Team 6

Testing different feature transformation and text clustering techniques and evaluating their results

text clustering

**Group members:**

* Esraa Badawi
* Esraa El-Kot
* Salma Sultan
* Sondos Ali

Table of Contents

[Books Selection 3](#_Toc106555329)

[Visualizing the books 3](#_Toc106555330)

[Preprocessing and Data Cleansing 4](#_Toc106555331)

[Data exploration and visualization 5](#_Toc106555332)

[Most Frequent N-Grams 5](#_Toc106555333)

[Book 1 partitions 5](#_Toc106555334)

[Book 2 partitions 5](#_Toc106555335)

[Book 3 partitions 6](#_Toc106555336)

[Book 4 partitions 6](#_Toc106555337)

[Book 5 partitions 6](#_Toc106555338)

[Using LDA to view topics 7](#_Toc106555339)

[Feature Engineering 8](#_Toc106555340)

[Feature Extraction and Transformation 8](#_Toc106555341)

[BOW 8](#_Toc106555342)

[TF/IDF 8](#_Toc106555343)

[LDA 9](#_Toc106555344)

[Word embedding 9](#_Toc106555345)

[Feature selection 9](#_Toc106555346)

[Selecting top features from BOW 9](#_Toc106555347)

[Using N-Grams 9](#_Toc106555348)

[Selecting top features from Tf-IDF 9](#_Toc106555349)

[LDA 10](#_Toc106555350)

[Word Embedding 10](#_Toc106555351)

[Text Clustering Models 11](#_Toc106555352)

[Selecting best number of clusters 11](#_Toc106555353)

[WCSS Score and Elbow method 11](#_Toc106555354)

[Silhouette Score 11](#_Toc106555355)

[Models 12](#_Toc106555356)

[Evaluation Metrics 12](#_Toc106555357)

[K-Means Model 12](#_Toc106555358)

[Gaussian Mixture Model 16](#_Toc106555359)

[Hierarchical Agglomerative Model 22](#_Toc106555360)

[Models’ comparison 26](#_Toc106555361)

[Champion Model Analysis 27](#_Toc106555362)

[Visualizing incorrect predictions 27](#_Toc106555363)

[Analyzing Incorrect labeled partitions 28](#_Toc106555364)

[Conclusion 31](#_Toc106555365)

# Books Selection

A group of five books were selected from Gutenberg library, each book had a different author but all of them fall under detective and mystery stories category, the labels are used to identify the books partitions throughout the analysis and modeling.

A picture containing text, sign, blackboard, blue

Description automatically generatedA close-up of a document

Description automatically generated with medium confidenceA picture containing text, mollusk, invertebrate

Description automatically generatedA picture containing diagram

Description automatically generatedDiagram

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Book name | Author | Book Label | Book category |
| The Adventures of Sherlock Holmes | Arthur Conan Doyle | a | Fiction |
| On the Origin of Species By Means of Natural Selection | Charles Darwin | b | Biology |
| Astronomy for Amateurs | Camille Flammarion | c | Astronomy |
| Memoirs of Extraordinary Popular Delusions and the Madness of Crowds | Mackay | d | Crowd psychology |
| History Of Egypt, Chaldæa, Syria, Babylonia, And Assyria In The Light Of Recent Discovery | Harry Reginald, Leonard William | e | History |

# Visualizing the books

The word cloud highlights the most frequent words as a significant text. So, to get a general idea about the contents of each book the following word clouds were generated.

|  |  |
| --- | --- |
| Figure 1-wordcloud for book "a" | Figure 2-wordcloud for book "b" |
| Figure 3-wordcloud for book "c" | Figure 4-wordcloud for book "d" |
| Figure 5-wordcloud for book "e" |  |

# Preprocessing and Data Cleansing

The following steps were followed to transform the raw data into useful and efficient format.

1. The books were downloaded from Gutenberg online library, the extra padding added by Gutenburg library which included copyrights information was removed and only the book text was retrieved.
2. The book’s sentences were tokenized using nltk word tokenizer.
3. Stop words and punctuation marks were removed, so that models can focus on unique information that can be used for classification.
4. The first 100 words were skipped to avoid getting cover page, table of contents, and introduction chapter text in the selected books partitions which would not have helped in classifying the book. It is an important step at data cleaning because they are unnecessary or useful input for the model and can affect the performance.
5. 200 book partitions of 150 words each were acquired from all the books, each books partitions had a unique label and all of the partitions were added to a dataframe to facilitate further processing, the books partitions were shuffled.

Text

Description automatically generated with medium confidence

# Data exploration and visualization

## Most Frequent N-Grams

N-grams plots the most frequent words or word combinations in each book. To get an overall understanding of the partitions of each book the n-grams of each book partitions were plotted.

### Book 1 partitions

Chart, bar chart

Description automatically generatedMost Frequent Unigrams and Bigrams for the 1st book partitions

### Book 2 partitions

Chart, bar chart

Description automatically generatedMost Frequent Unigrams and Bigrams for the 2nd book partitions

### Book 3 partitions

Chart, bar chart

Description automatically generatedMost Frequent Unigrams and Bigrams for the 3rd book partitions

### Book 4 partitions

Most Frequent Unigrams and Bigrams for the 4th book partitions

Graphical user interface

Description automatically generated

### Book 5 partitions

Graphical user interface, application

Description automatically generated with medium confidenceMost Frequent Unigrams and Bigrams for the 5th book partitions

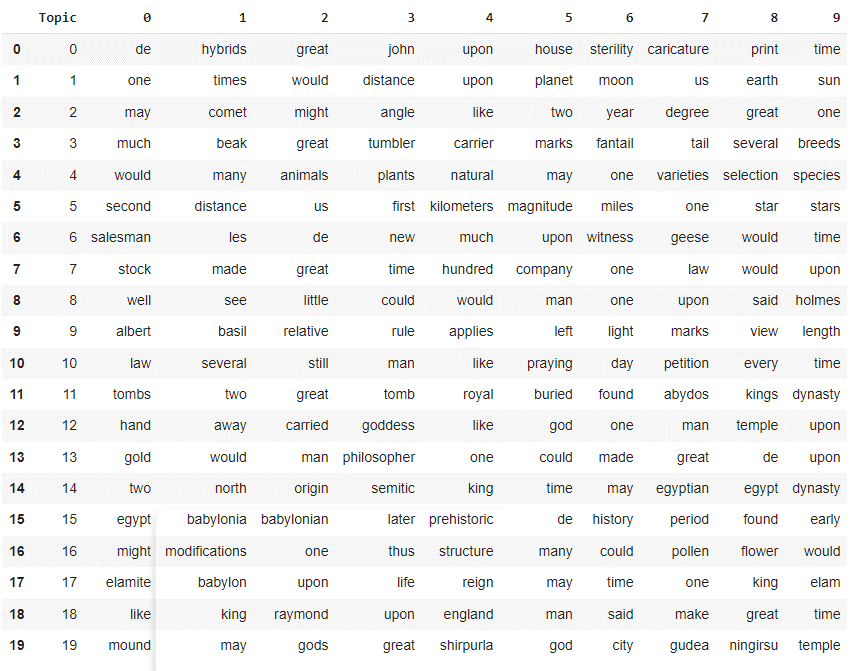
# Using LDA to view topics

Latent Dirichlet Allocation (LDA) is a generative statistical model that uses unseen groups to describe a set of observations, with each group explaining why some sections of the data are similar. Each document is formed by a statistical generative process, each document is a mixture of subjects, and each topic is a mixture of words, according to LDA's primary assumption. The weight of linkages between documents and subjects, as well as between topics and words, is determined by this method.

A computer screen capture

Description automatically generated with medium confidenceThe 1000 books partitions were divided to 20 topics. To get the dominant topic in for each text partition, the highest probability from each topic was returned by argmax function, then added to the partition row in the column named “Topic”. Also, the most frequent word from each topic is shown

The topmost frequent words in every topic were printed to get a sense of the ideas that what each topic represented.



# Feature Engineering

## Feature Extraction and Transformation

### BOW

Bag of words is used to transform raw text partitions to a numerical format using words occurrences counts. CountVectorizer was used to model bag of words for books text partitions and transform partitions into numerical vectors.

A sheet of music

Description automatically generated with low confidence

The total number of unique words found in all text partitions is 16876

### TF/IDF

TF/IDF gives higher score for more important words and lower score for less important words. The TF-IDF score was calculated for each word in the text using TfidfTransformer. When using this transformation method with different models, the TfidfVectorizer was used instead.

Table

Description automatically generated with medium confidence

### LDA

LDA is a topic modeling technique that can be used to get topics from corpus, it divides the passed corpus into a specified number of topics then for each passed document it creates a vector that contains the similarities percentages of the passed document and the generated topics. This vector can be used as a numeric representation of the documents.

### Word embedding

Word embedding is used to represent the words of text in a real-valued vector that encodes the meaning of the word, the words that have approximate vectors are expected to be similar in meaning.

Doc2vec is an algorithm of word embedding which is used to convert every partition of books into a vector to get similarity between partitions while using the clustering models.

## Feature selection

### Selecting top features from BOW

To reduce the number of features, top features were selected from BOW by specifying that min\_df equals 100, meaning that we ignore terms that appear in lessthan 100 of the partitions.

### Using N-Grams

After printing the bigrams of each book partitions, it was noticed that in every partition contained at most two bi grams that were reasonable. So, the experiments were conducted using unigrams only.

### Selecting top features from Tf-IDF

To reduce the number of features, top features were selected from TF-IDF by specifying that min\_df equals 50, meaning that we ignore terms that appear in lessthan 50 of the partitions.

### LDA

Using an LDA model to do feature transformation with number of topics was set to 20, a new set of features were acquired and later used with each clustering model. The output of the LDA was a vector for every input text partition that contained the percentage of how much the text partition was similar to LDA generated topics.

### Word Embedding

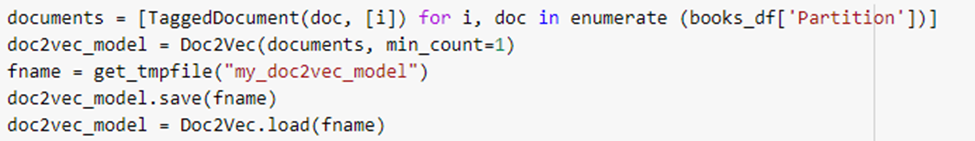
There were two options to use dec2vec algorithm, first, to use a pretrained model on Wikipedia data, but using this pretrained model led to problems because the model couldn’t recognize some words in each partition, some of these words were characters of peoples’ names. The number of partitions that contained unknown words were 600. Dropping the unknown words from the partitions was considered but it may lead to losing significant tokens that may help other clustering algorithms in getting accurate results, so this solution was discarded.

The second option -which was selected for this analysis- was to train a word embedding model and use it to generate features from text partitions.

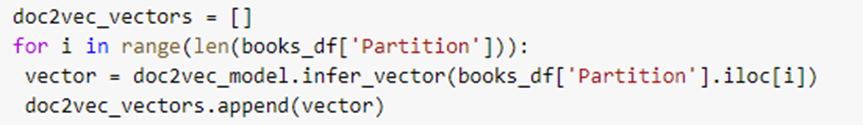
The code for creating and training a custom doc2vec model to be used in converting partitions to vector is as follows:

**The main steps of building model:**

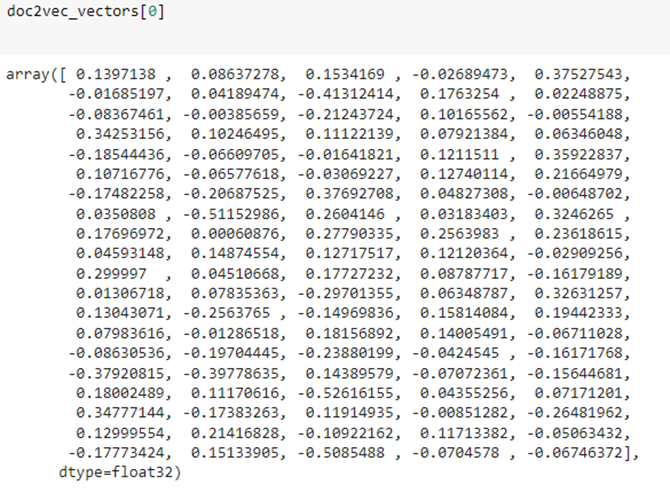
1. Build the model using all books partitions as training data.
2. Save the model to file and then load the model.



1. Use the model to generate vectors.



The output is 1000 vectors, each with 100 numbers (1000,100), below is a sample of the output that represents the first book partition in the dataset.



# Text Clustering Models

## Selecting best number of clusters

Since every method of feature transformation yields a different set of features, that meant that the best number of clusters varied from one method to another. So, the best number of clusters was defined for each feature transformation method and every model.

### WCSS Score and Elbow method

The elbow technique plots the cost function value produced by various k values. When k is increased, the average distortion decreases, each cluster has fewer constituent examples, and the instances are closer to their respective centroids. As k grows larger, however, the average distortion improves less. The elbow is the value of k at which the improvement in distortion diminishes the most, and at which we should cease dividing the data into more clusters.

### Silhouette Score

A point's silhouette score indicates how close it is to its closest neighbor points across all clusters. It gives information regarding clustering quality, which can be used to assess whether more clustering refinement on the existing clustering is necessary.

**Note:** For the K-means algorithm both **WCSS** and **Silhouette scores were used to get the best number of clusters, while for the remaining algorithms only the Silhouette score was used.**

## Models

In the following section we describe every model, the different feature transformation methods used, the best number of clusters for each feature transformation method, the resulting clusters predicted, and the metrics used to evaluate the clusters quality.

### Evaluation Metrics

#### Kappa score

Kappa= (observed agreement−expected agreement) / (1−expected agreement).

When two measurements agree only at the chance level, the value of kappa is zero. When the two measurements agree perfectly, the value of kappa is positive.

To calculate the Kappa score for very cluster, a label was selected from the original five labels values to replace each generated cluster. This happened as follows:

1. For every cluster the number of original partitions label were counted.
2. The label with the maximum number of occurrences in the cluster was selected to replace the cluster name generated by the clustering algorithm.
3. The kappa score was calculated based on the original partitions labels and the labels acquired from previous steps.

#### **Coherence score**

In statistics, coherence is a measure of the information's quality, either inside a single data collection or between data sets that are comparable but not identical. Fully coherent data is logically consistent and can be combined for analysis with confidence.

#### **Silhouette score**

This score gives information regarding clustering quality, it was used to determine the best number of clusters as well.

#### Cluster Scatter plot

T\_SNE method was used to reduce the number of features to 2 to draw the Scatter between clusters acquired from each clustering model.

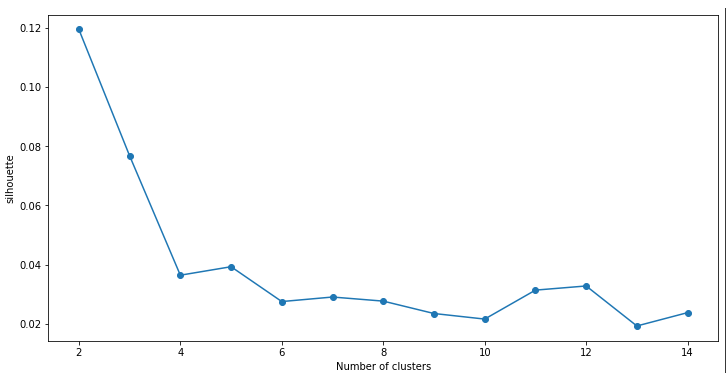
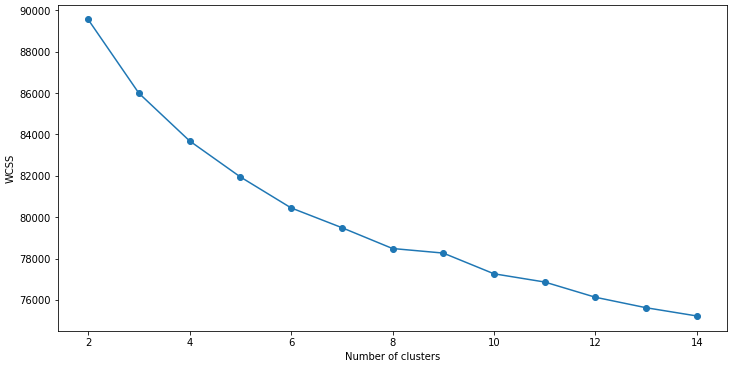
### K-Means Model

K-Means Clustering is an unsupervised learning technique that divides an unlabeled dataset into clusters.

#### BOW

##### Selecting the best number of clusters

**The following graphs show the WCSS and silhouette scores for different number of clusters using features acquired from BOW.**



From the figures shown, the best number of clusters is 5.

##### Model

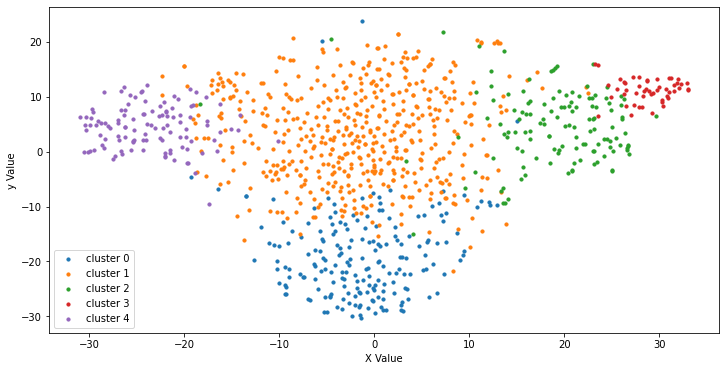
A pipeline is created. First, apply BOW transformer, then add the result to our estimator which is k means. After that, we fit the pipeline, apply evaluation by calculating the kappa, coherence, and silhouette.

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| Kappa | 0.5825 |
| coherence | 44080.846935 |
| silhouette | 0.0717 |

**Cluster Quality**

**Clusters scatter plot using T-SNE.**



#### Tf-IDF

##### Selecting the best number of clusters

**The following graphs show the WCSS and silhouette scores for different number of clusters using features acquired from BOW.**

Chart, line chart

Description automatically generated

From both figures, the best number of clusters is 5.

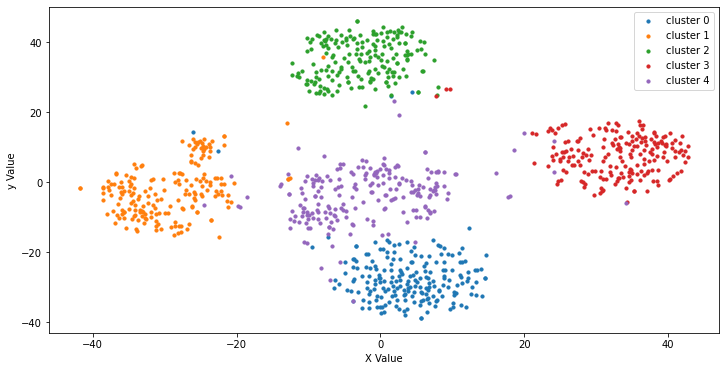
##### Model

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| Kappa | 0.96125 |
| Coherence | 881.95255 |
| silhouette | 0.05923 |

**Cluster Quality**

**Clusters scatter plot using T-SNE.**



#### LDA

##### Selecting the best number of clusters

Elbow, Silhouette methods for LDA

From the 2 figures the best number of the clusters is 5

Chart, line chart

Description automatically generated

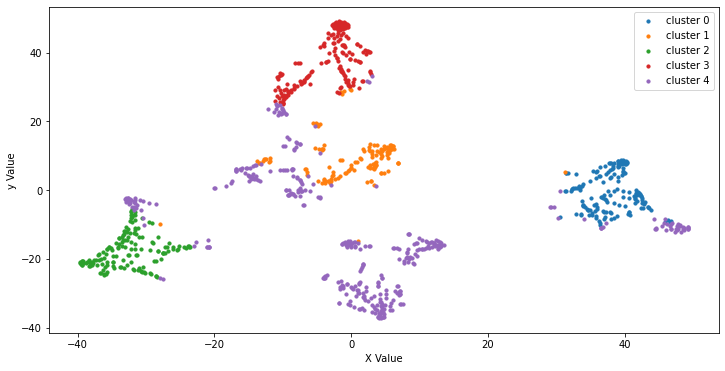
##### Model

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| kappa | 0.72875 |
| coherence | 245.96424 |
| silhouette | 0.34222 |

**Cluster Quality**

**Clusters scatter plot using T-SNE.**



#### word-embedding

##### Selecting the best number of clusters

Elbow, Silhouette methods for word-embedding

Graphical user interface, chart, line chart

Description automatically generated

From the 2 figures the best number of the clusters is 4

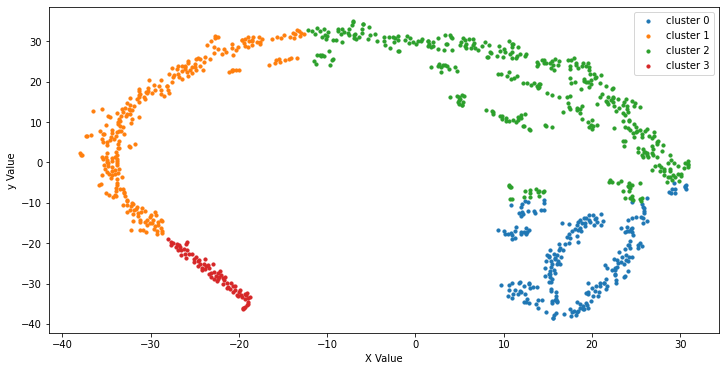
##### Model

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| kappa | 0.180000 |
| coherence | 32.461126 |
| silhouette | 0.470511 |

**Cluster Quality**

**Clusters scatter plot using T-SNE.**



### Gaussian Mixture Model

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. it works based on an algorithm called Expectation-Maximization, or EM.

Because this model takes too much time in training, all the features resulting from different feature extraction methods were compressed using TruncatedSVD algorithm. The selected number of components was determined by calculating the variance of different component numbers and choosing the components that kept the features variance = 0.95, to avoid losing too much information.

#### BOW

##### Selecting the best number of clusters

**The following graphs show the silhouette score for different number of clusters using features acquired from BOW.**

Chart, line chart

Description automatically generated

From the graph we can see that the best number of clusters is 4.

##### Model

**A gaussian mixture model was trained on features from BOW and TruncatedSVD transformations yielding the following clustering results:**

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| **Kappa** | **0.47875** |
| **Silhouette** | **0.0753171425476114** |

**Clusters quality**

**Clusters scatter plot using T-SNE.**

**Chart, scatter chart

Description automatically generated**

#### Tf-IDF

##### Selecting the best number of clusters

**The following graphs show the silhouette score for different number of clusters using features acquired from TF-IDF.**

Chart, line chart

Description automatically generated

From the graph we can see that the best number of clusters is 5.

##### Model

**A gaussian mixture model was trained on features from TF-IDF and TruncatedSVD transformations yielding the following clustering results:**

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| **Kappa** | **0.96125** |
| **Silhouette** | **0.06249725400943369** |

**Clusters quality**

**Clusters scatter plot using T-SNE.**

Chart, scatter chart

Description automatically generated

#### LDA

##### Selecting the best number of clusters

**The following graphs show the silhouette score for different number of clusters using features acquired from LDA.**

**Chart, line chart

Description automatically generated**

From the graph we can see that the best number of clusters is 8.

##### Model

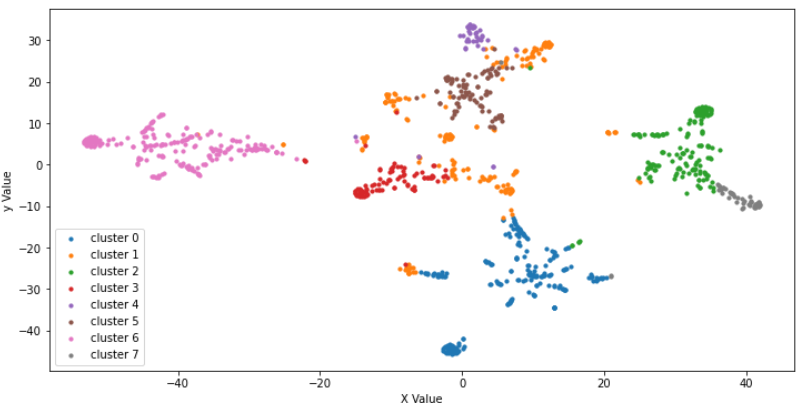
**A gaussian mixture model was trained on features from LDA and TruncatedSVD transformations yielding the following clustering results:**

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| **Kappa** | **0.8362499999999999** |
| **Silhouette** | **0.399243497533306** |

**Clusters quality**

**Clusters scatter plot using T-SNE.**



#### word-embedding

##### Selecting the best number of clusters

**The following graphs show the silhouette score for different number of clusters using features acquired from word-embedding.**

**Chart, line chart

Description automatically generated**

From the graph we can see that the best number of clusters is 4.

##### Model

**A gaussian mixture model was trained on features from word-embedding and TruncatedSVD transformations yielding the following clustering results:**

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| **Kappa** | **0.31375** |
| **Silhouette** | **0.09379908442497253** |

**Clusters quality**

**Clusters scatter plot using T-SNE.**

Chart, scatter chart

Description automatically generated

### Hierarchical Agglomerative Model

Hierarchical Agglomerative is a model in which lower levels are sorted under a hierarchy of successively higher-level units and the data is grouped into clusters at more levels.

AgglomerativeClustering function was used to create the clusters which takes clusters\_Number, then the kappa and Silhouette scores were calculated for the model. Dendrogram method was used to draw the Hierarchical graph.

#### BOW

##### Selecting the best number of clusters

The best number of clusters = 6

Chart, line chart

Description automatically generated

##### Model

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| **Kappa** | 0.57 |
| **Silhouette** | 0.04271552890500441 |

**Clusters quality**

The Hierarchical graph: the scatter plot:

Chart

Description automatically generated Chart, scatter chart

Description automatically generated

#### TF\_IDF:

##### Selecting the best number of clusters

The best number of clusters = 6

Chart, line chart

Description automatically generated

##### Model

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| **Kappa** | 0.98875 |
| **Silhouette** | 0.02250727234524136 |

**Clusters quality**

The Hierarchical graph: The scatter plot:

Chart

Description automatically generated Chart, scatter chart

Description automatically generated

#### word Embedding

##### Selecting the best number of clusters

The best number of clusters = 8

Chart, line chart

Description automatically generated

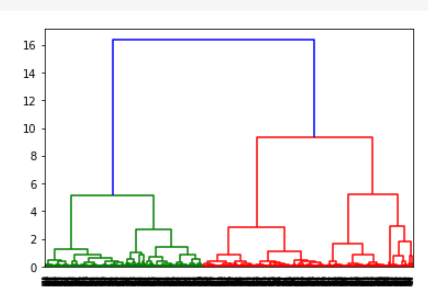
##### Model

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| **Kappa** | 0.17625000000000002 |
| **Silhouette** | 0.4040016233921051 |

**Clusters quality**

The Hierarchical graph: The scatter plot:

 Chart, scatter chart

Description automatically generated

#### LDA

##### Selecting the best number of clusters

The best number of clusters = 16

Chart, line chart

Description automatically generated

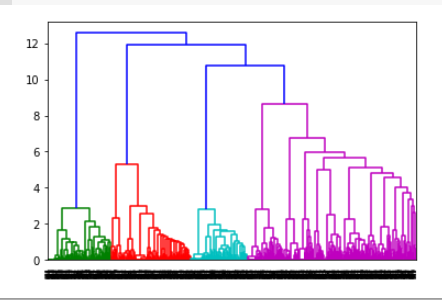
##### Model

**Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| **Kappa** | 0.86625 |
| **Silhouette** | 0.4063031226347956 |

**Clusters quality**

The Hierarchical graph: The Scatter Plot:

 Chart, scatter chart

Description automatically generated

## Models’ comparison

The following table summarizes the previous experiments conducted

Table 1-Different Modeling techniques results comparison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Feature Extraction** | **K value** | **Kappa score** | **Silhouette score** |
| K-Means | BOW | 5 | 0.5825 | 0.0717481628763099 |
|  | TF-IDF | 5 | 0.96125 | 0.0592342330995609 |
|  | LDA | 5 | 0.72875 | 0.3422239906442638 |
|  | Word-Embedding | 4 | 0.1800000000000000 | 0.4705114066600799 |
| Gaussian Mixture | BOW | 4 | 0.47875 | 0.0753171425476114 |
|  | TF-IDF | 5 | 0.96125 | 0.0624972540094336 |
|  | LDA | 8 | 0.836249999999999 | 0.3992434975333 |
|  | Word-Embedding | 4 | 0.4625 | -0.0350672826170921 |
| Hierarchical Agglomerative | BOW | 6 | 0.57 | 0.04271552890500441 |
|  | TF-IDF | 6 | 0.98875 | 0.0225072723452413 |
|  | LDA | 16 | 0.86625 | 0.406303122634795 |
|  | Word-Embedding | 8 | 0.1762500000000000 | 0.404001623392105 |

It is noticed that some of the models had the best number of clusters = 4, which meant that one of the books labels won’t appear in the final results, and indeed for such model the kappa score was quite low.

From the displayed results, it is seen that the model with the highest Kappa score was Hierarchical Agglomerative model using TF-IDF features transformation, but the model with best kappa and silhouette score combination was Hierarchical Agglomerative model using LDA features transformation. Yet the later model had a relatively low kappa score in comparison with other models, so the model with the highest kappa score was selected for the error analysis.

# Champion Model Analysis

The selected champion model was Hierarchical Agglomerative model using TF-IDF features transformation with k value equal to 6.

To analyze the best model, the selected clusters labels were analyzed and compared with original text partition labels. The following partitions were put in clusters that were labeled with a different label.

Graphical user interface, application

Description automatically generated

Figure 6 - Falsely labeled text partitions

It is noticed that many partitions that were originally taken from book with label c, ended up in clusters that contained many partitions from book d. So, the error analysis was performed on those partitions specifically.

Book c category was astronomy, so seeing the word “astronomy” in the incorrectly labeled partition was quite interesting. Book d category was crowd psychology, so it contained multiple different topics.

## Visualizing incorrect predictions

The following visualization are of book c only.

**Wordcloud for incorrect labeled partitions from book c**

**A close-up of a dollar bill

Description automatically generated with medium confidence**

**Ngrams for incorrect labeled partitions from book c**

Chart, bar chart

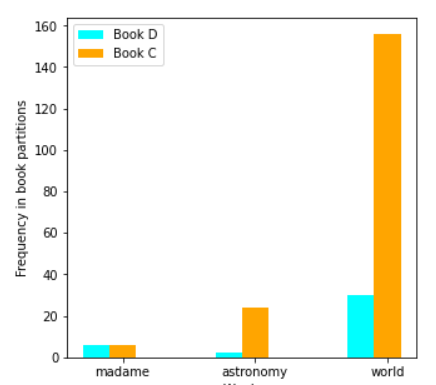
Description automatically generated

From the previous graphs, the following words stood out the most ("madame","astronomy","world")

## Analyzing Incorrect labeled partitions

The analysis focuses more the 3 words ("madame","astronomy","world")

**Printing the selected words frequency in both books’ partitions**



It is noticed that the frequency of the selected words in book C is higher, which is confusing because it means that the text partitions should have been labeled as book C.

**Printing Tf-IDF of the top 20 words in all books’ partitions**

To try and understand why these partitions were added to the wrong clusters, the TF-IDF feature transformation results were printed to check if any of the previously mentioned words -that contained a high importance- had a high IDF weight.

Table

Description automatically generated

It was noticed that indeed the three selected words had high IDF values, all of them were included in the top 20 words that has a high IDF score.

**Printing Tf-IDF of the selected words in book d partitions only**

Graphical user interface

Description automatically generated with medium confidence

**Printing Tf-IDF of the selected words in book c partitions only**

Graphical user interface, application

Description automatically generated

From the previous Tf-IDF results of different books it appears that the specified words had higher scores in the TF-IDF calculation of book d which is why the partitions that contained these words from book c were falsely included in clusters of book d partitions.

# Conclusion

The objective of the assignment was to explore different text clustering models using different feature transformations techniques and different number of clusters. Then check if the resulting clusters truly represented each topic in the books independently without overlapping with topics from other books.

Five books of different authors and different categories were selected to create the text partitions that was used throughout the assignment. Different combinations of feature extraction techniques and clustering models with different values of k were used. The selected champion model was Hierarchical Agglomerative clustering model using TF-IDF features transformation -min\_df equals to 50- with k value equal to 6. The error analysis of the champion model showed that the model was adding partitions of specific book to another books cluster because the terms that appeared in these partitions had a higher IDF weight in the other book.