Text Clustering Assignment #2

Report Submitted to Prof. Arya Rahgozar

GNG5125

by

Lasya Bhaskara

Tulika Shukla

Xun Xun Shi

February 28, 2021

Department of Engineering

University of Ottawa

Ottawa, Canada

Table of Contents

Table of Figures

Introduction

This report contains a solution towards clustering for authorship. Given a set of a data that came from five different books, the goal is to use clustering algorithm to group passages together. Because the annotated labels were taken out initially, this analysis looks at a series of unsupervised machine learning algorithms, including K-Means, hierarchical clustering, and expectation maximization, along with three preprocessing steps such as Bag of words (BOW), TF-IDF and Word2Vec (W2V). Principle Component Analysis (PCA) was also implemented during the preprocessing step with certain feature engineering methods. Lastly, topic modelling techniques including LDA was used to determine the clustering performance.

Data

We have chosen a range of books that were published between the 2000s the 1600s. We hypothesize that the books written in the same time frame will likely to be clustered together, and the features will be more similar.

The following books includes:

1. The Man Who Was Thursday - G. K. Chesterton (1908)/ Chesterton-Thursday
2. Melville – Moby Dick
3. Stories – E. Bryant (1900-2000) / Bryant-stories
4. Paradise Lost – John Milton (1667) / Militon-Paradise
5. Hamlet – William Shakespeare (1609) /Shakespeare-Hamlet

Data Preparation for Book Partition Dataset:

To have a balanced data set, we have taken 200 passages containing 150 words from each book. Five different books from Gutenberg's digital library were selected to create the dataset. A function was developed to return the partitioned book as a data frame with 200 rows, each row containing 150 words. A total of 1000 records consisting of 200 passages from each book were generated.

	Unnamed: 0	passage	bookName
0	72	poet ; he was really a poet who had become a d...	chesterton-thursday
1	172	that when we broke up rather hurriedly the who...	chesterton-thursday
2	148	as to send a lame man ? `` He set off at a sma...	chesterton-thursday
3	10	, Gregory , upon the whole nature of poetry	chesterton-thursday
4	156	threading . He came nearer and nearer , the la...	chesterton-thursday
...	...	...	...
995	115	unexampled love , Love no where to be found le...	milton-paradise
996	355	then returned at shut of evening flowers . But...	milton-paradise
997	118	, to be deemed A God , leaped fondly into Aetn...	milton-paradise
998	11	his malice served but to bring forth Infinite ...	milton-paradise
999	434	the mouth of Hell For ever , and seal up his r...	milton-paradise

1000 rows × 3 columns

Figure 1. Data Preparation.

Data Preprocessing and Data Cleansing:

Spacy

Our choice of Spacy library as an open-source library for preprocessing was motivated by the fact that it can perform tokenization in 0.2 milliseconds compared to NLTK's 4 milliseconds. It creates pipelines which can automatically perform the tokenization including tagging, parsing, named entity recognition etc.

Motivation for using lemmatization and avoiding stop words.

Getting rid of the punctuations and performing Lemmatization on the text data was done using the Spacy library.

As a first step, lemmatization was performed. It is a process of converting many different forms to its root word. For example, words such as "he", "she", "we" will all be converted to the word "pronoun". The NLP pipeline created using Spacy also automates the process of tokenization, parsing and tagging. The code was executed with and without this preprocessing step and the accuracy has improved with preprocessing.

Stop words were first not removed during preprocess as there could be important discriminatory features. Since our data contains both past century books such as Shakespeare's Hamlet and Milton (written in the 1600s), the common language is different than modern books. These stop words could serve as key feature to help distinguish these differences from modern books in this data set such as Bryant's stories, which was written in the 1900s. Removing them could potentially reduce the cluster quality. In our case, we did look at other feature engineering steps (such as TF-IDF) that will remove common words based on the frequency instead.

Feature Engineering:

Feature engineering is the process of converting raw textual data into numerical vectors. Bag of Words (BoW) and Term frequency – Inverse document frequency (TF-IDF) were the statistical measures used in order to perform text feature extraction on the book passages. Word2Vec use a machine learning algorithm to perform vectorization. Lastly, PCA was selectively used to reduce the feature dimension. [1]

TF-IDF:

As a first choice, TF-IDF was used to transform X (Passages) into vectors. TF-IDF is a vectorization algorithm which is used to represent textual data in numerical vectors. It gives weights to the words depending on their frequency. Compared to bag of words, TF-IDF also calculates the inverse document frequency, which will factor in the frequency of the word to occur in all documents. This will take out very commonly used words. [1]

Bag of Words:

Bag of words is a form of vectorization process for extracting features from the textual data. Like TF-IDF, this is another step of feature processing, but it does not consider how often the words will appear across different documents. The term 'bag' implies that the order of the words appearing in the document is ignored. The model is only concerned with whether the word is present in a document, not with the location of the word in the document. [1]

Word2Vec:

Word2Vec is vectorization method, where the model is first trained on a neural network from a large text corpus. During the training process, the words are represented as a vector and projected onto each other through a cosine similarity. The value of these similarity scores indicates the semantics between each word. In our model, we used the Spacy pretrained model, which is projected onto 300 features of space. [1]

Principal Component Analysis:

Principal component analysis (PCA) is a algorithm that allows for reducing dimensionality. This algorithm determines a hyperplane and projects the data on it. Such algorithm sorts the components based on variance. The first component has the highest variance, and the last component has the least [1] .

Upon the feature engineering step, the features size for TFIDF and BOW are huge, and these unsupervised ML algorithms may not scale properly. To overcome these limitations, we applied PCA to reduce the dimensionality of over 12,000 features to only 300 features. We compared the results of K-means with TFIDF with and without PCA to justify the choice of using PCA on the rest of the algorithms. This step was omitted for Word2Vec as this feature engineering produced a vector with low dimensionality to begin with (300 features).

Modelling

The models:

Unsupervised Machine Learning:

Unsupervised learning is a machine learning technique in which the model draws similarities within the data on its own, given that the data is unlabeled i.e., no human intervention is needed to train the model. It helps in identifying unknown information that was not detected previously and performs complex processing tasks better than supervised learning algorithms.

K-Means

K-Means is a clustering technique that separates the data into the required number of clusters of equal variances. It tries to minimize a criterion called Inertia which is the within cluster variation of the data points. K means usually works best with large sample sizes. The mean of all the data points within a cluster is called the centroid. The centroids are chosen in such a way that the inertia is minimized, and the data point is representing the exact center of the cluster.

In a high-level implementation of K-means, random data points are chosen as the centroids for the clusters. The other data points are assigned to the nearest cluster based on these centroids. Then the centroids are iteratively adjusted by calculating the average of all the data points within the cluster. This is done until the centroid has stabilized or until the defined number of iterations are completed. [2]

Hierarchical Clustering

Hierarchical clustering is a technique that outputs a hierarchy, a structure that is more informative. It does not require us to prespecify the number of clusters and the most hierarchical algorithms that have been used are deterministic.

Hierarchical clustering algorithms are either top-down or bottom-up. Bottom-up algorithms treat each document as a singleton cluster at the outset and then successively merge (or agglomerate) pairs of clusters until all clusters have been merged into a single cluster that contains all documents. Bottom-up hierarchical clustering is therefore called agglomerative hierarchical clustering. [3]

In Hierarchical clustering, all objects start as singletons or an individual cluster. They are then merged using one of the following linkage methods:

- Single Linkage

- Complete Linkage
- Centroid Linkage
- Ward's Linkage
- Average Linkage

The linkage methods work by calculating the distances or similarities between all objects. Then the closest pair of clusters are combined into a single cluster, reducing the number of clusters remaining. The process is then repeated until there is only a single cluster left. [3]

Single Linkage

For the Single linkage, two clusters with the closest minimum distance are merged. This process repeats until there is only a single cluster left.

Complete Linkage

For the Complete linkage, two clusters with the closest maximum distance are merged. This process repeats until there is only a single cluster left.

Centroid Linkage

For the Centroid linkage, two clusters with the lowest centroid distance are merged. This process repeats until there is only a single cluster left.

Ward's Linkage

For Ward's linkage, two clusters are merged based on their error sum of square (ESS) values. The two clusters with the lowest ESS are merged. This process repeats until there is only a single cluster left.

Average Linkage

The Average linkage method uses the average pair-wise proximity among all pairs of objects in different clusters. Clusters are merged based on their lowest average distances.

Gaussian Mixtures and Expectation Maximization

A gaussian mixture model assumes that the data is generated from K number of gaussian distributions with unknown weights, where each individual distribution can be thought of as an ellipsoid shaped cluster of various size, density, orientation etc. The expectation maximization algorithm aims to optimize for each cluster's weight and the assignment of data to the corresponding cluster. Through this iterative process, the algorithm generalizes the data and assign it to a corresponding cluster. [2]

LDA

Latent Dirichlet Allocation is one of the most popular topic modelling technique. The generic idea of topic modelling is to automatically draw out the topics from a corpus of documents. LDA provides a model that can identify how all the documents in a corpus were created i.e., how the document has the words that it contains. The emphasis of the model is that each document has a distribution of multiple topics in different proportions. Additionally, each word in a document

is recognized as being drawn from one of the topics as a proportion to document's distribution over topics. Also, the order of the words in document is not important just like in a bag of words technique. [3]

Generally, LDA method is used as a feature engineering step, but it can also be used as an unsupervised machine learning algorithm. We have used LDA in the context of topics modelling, and directly as an unsupervised machine learning algorithm.

Parameter Selection & Unsupervised Results

Coherence Score

Topic Coherence measures score a single topic by measuring the degree of semantic similarity between high scoring words in the topic. These measurements help distinguish between topics that are semantically interpretable topics and topics that are artifacts of statistical inference. A set of statements or facts is said to be coherent if they support each other. Thus, a coherent fact set can be interpreted in a context that covers all or most of the facts. [3]

In this analysis, coherence score was used to evaluate the effect of topics numbers used in LDA analysis with BOW and TFIDF.

Elbow method

During the clustering process, as the number of clusters increase, the variation of the different data points within a cluster to its centroid decreases. It implies that when the number of clusters is equal to the total data points in the given data set, then the variation is zero since each data point has zero variance with itself. Through elbow method, we try to find an optimal number of clusters for which the sum of square distances of data points within cluster have stabilized and no longer shows a notable change. In other words, if the plot represented looks like an arm, then the value of k at the elbow represents the optimal number of clusters to generate. [2]

Silhouette score

Silhouette score or silhouette coefficient is used to measure the accuracy and determine how good a clustering technique is. The score ranges from a –1 to +1. Analyzing the silhouette coefficients helps us in studying the distance between the different clusters and see how far away the data points in one cluster are from the data points in the neighboring cluster. A value of 1 indicates that the data point is far away from the nearest cluster and is correctly assigned to its cluster. A value of 0 indicates that a very narrow boundary exists for the data point to belong to either one of the neighboring clusters. A negative value indicates that the data point is wrongly assigned to its current cluster. We can also decide on the optimal number of clusters by using the silhouette score. [2]

Dendrogram method

A dendrogram is a type of tree diagram showing hierarchical clustering — relationships between similar sets of data. We have the samples of the dataset on the x-axis and the distance on the y-

axis. Whenever two clusters are merged, we will join them in the dendrogram, and the height of the join will be the distance between these points. [2]

Bayesian Information Criterion and Akaike Information Criterion

Literature have suggested that for Gaussian mixtures, silhouette score is not as reliable due to the lack of spherical shape resulted from GM models. Instead, theoretical information criterion was recommended as a minimalization metric to determine the optimal cluster number. This includes Bayesian information criterion (BIC) and Akaike Information criterion. [2]

LDA

TFIDF and BOW

LDA can be used for topic modelling. In this case, LDA was used on both BOW features and TFIDF features, and the coherence score was reported for the selection of different topics (k).

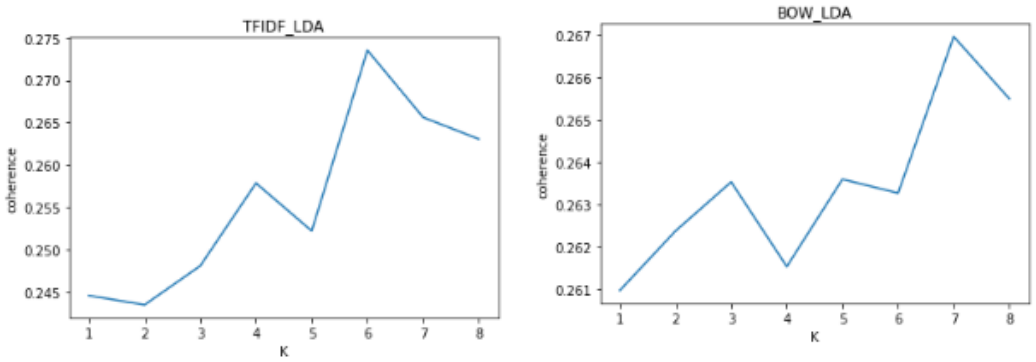

Figure 2. Coherence Score for Different Topic Numbers. Left: LDA With Tfidf; Right: LDA With Bow.

The overall coherence score of both TF-IDF and BOW with LDA is around the similar range, between 0.24 to 0.275. There is a greater discrepancy of the coherence score with the change of topics for TFIDF compared to with BOW.

From TF-IDF used with LDA, the optimum coherence score is when K=6, indicating that the optimum topic number is six . When BOW is used with LDA however, the highest coherence score is seven, indicating that seven topics should be used for best cluster. Knowing that the empirical number of clusters should be five, as the dataset was taken from five different books, the results of TFIDF with LDA may provide a better representation of the data nature. Furthermore, the coherence score at five cluster is also higher for TFIDF. This coherence score analysis provides us with insight that TF-IDF should be a better feature engineering compared to BOW for this dataset. We hypothesize that this behavior will be consistent for other algorithms as well.

K-means

TFIDF with and without PCA

Selection of Cluster Number

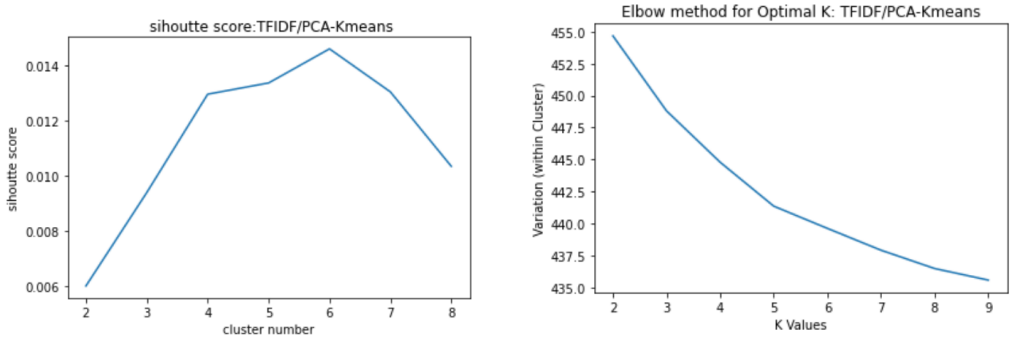

Figure 3. TFIDF/PCA with K-means method's Performance with cluster number. Left – Silhouette score with TF-IDF/PCA. Right – Elbow method with TF-IDF/PCA

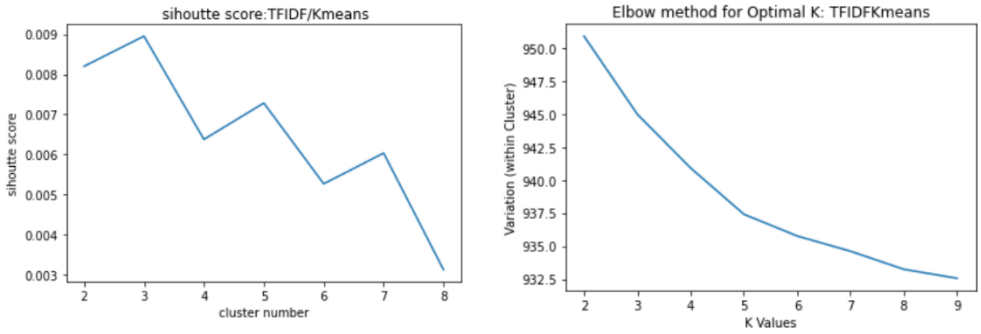

Figure 4. TFIDF with K-means method's Performance with cluster number. Left – Silhouette score with TF-IDF without PCA. Right – Elbow method with TF-IDF without PCA.

From the silhouette score graphs, at the required number of clusters k = 5, we can see that silhoutte score is higher with tfidf-PCA at nearly 0.013 when compared to tfidf-without pca at 0.007. With PCA feature reduction, all the noisy features are made redundant which could have increased the distances between data points of different clusters and thereby increased the silhoutte score. In the absence of PCA, with increase in the number of clusters, the silhoutte score is dropping.

Also, the silhouette score doesn't give consistent results for different k values, as we see that the optimal k = 6 for TFIDF with pca and k=3 for TFIDF without pca. Since the score ranges are very close, the differences are not really significant. Hence, we can use the elbow method as a complement to justify choosing k =5.

In both approaches of K means TFIDF (with and without PCA), the elbow method gives consistent results. The value of K, for which the variation within the clusters is minimized i.e the coherence within the cluster is maximized, is an ideal value for k. We can see from the above graphs that the variation drastically dropped and stabilized when k = 5. Hoewever, we already know that we need to divide the input passages from the dataset into 5 different clusters since we are aware that the passages are from five different books. The elbow method demonstrated above simply justified our value for k.

The above analysis justifies using PCA for further analysis with the remaining models.

Clustering results with five centers

Silhouette score and feature 1 vs 0 of PCA-TFIDF

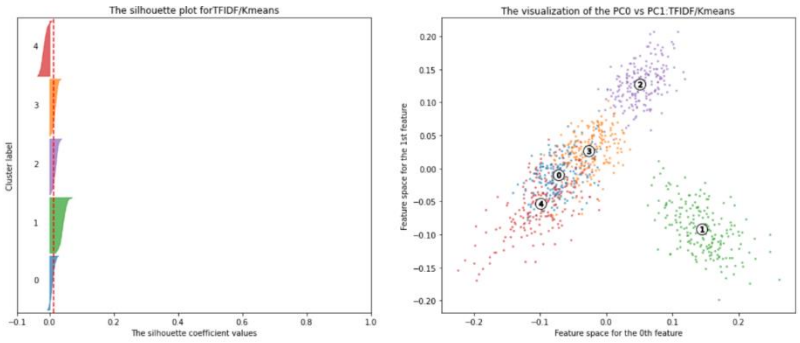

Figure 5. Silhouette score and PCA analysis of feature 1 vs 0 of PCA-TFIDF with K-means(K=5).

The red dotted line shows the average silhouette score for all the clusters as around 0.013 at k = 5. The green cluster seems to be the best one with the highest score, and it can likely be Shakespeare because of the writing style of 16th century that makes it unique.

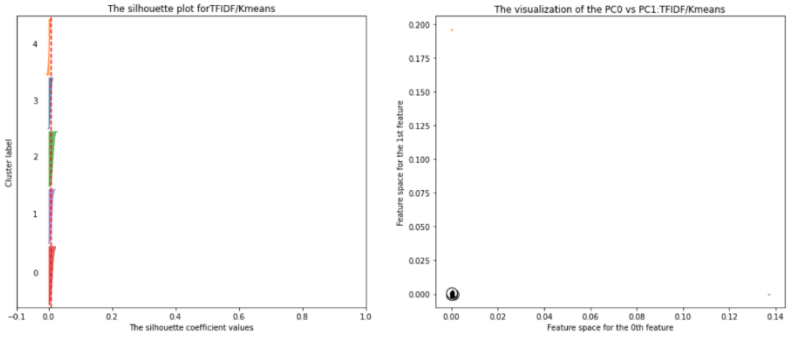

Figure 6. Silhouette score and feature 1 vs 0 of TF-IDF with K-means(K=5).

The average silhouette score with TFIDF and no PCA is around 0.0075. This score is less than the score of TFIDF with PCA. Since there is no feature reduction in this case, it is very difficult to visualize since there are 12,019 features. The feature space representation is very insignificant that the clusters are represented at around 0.00 and 0.000.

BOW with PCA

Selection of Cluster Number

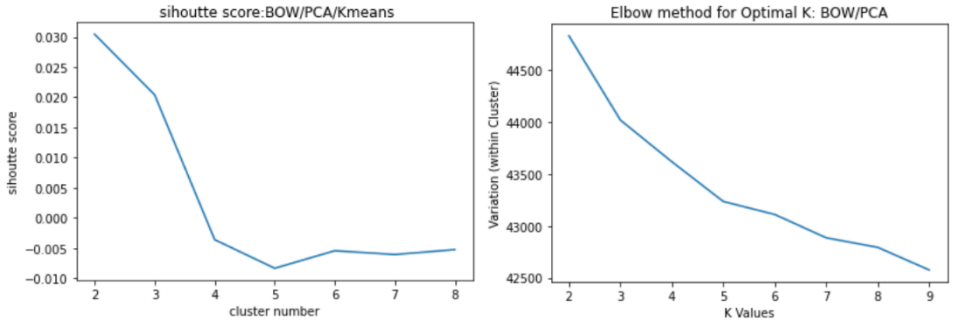

Figure 7. BOW/PCA with K-means method's Performance with cluster number. Left – Silhouette score with BOW/PCA. Right – Elbow method with BOW/PCA

The optimal number of clusters according to the elbow method, is either k = 5 or 6.

The optimal number of clusters according to silhouette is 2. This means that BOW features are most useful at discriminating 2 clusters and that the cluster quality decreases thereafter with increase in k. With prior knowledge that there are 5 clusters, this shows that BOW features are harder to discriminate between certain passages. The hypothesis of 2 clusters is due to styles of writing of the old vs new age.

Predictions with Cluster of 5

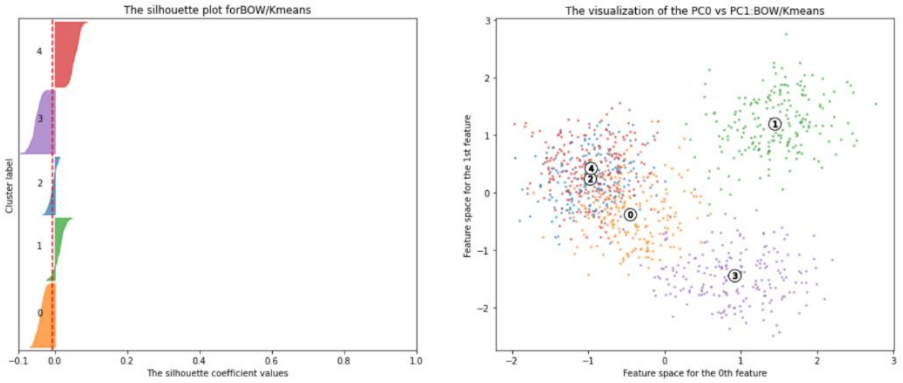

Figure 8. Silhouette score and feature 1 vs 0 of BOW with K-means(K=5).

The average silhouette score from the above plot is: -0.0073.

The negative value indicates that data points are wrongly identified as belonging to a cluster.

Word 2 Vec

Selection of Cluster Number

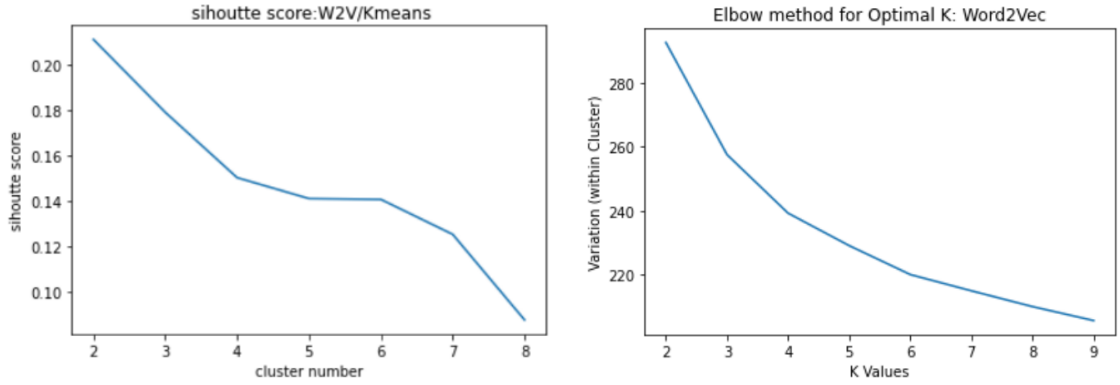

Figure 9. Word2Vec with K-means method's Performance with cluster number. Left – Silhouette score with W2V/Kmeans. Right – Elbow method with Word2Vec

The optimal number of clusters according to the elbow method, is at k = 5 or 6 which looks like the elbow.

The optimal number of clusters according to silhouette is 2. This means that Word2Vec features are most useful at discriminating 2 clusters and that the cluster quality decreases thereafter with increase in k. With prior knowledge that there are 5 clusters, this shows that Word2Vec features are harder to discriminate between certain passages. The hypothesis of 2 clusters is due to styles of writing of the old vs new age.

Predictions with Cluster of 5

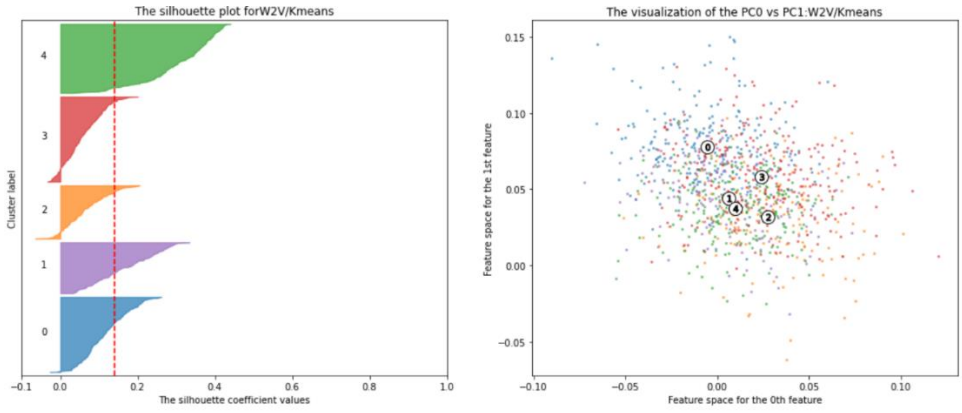

Figure 10. Silhouette score and feature 1 vs 0 of W2V/K-means.

The average silhouette score from the above plot is: 0.14.

The near zero value indicates that there is a chance for data points of wrongly identified as belonging to a cluster.

Hierarchical Clustering

TFIDF with PCA

Dendrograms

A dendrogram is a tree-like diagram that records the sequences of merges or splits.

At the bottom, we start with data points, each assigned to separate clusters. Two closest clusters are then merged till we have just one cluster at the top. The height in the dendrogram at which two clusters are merged represents the distance between two clusters in the data space.

The best choice of the number of clusters is when the number of vertical lines in the dendrogram cut by a horizontal line that can transverse the maximum distance vertically without intersecting a cluster. Here, we have a prior knowledge of the number of clusters.

We make use of truncated dendrogram as the dendrogram can be hard to read when the original observation matrix from which the linkage is derived is large. Hence, we use truncation to condense the dendrogram.

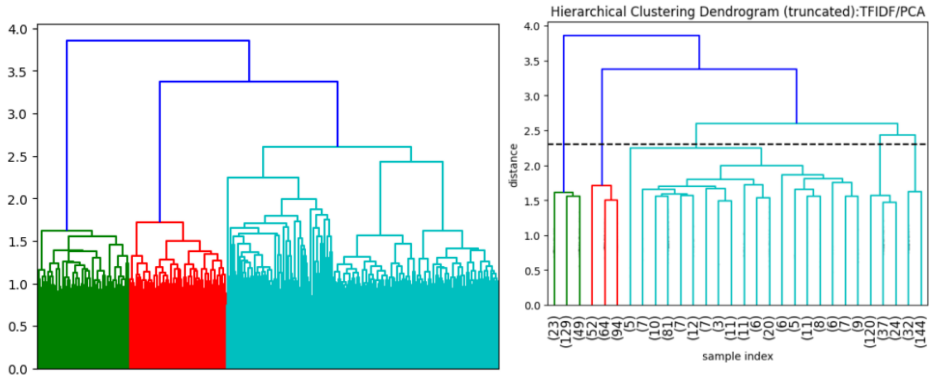

Figure 11. Dendrogram for TF-IDF/PCA.

This dendrogram looks disperse, but the size of the group is similar to others. Taking a look at the dendrogram, approximately all the groups has 200 books.

BOW with PCA

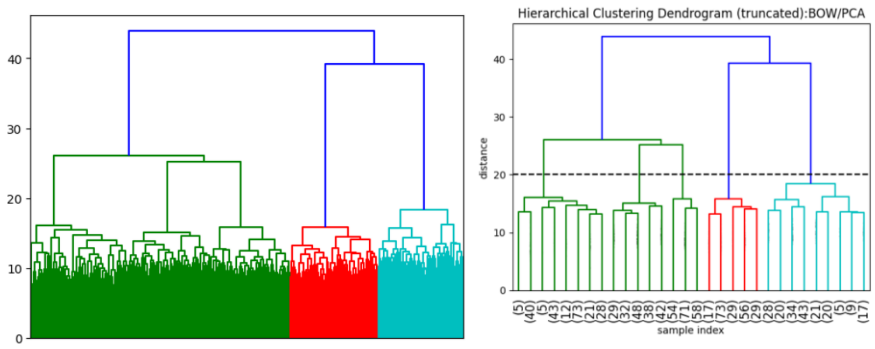

Figure 12. Dendrogram for BOW/PCA.

Looking at our dendrogram for TF-IDF and BOW, we can see the difference in the distance on the y axis is different for both the methods.

For TF-IDF the horizontal line cuts five clusters at a distance of ~ 2.3 whereas for BOW it cuts the horizontal line at ~ 20.

Word 2 Vec

Dendrograms

Similarly, for Word 2 Vec we can see the clusters look more dispersed but have similar data size and the distance at which we get 5 clusters is 3.9 which is closer to the results of TF-IDF more than the BOW inference.

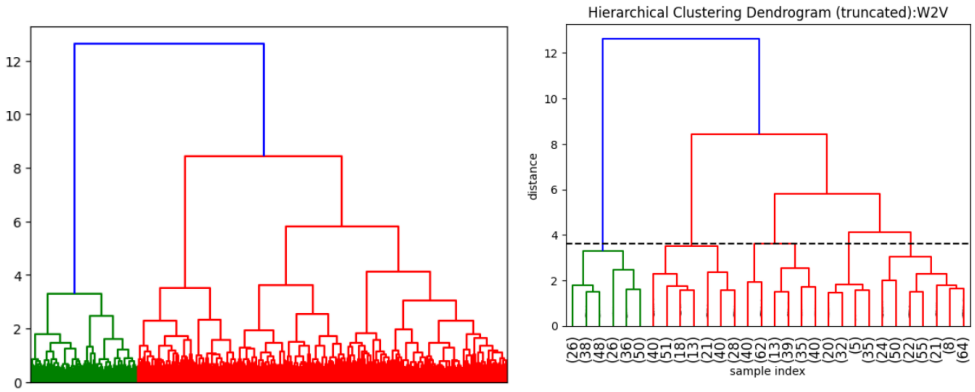

Figure 13 : Dendrogram for W2V/PCA.

Gaussian Mixture

TFIDF with PCA

The figure below demonstrates the effect of the cluster number on cluster's qualities as a few different metrics for TFIDF-PCA with Gaussian Mixtures. On the left is BIC and AIC score vs cluster number, on the right is the silhouette score vs cluster number.

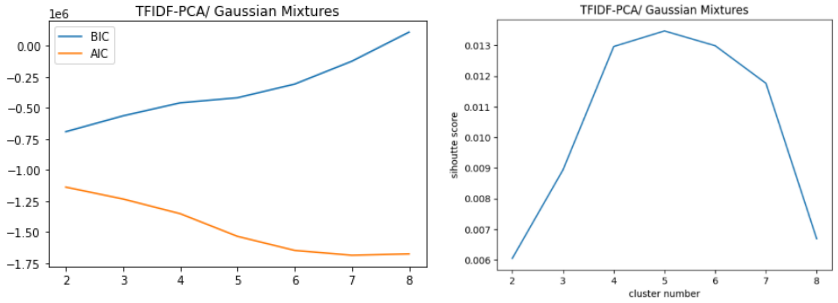

Figure 14. TFIDF/PCA with EM method's Performance with cluster number. Left: BIC and AIC score vs cluster number. Right. silhouette score vs cluster number.

The graph of BIC and AIC vs cluster number have some coinciding results, where with more clusters the AIC score decreases but BIC score increases. However, one thing to point out is that after the sixth cluster the AIC score is stabilized with the increase of cluster number. On the other hand, after the fifth cluster, the BIC score drastically increases with the increase of cluster number. Looking at this graph, the optimum cluster number is approximately between 5-6 clusters, which is a good point for tradeoff between BIC and AIC score. This cluster number coincides with the prior knowledge of 5 books in the dataset and demonstrates that TF-IDF is a good feature engineering fit for the expectation maximization model.

The graph of Silhouette score vs cluster number demonstrate that five clusters is an optimum number, since it is where the score is the closest to 1. This also coincides with the empirical prior knowledge of the five books in the dataset, which validates using TFIDF as a feature engineering model.

BOW with PCA

The figure below demonstrates the effect of the cluster number on cluster's qualities as a few different metrics for BOW-PCA with Gaussian Mixtures. On the left is BIC and AIC score vs cluster number, on the right is the silhouette score vs cluster number.

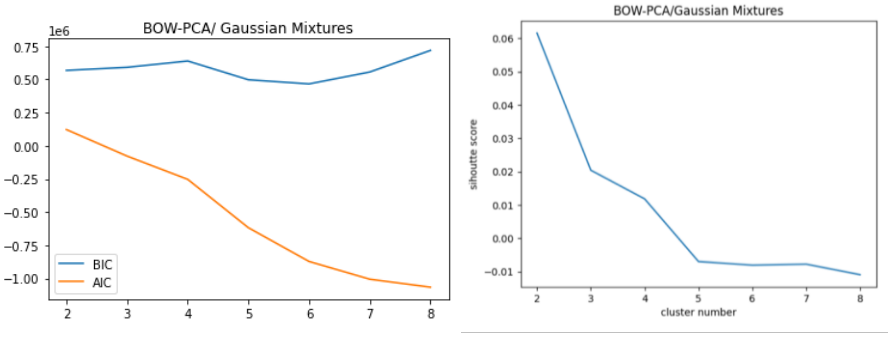

Figure 15. BOW/PCA with EM method's Performance with cluster number. Left: BIC and AIC score vs cluster number. Right. silhouette score vs cluster number.

The graph of BIC and AIC is the same trend as with TFIDF-PCA (for Gaussian Mixtures), whereby increasing cluster number increases the BIC score but decreases the AIC score overall. However, around the sixth cluster, the AIC score seems to have stabilized and the BIC score is also at the lowest. Any cluster before the sixth cluster has a very high AIC score, where as any cluster after the sixth cluster increases the BIC score. The optimum cluster according to the figure should therefore be six, as those are lowest values for both BIC and AIC. This is close to the empirical cluster number of 5.

The graph of Silhouette score vs cluster number demonstrate that two clusters is an optimum number, since the score decreases with more cluster numbers. This contradicts with the empirical prior knowledge of the five books in the dataset, which means that BOW may not be an appropriate choice of feature engineering step for this problem.

Word 2 Vec

The figure below demonstrates the effect of the cluster number on cluster's qualities as a few different metrics for W2V with Gaussian Mixtures. On the left is BIC and AIC score vs cluster number, on the right is the silhouette score vs cluster number.

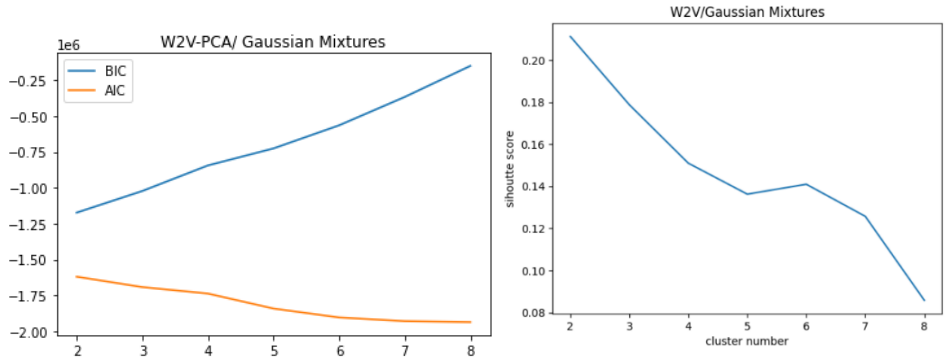

Figure 16. Word2Vec with EM method's Performance with cluster number. Left -BIC and AIC score vs cluster number. Right - silhouette score vs cluster number.

The graph of BIC and AIC is the same trend as with TFIDF-PCA (for Gaussian Mixtures), except BIC score and AIC score have entirely inverse relationship. In general BIC is a more simplistic model than AIC and may not fit to large data as well as AIC. Since the results are entirely contradictory, if a decision must be made, it should be based on the AIC score based on the large dataset. Therefore, the optimum number of clusters following this figure is between 6-8. This does not coincide with the empirical prior knowledge of 5 clusters. Therefore, W2V may be an inappropriate feature engineering selection for this problem.

Similar to BOW, the silhouette score vs cluster number demonstrates that two cluster is also the optimum number of clusters. This contradicts with the prior knowledge and furthers the hypothesis that W2V is not the best choice of feature engineering to be used with GM model on this dataset.

Evaluation of Results Compared to the Annotated Labels

Because this dataset was generated from five different textbook set and was used as a mean of unsupervised ML exploration, the initial labels were available. It is important to acknowledge that this not the case in classic unsupervised ML problem cases. Often, real life dataset will not be included the annotated data labels. In this specific case, the empirical data labels were taken advantage of and used to re-annotate each cluster for comparison.

Metrics

Annotating Clusters

To compare the quality of the clusters, it is important to identify and label them, and compare them to the empirical label. This annotation process is done by matching all the passages in one cluster to its empirical label (book source). Then, the most frequently occurring label becomes the annotation of that cluster. As an example, the following figure represents this process for BOW/PCA-KMeans.

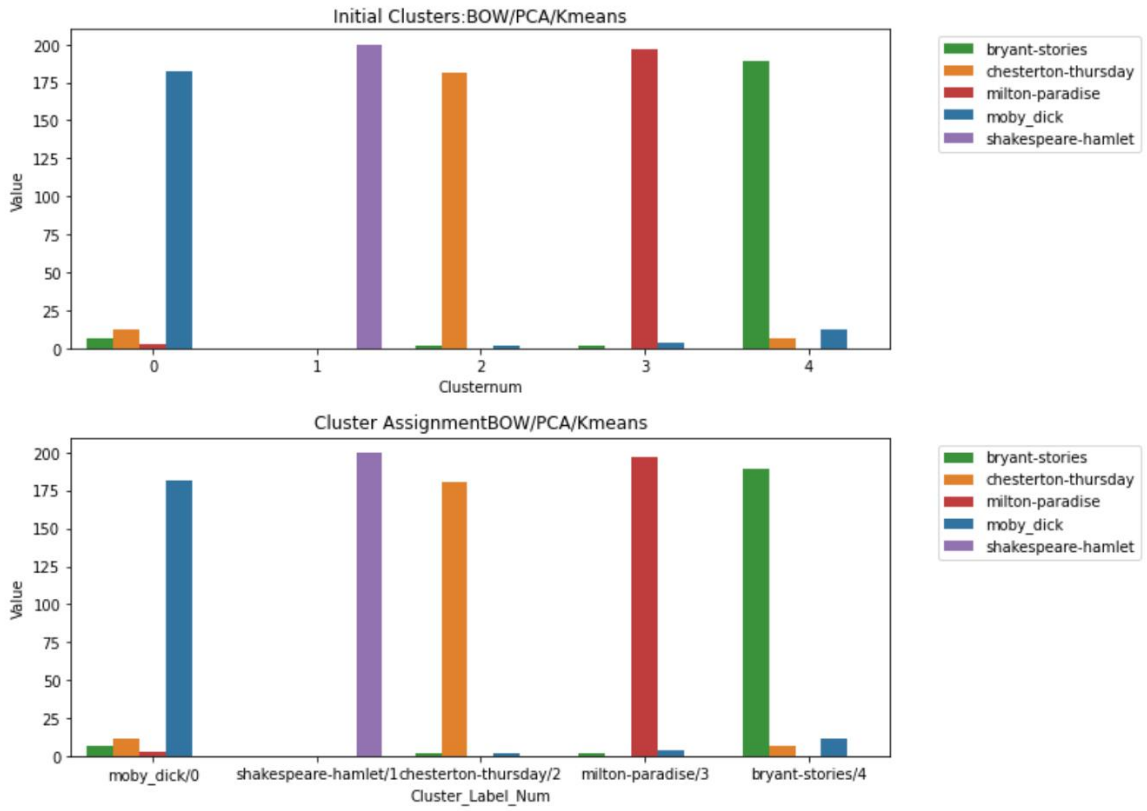

Figure 17. Example of cluster annotation for BOW/PCA-Means. Top figure represents the cluster number and the frequency of the passages from each book. Bottom figure represents the annotated cluster.

Following such figure, because most of the passages from cluster 0 belongs to moby_dick, therefore, cluster 0 is annotated as moby_dick. This process was repeatedly done for all clusters for annotation.

This annotation process was also applied for all algorithms and different feature engineering steps. Using these annotated clusters, we can compare the empirical data with the annotated clusters.

Cohen Kappa Score

Cohen kappa score is a statistical evaluation of the inter-rate reliability for categorical items. In our analysis, we compared all passages' annotated cluster labels against its empirical labels and generate this kappa score for comparison. [2]

PCA visualisations

Because it is not viable to visualize the cluster's space across all 300 features, the visualization process only explored the feature space of the zeroth, first and second components for the clustered results. Because these are the components with the highest variance, they are probably the most discriminatory feature spaces.

K-means

TF-IDF with PCA

The following figure represents different evaluations of the annotated cluster labels against the empirical data labels for TF-IDF/PCA with K-means.

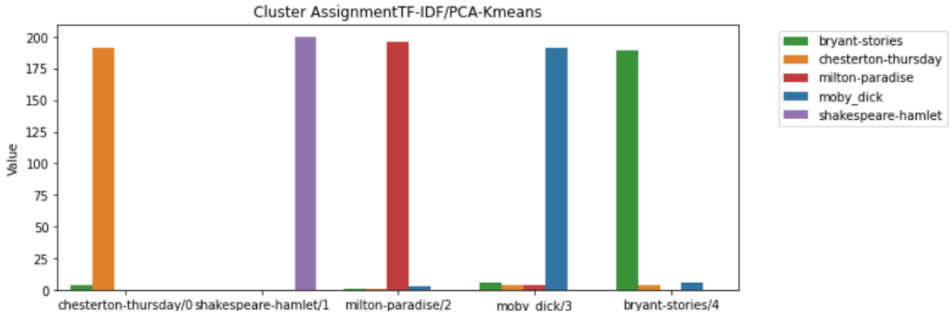

Figure 18. Frequency of passages per book in each annotated cluster from TF-IDF/PCA with K-means.

	bryant-stories	chesterton-thursday	milton-paradise	moby_dick	shakespeare-hamlet	clusterNum	cluster_label
0	4	191	0	0	0	0	chesterton-thursday
1	6	4	4	191	0	3	moby_dick
2	189	4	0	6	0	4	bryant-stories
3	1	1	196	3	0	2	milton-paradise
4	0	0	0	0	200	1	shakespeare-hamlet

Figure 19. Passage frequency of books (empirical label) present in each annotated cluster with TF-IDF PCA with K-means.

From the figures above, each of these clusters are very pure, as most of the data points came from a single book. The purest cluster is the one that is annotated as Chesterton Thursday and Shakespeare, whereas the least pure cluster is the one that was annotated as Moby Dick. The passage frequency table provides insights on these errors, which showed that 6 passages of Bryant stories and 4 passages from Chesterton Thursday that were mislabeled into Moby Dick. Overall, there were 36 total of mislabeled data points and a Cohen kappa score of 0.9587. These are impressive clustering results.

To determine some more insights about these clustering results, here is the visualization of selective components of the feature.

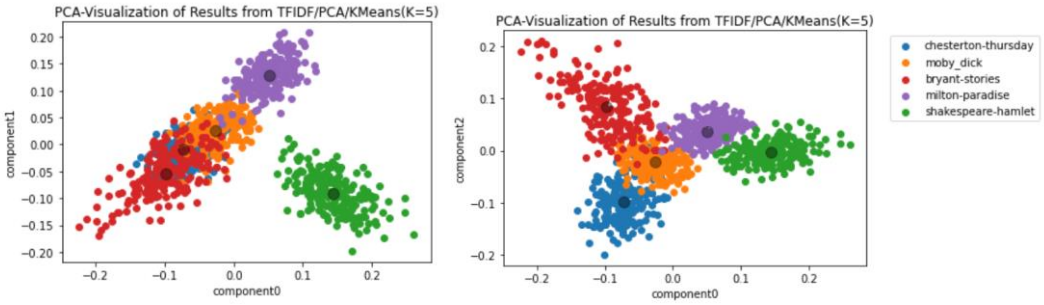

Figure 20. Visualization of the of the 0th component vs 1st component, and 0th vs 2nd component of the clustering results from TFIDF-PCA/K-Means.

These clusters are far apart. The green cluster which represents Shakespeare is especially further apart than the other clusters. The difference in their feature space values reveals why there was very little misclassification of Shakespeare into another cluster while it remained a highly pure cluster itself. Overall, these clusters did not overlay on top of each other, and this is especially obvious in component 0 vs component 2. Therefore, TF-IDF provided a great discriminatory method of vectorization and is what contributed to great cluster purity and high kappa scores.

TF-IDF without PCA

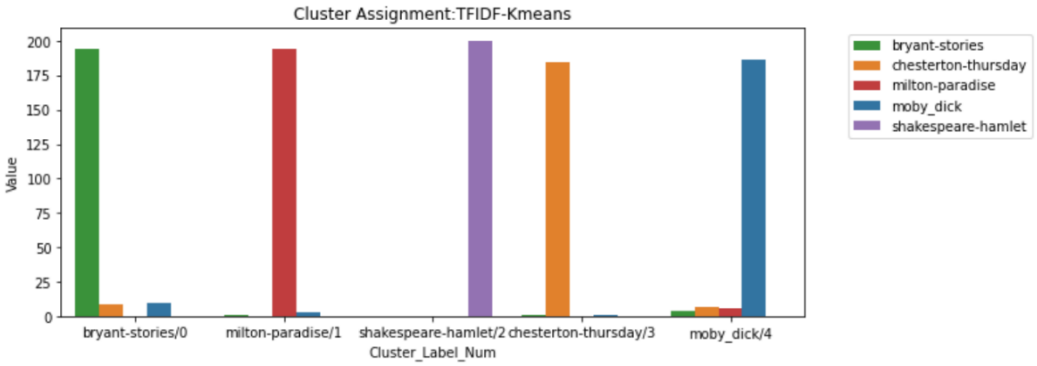

Figure 21. Frequency of passages per book in each annotated cluster from TF-IDF/PCA with K-means.

We repeated the analysis above without PCA to justify the use. The results were similar, however the kappa score decreased to 0.9475.

BOW with PCA

The following figure represents different evaluations of the annotated clusters labels against the empirical data labels for BOW/PCA with K-means.

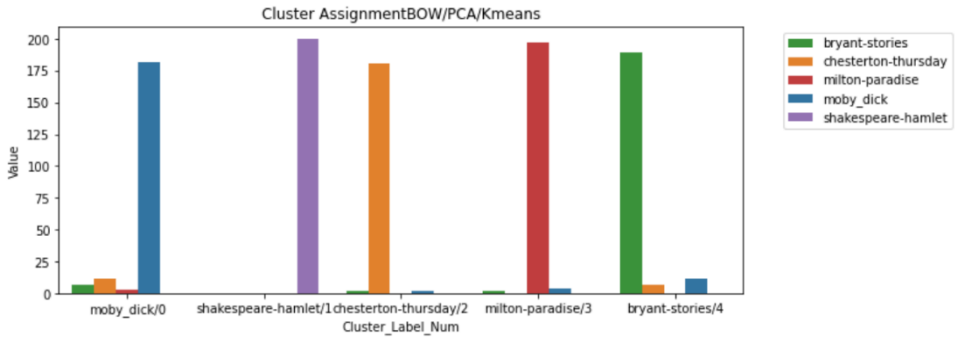

Figure 22. Frequency of passages per book in each annotated cluster from BOW/PCA with K-means.

	bryant-stories	chesterton-thursday	milton-paradise	moby_dick	shakespeare-hamlet	clusterNum	cluster_label
0	7	12	3	182	0	0	moby_dick
1	2	181	0	2	0	2	chesterton-thursday
2	189	7	0	12	0	4	bryant-stories
3	2	0	197	4	0	3	milton-paradise
4	0	0	0	0	200	1	shakespeare-hamlet

Figure 23. Passage frequency of books (empirical label) present in each annotated cluster for BOW/PCA with K-means.

From the results above, the clusters are mostly pure, with more than 90% of the passages in each cluster coming from one book. However, the results decrease slightly compared to when using TF-IDF, with a lot more of the data points that are being misclassified into Bryant stories when using BOW compared to when using TF-IDF. This is reflected in the Cohen kappa score, as a 0.936 value.

To determine some more insights about these clustering results, here is the visualization of selective components of the feature.

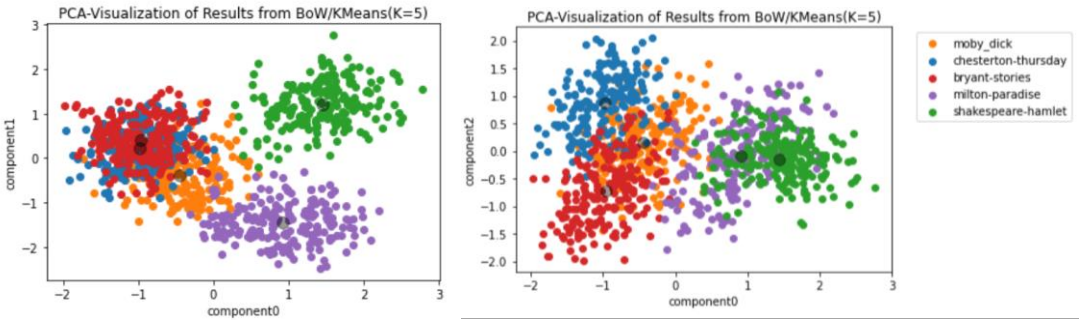

Figure 24. Visualization of the of the 0th component vs 1st component, and 0th vs 2nd component of the clustering results from BOW-PCA/K-means.

The clusters generated from BOW/PCA are much closer and overlayed compared to TF-IDF/PCA. This demonstrates that compared to TF-IDF with PCA, BOW with PCA results in less discriminatory features which are closer together in the feature space, and therefore results in a lower purity of clusters, and more wrongly clustered data points. Similar to TF-IDF with PCA, the green and purple (Shakespeare and Milton) clusters are still more distant compared to the other clusters.

Word 2 Vec

The following figure represents different evaluations of the annotated clusters labels against the empirical data labels for Word2Vec with K-means.

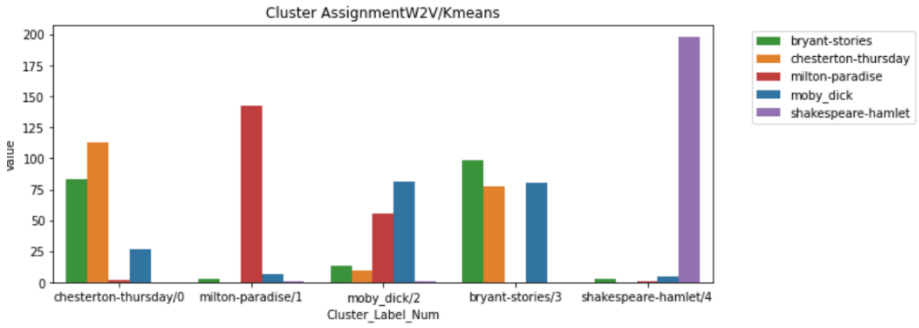

Figure 25. Frequency of passages per book in each annotated cluster from W2V with K-means.

	bryant-stories	chesterton-thursday	milton-paradise	moby_dick	shakespeare-hamlet	clusterNum	cluster_label
0	98	77	0	80	0	3	bryant-stories
1	83	113	2	27	0	0	chesterton-thursday
2	13	10	55	81	1	2	moby_dick
3	3	0	142	7	1	1	milton-paradise
4	3	0	1	5	198	4	shakespeare-hamlet

Figure 26. Passage frequency of books (empirical label) present in each annotated cluster for W2V with K-means.

Figure 25 and Figure 26 demonstrates that the results have drastically decreased when word 2 vec was used as feature engineering step. Cluster 2, which was annotated as the moby dick cluster, has more than half of the passages coming from other books. A similar pattern can be seen with cluster 3 which is Bryant stories. Only cluster 0, annotated as Shakespeare/Hamlet, having more than 90% of its' passages coming from a single author. These poor results are also reflected in the low kappa score of 0.54 and 368 errors in labels.

To determine some more insights about these clustering results, here is the visualization of selective components of the feature.

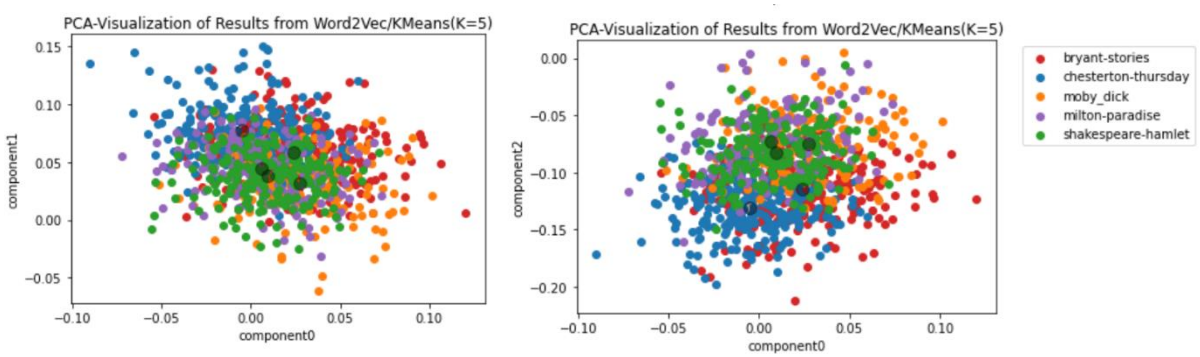

Figure 27. Visualization of the of the 0th component vs 1st component, and 0th vs 2nd component of the clustering results from W2V/K-means.

The cluster analysis reveals that all the five clusters are overlapping on top of each other in these dimensional spaces. This reveal that W2V features have very similar values in the feature space and are not discriminatory. This could explain the poor performance of the clustering results when K-means is used with W2V. Overall, W2V is not recommended as a feature engineering step for this clustering problem.

Hierarchical Clustering

TF-IDF with PCA

The following figure represents different evaluations of the annotated clusters labels against the empirical data labels for TF-IDF/PCA with Hierarchical clustering.

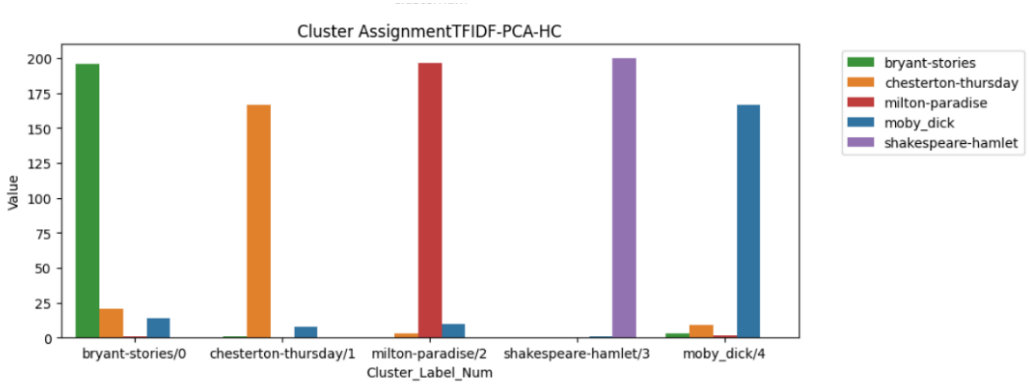

Figure 28. Frequency of passages per book in each annotated cluster from TF-IDF/PCA with Hierarchical Clustering.

	bryant-stories	chesterton-thursday	milton-paradise	moby_dick	shakespeare-hamlet	clusterNum	cluster_label	cluster_label_num
0	196	21	1	14	0	0	bryant-stories	bryant-stories/0
1	1	167	0	8	0	1	chesterton-thursday	chesterton-thursday/1
2	0	3	197	10	0	2	milton-paradise	milton-paradise/2
3	3	9	2	167	0	4	moby_dick	moby_dick/4
4	0	0	0	1	200	3	shakespeare-hamlet	shakespeare-hamlet/3

Figure 29. Passage frequency of books (empirical label) present in each annotated cluster with TF-IDF PCA with Hierarchical Clustering.

From the figures above, each of these clusters are very pure, as most of the data point came from a single book. The purest cluster is the one that is annotated as Shakespeare, whereas the least pure cluster is the one that was annotated as Moby Dick. The passage frequency table provides insights to these errors, which showed that 3 passages of Bryant stories and 9 passages from Chesterton Thursday that was mislabeled into Moby Dick.

Overall, there was 73 total of mislabeled data points with an average Cohen kappa score of 0.9075. To determine some more insights about these clustering results, here is the visualization of selective components of the feature.

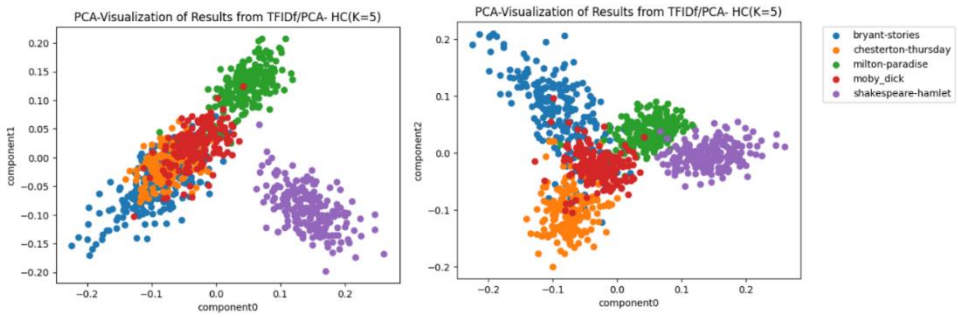

Figure 30. Visualization of the of the 0th component vs 1st component, and 0th vs 2nd component of the clustering results from TFIDF-PCA/Hierarchical Clustering.

These clusters are apart. The purple cluster which represents Shakespeare is especially further apart than the other clusters. The overlap between the blue, orange and red clusters provides us information that these clusters are not very pure. Thus, a decent Cohen kappa score.

Case study of different linkage/distance type on cluster performance

In order to select our optimal parameter, we did a comparative analysis for the different combinations of linkage and distance.

We have following types of linkages - *Ward, average, complete and single.*

There are more but we make use of these four in our study.

And the following types of distances (affinity) - *Euclidean, Manhattan and Cosine.*

We compare the cohen kappa score for all the combinations of the linkage and distance. (ward linkage cannot be combined with Manhattan and cosine, thus no combination for the same in the graph). From the following graph we can infer that there are only two combinations for TF-IDF that has a high cohen kappa score which are - 'Ward- Euclidean' and 'Average-Manhattan' and out of these two 'Ward-Euclidean' has a better cohen kappa score, thus our selection as an optimal parameter for TF-IDF.

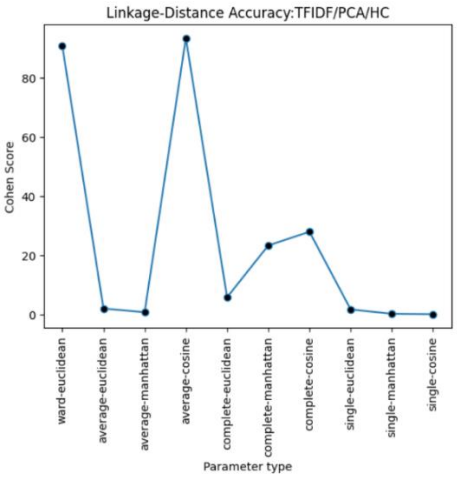

Figure 31. Different combination of linkage/distance pattern for TF-IDF/PCA with HC.

BOW with PCA

The following figure represents different evaluations of the annotated clusters labels against the empirical data labels for BOW/PCA with Hierarchical Clustering.

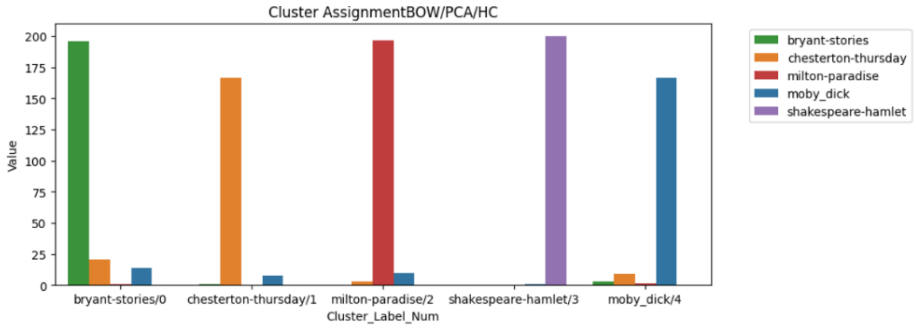

Figure 32. Frequency of passages per book in each annotated cluster from BOW/PCA with Hierarchical Clustering.

	bryant-stories	chesterton-thursday	milton-paradise	moby_dick	shakespeare-hamlet	clusterNum	cluster_label	cluster_label_num
0	196	21	1	14	0	0	bryant-stories	bryant-stories/0
1	1	167	0	8	0	1	chesterton-thursday	chesterton-thursday/1
2	0	3	197	10	0	2	milton-paradise	milton-paradise/2
3	3	9	2	167	0	4	moby_dick	moby_dick/4
4	0	0	0	1	200	3	shakespeare-hamlet	shakespeare-hamlet/3

Figure 33. Passage frequency of books (empirical label) present in each annotated cluster for BOW/PCA with Hierarchical Clustering.

From the cluster results above, the clusters are mostly pure, with more than 90% of the passages in each cluster coming from one book. However, the results decrease slightly compared to when using TF-IDF, with a lot more of the data points that are being misclassified into Bryant stories when using BOW compared to when using TF-IDF. This is reflected in the Cohen kappa score, as a 0.8475 value was calculated given 122 misclassifications.

To determine some more insights about these clustering results, here is the visualization of selective components of the feature.

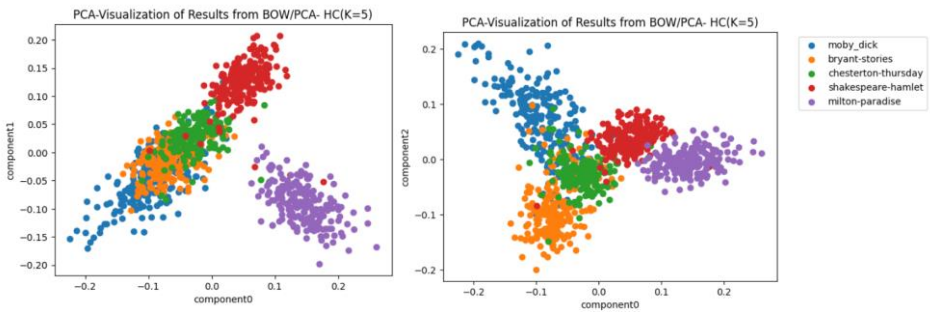

Figure 34. Visualization of the of the 0th component vs 1st component, and 0th vs 2nd component of the clustering results from TFIDF-PCA/Hierarchical Clustering

The clusters generated from BOW/PCA are much closer and overlayed compared to TF-IDF/PCA. This demonstrates that compared to TF-IDF with PCA, BOW with PCA results in less discriminatory features which are closer together in the feature space, and therefore results in a lower purity of clusters, and more wrongly clustered data points.

Case study of different linkage/distance type on cluster performance

From the following figure, we can see that 'Average-Manhattan' which was amongst the best combination for TF-IDF here for BOW shows a very low cohen kappa score.

For BOW we have 'Ward-Euclidean' and 'Average-Cosine' as our best parameters out of which we select 'average-cosine' with a high cohen kappa score.

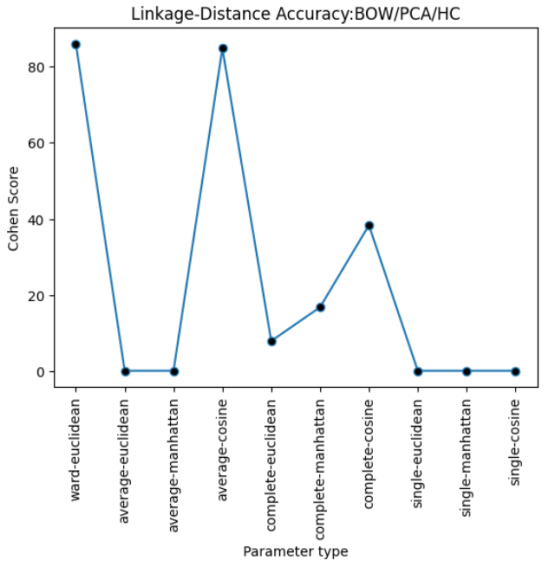

Figure 35. Different combination of linkage/distance pattern for BOW with Hierarchical clustering.

Word 2 Vec

The following figure represents different evaluations of the annotated clusters labels against the empirical data labels for Word2Vec with Hierarchical clustering.

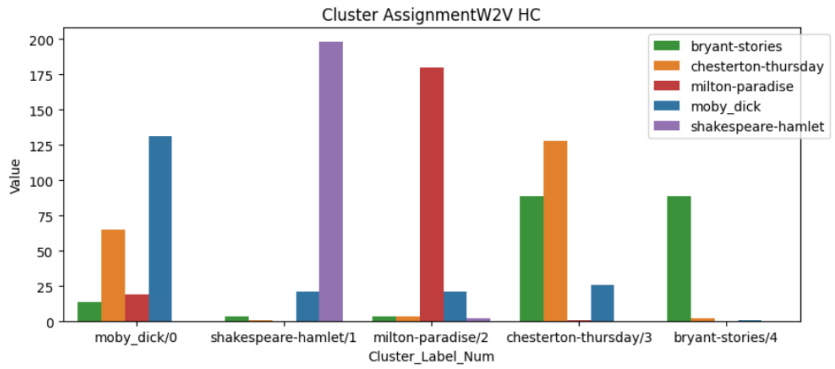

Figure 36. Frequency of passages per book in each annotated cluster from W2V with Hierarchical clustering.

	bryant-stories	chesterton-thursday	milton-paradise	moby_dick	shakespeare-hamlet	clusterNum	cluster_label	cluster_label_num
0	14	65	19	131	0	0	moby_dick	moby_dick/0
1	89	128	1	26	0	3	chesterton-thursday	chesterton-thursday/3
2	4	4	180	21	2	2	milton-paradise	milton-paradise/2
3	89	2	0	1	0	4	bryant-stories	bryant-stories/4
4	4	1	0	21	198	1	shakespeare-hamlet	shakespeare-hamlet/1

Figure 37. Passage frequency of books (empirical label) present in each annotated cluster for W2V with Hierarchical clustering.

Figure 36 and Figure 37 demonstrates that the results has drastically decreased when word 2 vec was used as data engineering step. Cluster 1, which was annotated as the Moby dick cluster, has 42.7% of its' passages coming from other books (14 from Bryant stories, 65 from Chesterton thursday and 19 from Milton paradise). Other clusters' impurity also drastically decreased, with only cluster 0, annotated as Shakespeare/Hamlet, having more than 90% of its' passages coming from a single author. These poor results are also reflected in the low kappa score of 0.675.

To determine some more insights about these clustering results, here is the visualization of selective components of the feature.

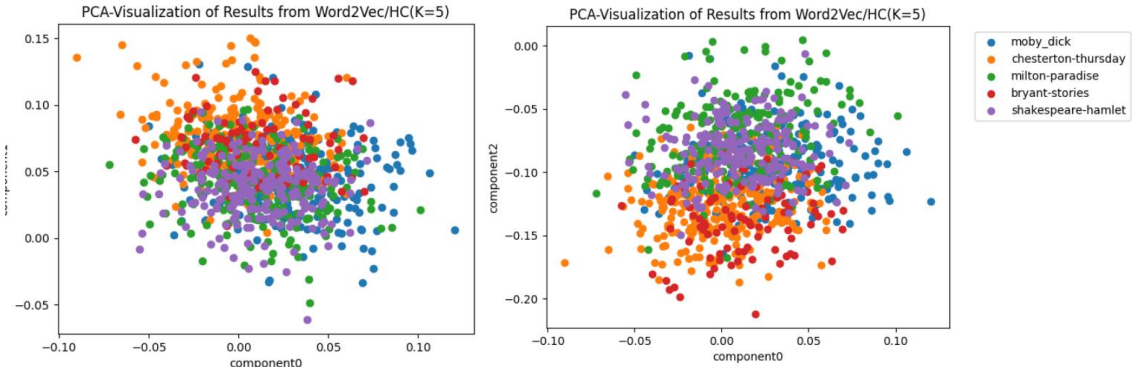

Figure 38. Visualization of the of the 0th component vs 1st component, and 0th vs 2nd component of the clustering results from W2V/Hierarchical Clustering

The cluster analysis reveals that all the five clusters are overlapping on top of each other in these dimensional spaces. This reveal that W2V features have very similar values in the feature space and are not discriminatory. This could explain the poor performance of the clustering results when Hierarchical clustering is used with W2V. Overall, W2V is not recommended as a feature engineering step for this clustering problem.

Case study of different linkage/distance type on cluster performance

For word 2 vec, we have an even more different trajectory for the linkage-distance combinations. However, for this method our best combination is 'Ward-Euclidean'.

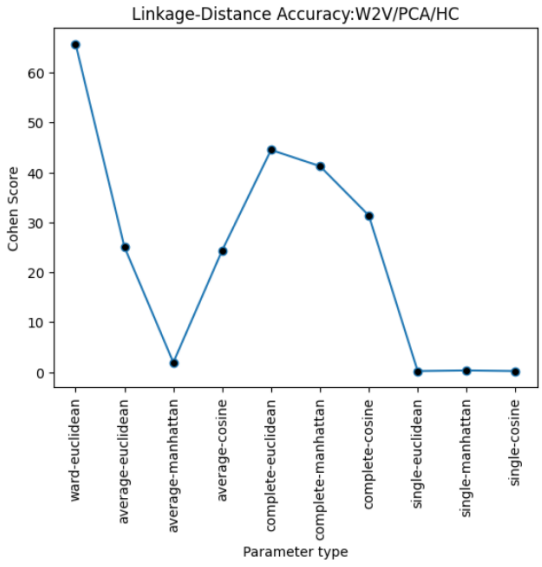

Figure 38. Different combination of linkage/distance pattern for Word 2 Vec.

Gaussian Mixture/ EM

TF-IDF with PCA

The following figure represents different evaluations of the annotated clusters labels against the empirical data labels for TF-IDF/PCA with EM.

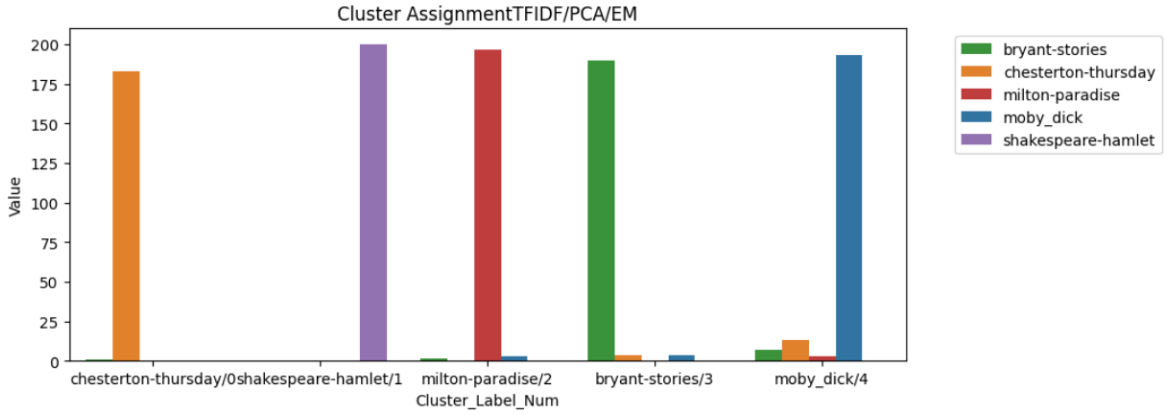

Figure 39. Frequency of passages per book in each annotated cluster from TF-IDF/PCA with EM.

	bryant-stories	chesterton-thursday	milton-paradise	moby_dick	shakespeare-hamlet	clusterNum	cluster_label
0	7	13	3	193	0	4	moby_dick
1	1	183	0	0	0	0	chesterton-thursday
2	190	4	0	4	0	3	bryant-stories
3	2	0	197	3	0	2	milton-paradise
4	0	0	0	0	200	1	shakespeare-hamlet

Figure 40. Passage frequency of books (empirical label) present in each annotated cluster with TF-IDF PCA with EM.

From the figures above, each of these clusters are very pure, as most of the data point came from a single book. The purest cluster is the one that is annotated as Chesterton Thursday and Shakespeare, whereas the least pure cluster is the one that was annotated as Moby Dick. The passage frequency table provides insights to these errors, which showed that 7 passages of bryant stories and 13 passages from Chesterton Thursday that was mislabeled into Moby Dick.

Overall, there was 37 total of mislabeled data points and a Cohen kappa score of 0.9537. These are impressive clustering results.

To determine some more insights about these clustering results, here is the visualization of selective components of the feature.

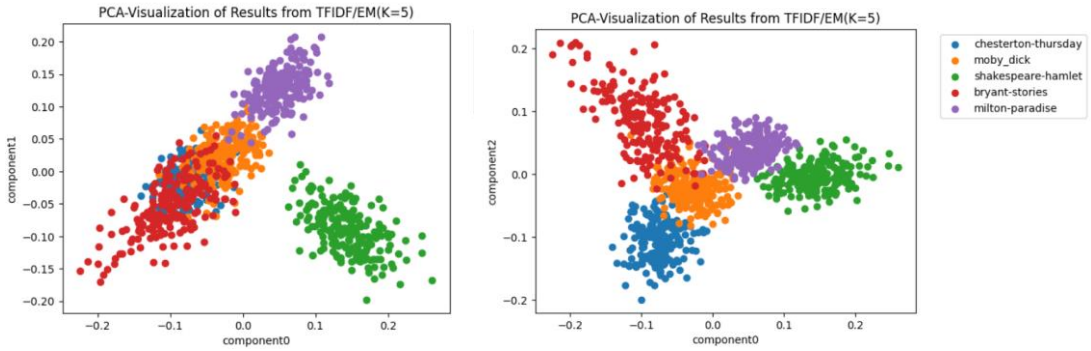

Figure 41. Visualization of the of the 0th component vs 1st component, and 0th vs 2nd component of the clustering results from TFIDF-PCA/EM.

These clusters are far apart. The green and purple cluster which represents Shakespeare and is especially further apart than the other clusters. The difference in their feature space values reveals why there was very little misclassification of Shakespeare into another cluster while it remained a highly pure cluster itself. Overall, these clusters did not overlay on top of each other, and this is especially obvious in component 0 vs component 2. Therefore, TF-IDF provided a great discriminatory method of vectorization and is what contributed to great cluster purity and high kappa scores.

BOW with PCA

The following figure represents different evaluations of the annotated clusters labels against the empirical data labels for BOW/PCA with EM.

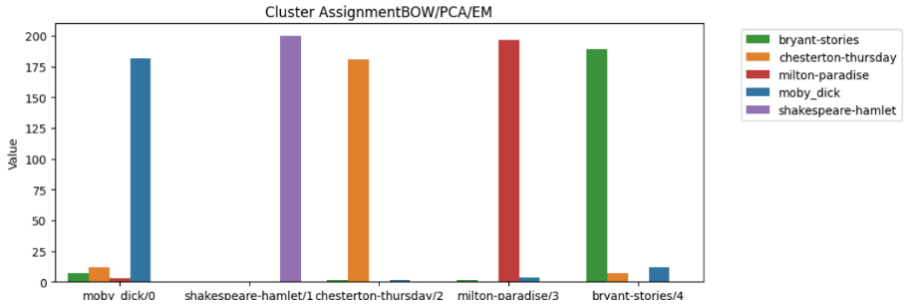

Figure 42. Frequency of passages per book in each annotated cluster from BOW/PCA with EM.

	bryant-stories	chesterton-thursday	milton-paradise	moby_dick	shakespeare-hamlet	clusterNum	cluster_label
0	7	12	3	182	0	0	moby_dick
1	2	181	0	2	0	2	chesterton-thursday
2	189	7	0	12	0	4	bryant-stories
3	2	0	197	4	0	3	milton-paradise
4	0	0	0	0	200	1	shakespeare-hamlet

Figure 40. Passage frequency of books (empirical label) present in each annotated cluster for BOW/PCA with EM.

From the cluster results above, the clusters are mostly pure, with more than 90% of the passages in each cluster coming from one book. However, the results decrease slightly compared to when using TF-IDF, with a lot more of the data points that are being misclassified into Bryant stories when using BOW compared to when using TF-IDF. This is reflected in the Cohen kappa score, as a 0.936 value was calculated given 51 misclassifications.

To determine some more insights about these clustering results, here is the visualization of selective components of the feature.

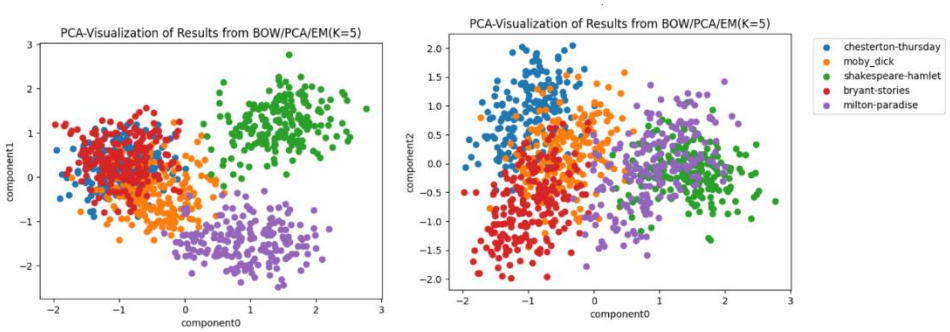

Figure 43. Visualization of the of the 0th component vs 1st component, and 0th vs 2nd component of the clustering results from TFIDF-PCA/EM.

The clusters generated from BOW/PCA are much closer and overlayed compared to TF-IDF/PCA. This demonstrates that compared to TF-IDF with PCA, BOW with PCA results in less discriminatory features which are closer together in the feature space, and therefore results in a lower purity of clusters, and more wrongly clustered data points. Similar to TF-IDF with PCA, the green and purple (shakespear and milton-paradise) clusters are still more distant compared to the other clusters.

Word 2 Vec

The following figure represents different evaluations of the annotated clusters labels against the empirical data labels for Word2Vec with EM.

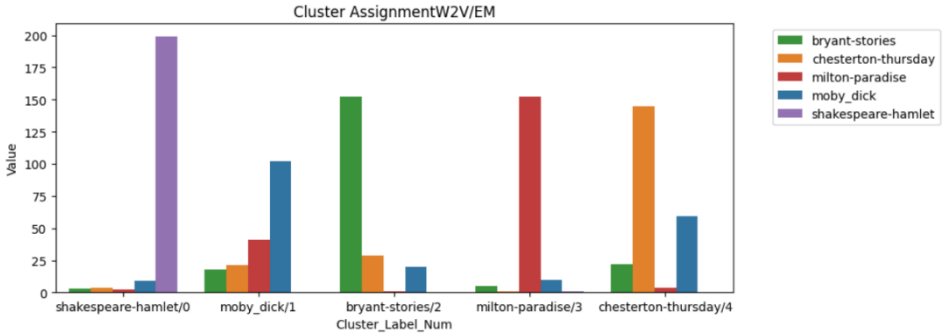

Figure 44. Frequency of passages per book in each annotated cluster from W2V with EM.

	bryant-stories	chesterton-thursday	milton-paradise	moby_dick	shakespeare-hamlet	clusterNum	cluster_label	cluster_label_num
0	22	145	4	59	0	4	chesterton-thursday	chesterton-thursday/4
1	18	21	41	102	0	1	moby_dick	moby_dick/1
2	152	29	1	20	0	2	bryant-stories	bryant-stories/2
3	3	4	2	9	199	0	shakespeare-hamlet	shakespeare-hamlet/0
4	5	1	152	10	1	3	milton-paradise	milton-paradise/3

Figure 45. Passage frequency of books (empirical label) present in each annotated cluster for W2V with EM.

Figure 44 and Figure 45 demonstrates that the results has drastically decreased when word 2 vec was used as data engineering step. Cluster 1, which was annotated as the moby dick cluster, has 43% of its' passages coming from other books (18 from bryant stories, 21 from chesteron thurday and 41 from milton paradise). Other clusters' impurity also drastically decreased, with only cluster 0, annotated as Shakespear/Hamlet, having more than 90% of its' passages coming from a single author. These poor results are also reflected in the low kappa score of 0.6875.

To determine some more insights about these clustering results, here is the visualization of selective components of the feature.

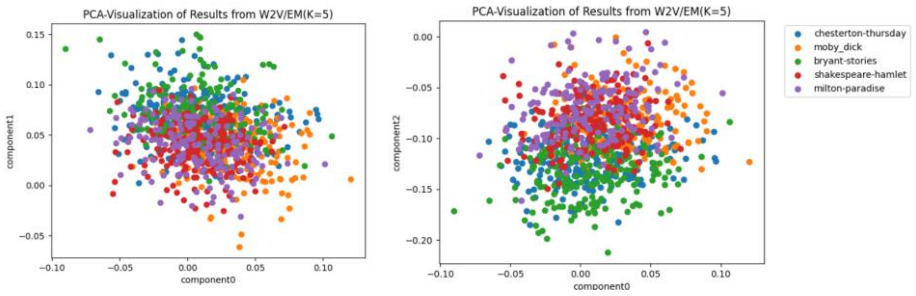

Figure 46. Visualization of the of the 0th component vs 1st component, and 0th vs 2nd component of the clustering results from W2V/EM.

The cluster analysis reveals that all the five clusters are overlapping on top of each other in these dimensional spaces. This reveal that W2V features have very similar values in the feature space and are not discriminatory. This could explain the poor performance of the clustering results when EM is used with W2V. Overall, W2V is not recommended as a feature engineering step for this clustering problem.

Summary of Results

Total Number of Errors

	TF-IDF	TF-IDF + PCA	BOW + PCA	Word 2 Vec
K-means	42	36	51	368
Hierarchical Clustering	N/A	73	122	274
Gaussian Mixtures/ EM	N/A	37	51	250

Kappa Score

No. of errors	TF-IDF	TF-IDF + PCA	BOW + PCA	Word 2 Vec
K-means	0.948	0.959	0.936	0.540
Hierarchical Clustering	N/A	0.909	0.847	0.657
Gaussian Mixtures/ EM	N/A	0.954	0.936	0.687

Overall, the optimum feature selection is TF-IDF with PCA, as it outperforms BOW and W2V with all three different clustering algorithms.

From these kappa score comparisons, it can be concluded that the optimum models are TF-IDF/PCA with K-Means or TF-IDF/PCA with EM. Both algorithms have the top kappa score and the same number of error points based on the annotated labels. Although K-means performed slightly higher than Gaussian Mixture method, this could be due to stochastic differences in the initialization stages. This means that if the same algorithm was to run many times, they are expected to achieve the same average score.

Error Analysis

Error Analysis is used to determine the type of errors made by the model and identify the causes for such a misclassification, to define an approach to fix them in the future.

In this report, the error analysis is performed on the 36 error points of TF-IDF/PCA algorithm with K-Means. These errors are shown by the figure below.

		passage
empirical cluster	cluster	
chesterton-thursday	moby_dick	13
bryant-stories	moby_dick	7
chesterton-thursday	bryant-stories	4
moby_dick	bryant-stories	4
milton-paradise	moby_dick	3
moby_dick	milton-paradise	3
bryant-stories	milton-paradise	2
	chesterton-thursday	1

Figure 47. Total number of mislabeled data points from each book. Clusters generated by TF-IDF/PCA K-Means.

Majority of these errors are passages taken from Chesterton Thursday and Bryant stories, and misclassified into Moby-dick. To determine this error analysis, the word visualization graphs for these passages was plotted.

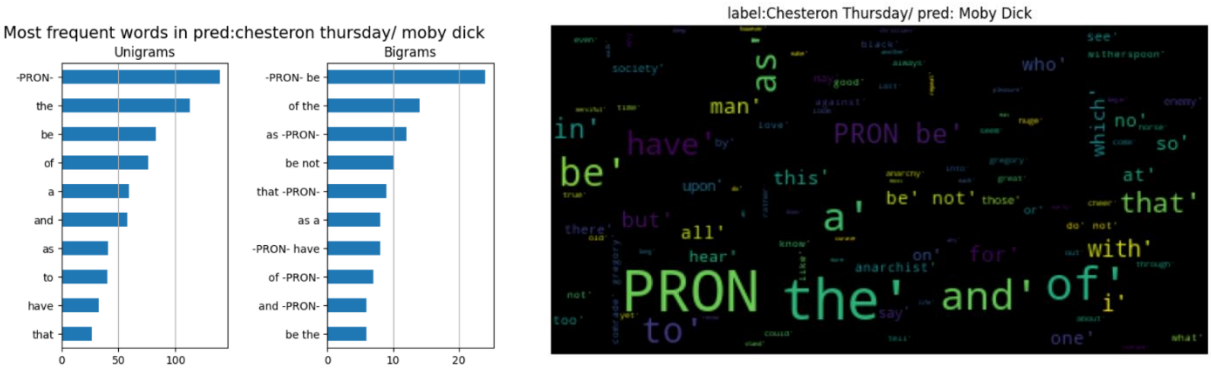

Figure 48. Visualization of Chesterton Thursday books mislabeled as moby dick. Left: Bigrams. Right: word cloud.

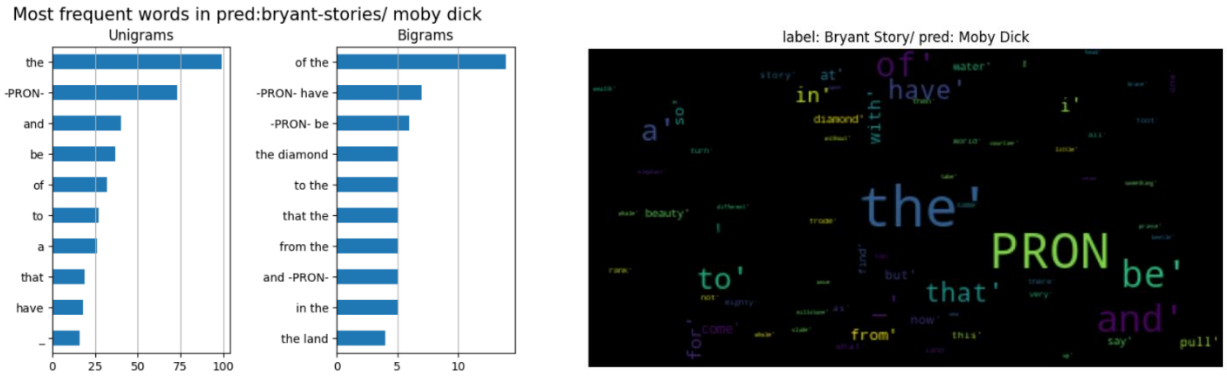

Figure 49. Visualization of bryant story books mislabeled as moby dick. Left: Bigrams. Right: word cloud.

From these figures, it can be determined that these passages contain a heavy amount of “pron” token. This token is supposed to be the lemmatized word for many pronouns such as “he”, “she”, “we”, etc. These words may have occurred so often that they do not include any discriminatory features. Other words that do not add any importance but are frequently occurring are common words such as “to”, “have”, “the”, etc.

For further analysis, the impact of the stop words was taken out as a case study shown below. The purpose of this is to investigate whether the error rate improved upon taking out these stop words.

Case study with removing frequent and rare words.

From the previous error analysis, it was determined that most of the common words in the misclassified data points were stop words. To determine the impact of these stop words, we took out these stop words re-run the algorithm with TF-IDF-PCA with K-means. This resulted in a kappa score of 0.936, which is worse compared to the algorithm without the stop word preprocessing.

The errors are shown as below.

		passage
empirical cluster	cluster	
moby_dick	bryant-stories	13
chesterton-thursday	bryant-stories	8
bryant-stories	moby_dick	6
	milton-paradise	5
chesterton-thursday	milton-paradise	5
moby_dick	milton-paradise	5
chesterton-thursday	moby_dick	4
milton-paradise	moby_dick	3
bryant-stories	chesterton-thursday	1
moby_dick	chesterton-thursday	1

Figure 50. Total number of mislabeled data points from each book. Clusters generated by TF-IDF/PCA K-Means.

It's important to specify that while this algorithm reduced the previous error of Chesterton Thursday and Bryant stories getting mis-classified into Moby dick, taking out the stop words did seem to introduce new sets of error.

To compare the type of error that is made in this algorithm, the word visualization graphs for the error of Bryant stories misclassified into Moby dick is plotted.

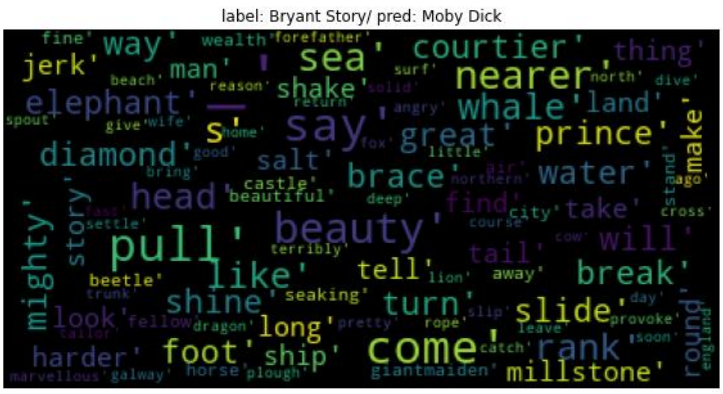

Figure 51. Visualization of Bryant story books mislabeled as moby dick. Left: Bigrams. Right: word cloud.

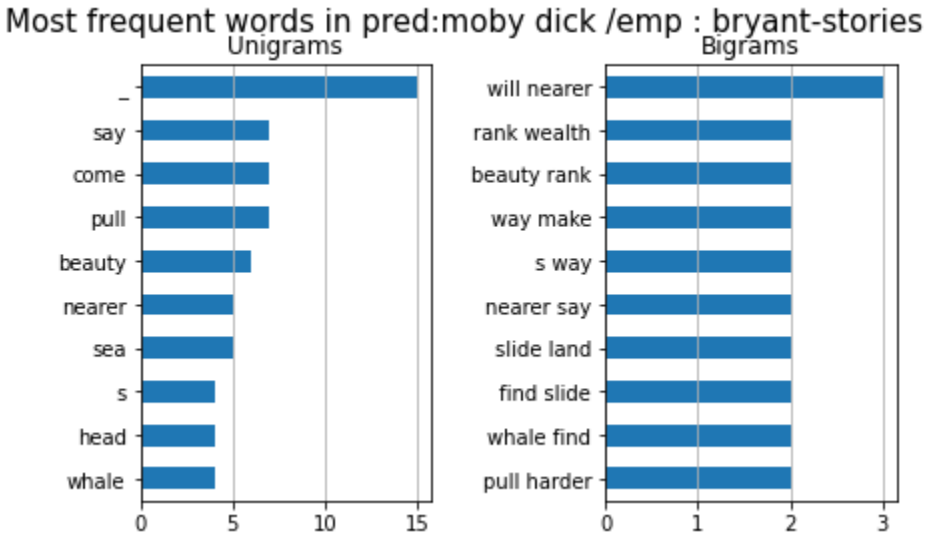

Figure 52. Visualization of Bryant story books mislabeled as moby dick. Left: Bigrams. Right: word cloud.

These figures demonstrates that the common stop words have been taken out, and a lot mor variety of different words have appeared. It is possible that these words such as “come” or “sea” or “foot” have appeared in both passages. In such cases, taking out the stop words may reduce the context of these words (e.g he comes becomes comes), and therefore reduces the clustering performance.

Conclusions

To conclude, we tuned and compared 4 different algorithms with 3 feature selections to overcome the challenge of clustering passages from five different books. The optimal combination after our analysis was K-Means with TF-IDF/PCA with a kappa score of 0.9587. The optimal feature engineering method is TF-IDF with PCA, followed by BOW-PCA, then W2V.

We saw that EM and K-Means resulted in similar kappa scores, whereas hierarchical clustering resulted in slightly lower kappa scores.

Although not removing stop words can include too many non-discriminatory features making it difficult to cluster certain passage, taking out stop words also reduces some important discriminatory features.

References

1. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.
2. Géron, A. (2020). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. Beijing: O'Reilly.
3. D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. The Journal of Machine Learning Research, 3:993–1022, 2003.

ReadMe

Problem Statement

Make use of clustering algorithms to take five different samples of Gutenberg digital books, which are of five different genres and authors, that are semantically different. Is it possible to predict for the authorship of a book given a passage?

Goal

The goal is to determine the optimal algorithm (with the optimal parameters) that would be able to correctly classify the author of a given passage.

Hypothesis

We hypothesize that using proper feature engineering, appropriate unsupervised machine learning methods, we could accurately cluster the authorship of the passage. These include four different algorithms namely, K-Means, Hierarchical Clustering, Gaussian Mixtures and LDA with, different pre-processing techniques like - TF-IDF, Bag of Words and Word 2 Vec.

We hypothesize that

Project Setup

1. Load Google colab - https://colab.research.google.com/
2. Create a new notebook.
3. Upload the book.xlsx file in the file section.
4. Follow this tutorial of Load-spacy_model_en_core_web_lg-on-google-collab to download the required dictionary for Word2Vec.

Note

The xlsx file that you upload will vanish if the session expires.

If uploading the file from local, remember to have the file in the folder where the .py file is located.

Exact path should be mentioned along with the file name in order to upload the file correctly.

Order of the cells should be followed during execution for optimized output and avoiding the reruns.

▾ Please specify the path below

```
[ ]  df1=pd.read_excel('books.xlsx')
```

There will be slight changes in the performance every time the notebook is run.

Process

Import all the required libraries.

Data

We need to download spacy large dictionary to benefit from the large vocabulary and to use Word2Vec. Run the below code in Jupyter notebook.

```
[ ]  !python -m spacy download en_core_web_lg
```

To load the following package for lemmatization on google colab, please follow this tutorial -
Load-spacy_model_en_core_web_lg-on-google-collab

```
!python -m spacy download en_core_web_lg
```

Then, **restart the colab runtime**!
(to do this in the colab menu, go for Runtime > Restart runtime…).
After that, executing

```
import spacy
nlp = spacy.load('en_core_web_lg')
```

We use PCA for dimensionality reduction to first reduce the features down to 300 and then feed the tfidf-pca and tfidf-bag of words into our models.

▾ Dimentionality reduction using PCA

```
[ ] from sklearn.decomposition import PCA
```

```
[ ] pca = PCA(n_components=300)
    X_train_tfidf_pca = pca.fit_transform(tfidf_train_data.toarray())
    X_train_bow_pca = pca.fit_transform(Bow_train_data.toarray())
```

```
[ ]
```

```
[ ]
    model = LdaModel(X_train_tfidf_pca, 5, common_dictionary)

    cm = CoherenceModel(model=model, corpus=common_corpus, coherence='u_mass')
    coherence = cm.get_coherence()  # get coherence value
```

Data Processing and Cleansing

Getting rid of the punctuations and Spacy Lemmatization

- Our choice of Spacy library was motivated by the fact that it can perform tokenization in 0.2 milliseconds compared to nltk's 4 milliseconds.
- We are performing a lemmatization in the below code. It is a process of converting many different forms to its root word. For example, words such as building, built, build etc will be converted to its base form or lemma 'build', based on the context of the words surrounding it.
- The nlp pipeline created using Spacy will automatically perform the tokenization, parsing and tagging processes for us.
- We have rerun this code without these preprocessing steps which lead to lower accuracy.

▾ Getting rid of punctuations

```
[ ] def return_split(passage):
      return (re.sub(r'[^\w\s]', '', str(passage).lower().strip()))
```

```
[ ] df1['passage']=df1['passage'].apply(return_split)
```

```
[ ] nlp=spacy.load('en_core_web_sm')
```

▾ Lemmatization

```
[ ]  def lemmatization(x):
       X_list=[]
       for token in nlp(x):
         lemma=token.lemma_
         X_list.append(lemma)
       return " ".join(X_list)
```

```
[ ]  df1['passage']=df1['passage'].apply(lemmatization)
```

We use clustering algorithms on the x data to generate labels.
We hold out the y to compare with the clustered labels for evaluation.

Feature Engineering

TF-IDF

As a first choice, we are using TF-IDF to transform our X (Passages) into vectors

- TF-IDF is a vectorization algorithm which is used to represent textual data in numerical vectors . It gives weights to the words depending on their frequency.
- Compared to bag of words, TF-IDF also calculates the inverse document frequency, which will factor in the frequency of the word to occur in all documents. This will take out very commonly used words.
- The 'Fit' is used to identify the vocabulary and frequency whereas the 'Transform' is used for conversion to a vector.

Bag of Words (BOW)

- Bag of words is a form of vectorization process for extracting features from the textual data. Like Tf-idf, this is another step of feature processing, but it does not take into account how often the words will appear across different documents.
- The term 'bag' implies that the order of the words appearing in the document is ignored. The model is only concerned with whether the word is present in a document, not with the location of the word in the document.

Word 2 Vec

- Word 2 vec trains a neural network to process text into a vector output
- This can be achieved using either continous bag of word or skipgram

Models

We make use of four models namely – K-Means, Hierarchical Clustering, Gaussian Mixture and LDA and try different combinations of feature engineering techniques – TF-IDF, BOW and Word 2 Vec to choose our champion model, visualize the results and do the comparative analysis.

K-Means

We set the init parameter to k-means++ to ensure that the initialized centroids are as distant from each other as possible. This will reduce the number of iterations of the model before it reaches the correct centroid values.

```
[ ]  k = 5
     km = KMeans(n_clusters=k, init='k-means++',max_iter=200, n_init=10,random_state=42)
     km.fit_predict(X_train_tfidf_pca)
```

Hierarchical Clustering

Building a model after selecting the optimal parameters

```
[ ]  k=5

     Hclustering = AgglomerativeClustering(n_clusters=k, affinity='euclidean',linkage='ward')
     Hclustering.fit(X_train_tfidf_pca)
     Predicted_cluster = Hclustering.labels_
     TFIDF_HC_cluster=returnBookToCluster(df1,Predicted_cluster)
     y_correspond_label=return_cluster_y_emp(TFIDF_HC_cluster,Predicted_cluster,df1)
     cohen=cohen_kappa_score(y_correspond_label['empirical cluster'],y_correspond_label['cluster'])
     cohen

     0.90875
```

Gaussian Mixture

```
[ ]
     gm=GaussianMixture(n_components=5,random_state=42,n_init=10)
     gm.fit(X_train_tfidf_pca)
     y_cluster_tfidf_pcagm=gm.predict(X_train_tfidf_pca)
```

Conclusion

To conclude, we tuned and compared 4 different algorithms with 3 feature selections to overcome the challenge of clustering passages from five different books. The optimal combination after our analysis was K-Means with TF-IDF/PCA with a staggering kappa score of 0.9587. The optimal feature engineering method is TF-IDF with PCA, followed by BOW-PCA, then W2V.