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0

6- https://nlml.github.io/in-raw-numpy/in-raw-numpy-t-sne/

7- https://scikitlearn.org/stable/modules/generated/sklearn.featue extraction.text.TfidfVectorizer.html#sklearn.feature_extractiont TfidfVectorizer