TEXT CLUSTERING

Team 6

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

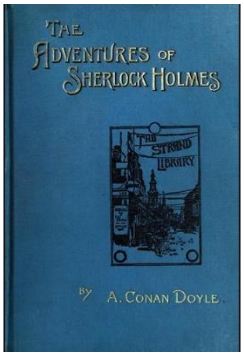
Group members:

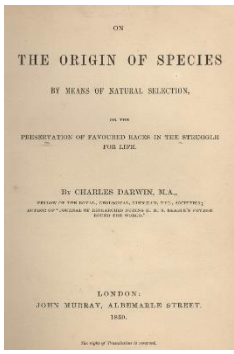
- Esraa Badawi
- Esraa El-Kot
- Salma Sultan
- Sondos Ali

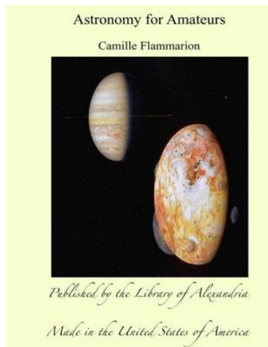
Table of Contents

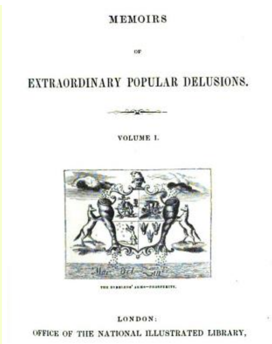
Books Selection

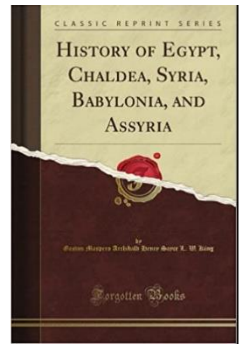
A group of five books were selected from Gutenberg library, each book had a different author but all of them fall under detective and mystery stories category, the labels are used to identify the books partitions throughout the analysis and modeling.











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"

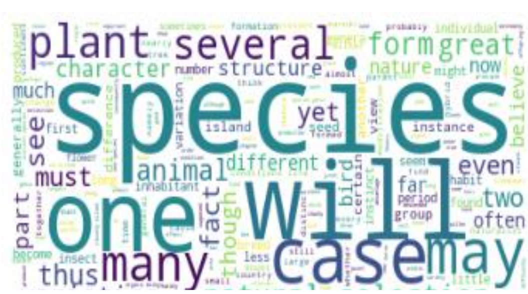


Figure 2-wordcloud for book "b"

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.

	Partition	Label	Partition Text
0	[race, time, romans, pliny, made, curious, dis...	c	race time romans pliny made curious distinctio...
1	[government, various, parts, egypt, course, la...	e	government various parts egypt course large nu...
2	[detects, three, others, times, ancient, greek...	c	detects three others times ancient greeks seve...
3	[pigeon, including, two, three, geographical, ...	b	pigeon including two three geographical races ...
4	[becomes, obvious, domestic, races, show, adap...	b	becomes obvious domestic races show adaptation...
...	...	...	...
995	[breast, unfortunate, youth, learned, read, ho...	d	breast unfortunate youth learned read house ne...
996	[right, hand, sleeve, observed, stained, fresh...	a	right hand sleeve observed stained fresh blood...
997	[culture, fundamentally, earliest, days, egypt...	e	culture fundamentally earliest days egypt rece...
998	[quarter, world, improvement, means, generally...	b	quarter world improvement means generally due ...
999	[us, square, scene, singular, story, listened,...	a	us square scene singular story listened mornin...

1000 rows × 3 columns

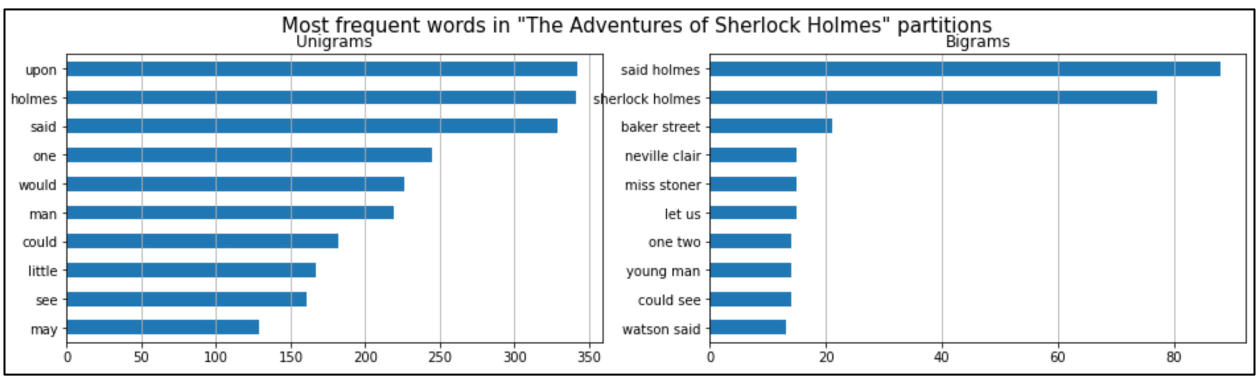
Data exploration and visualization

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.

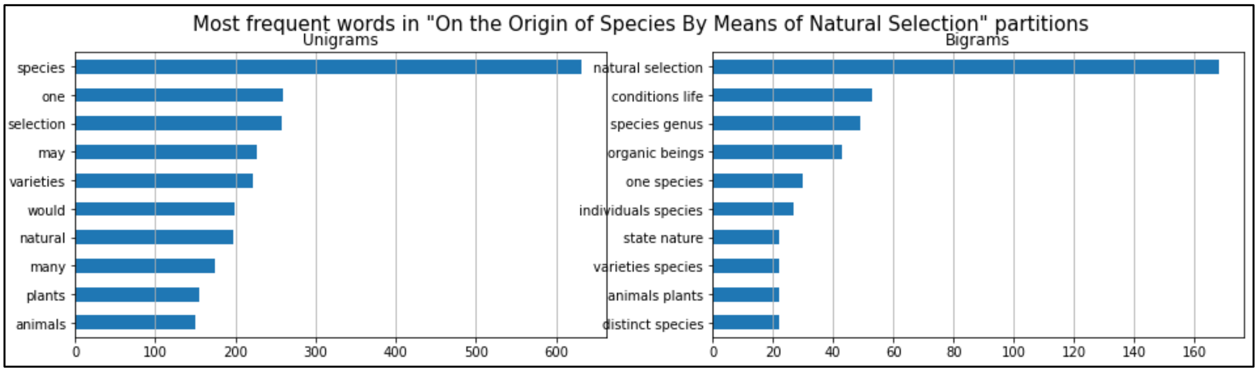
Book 1 partitions

Most Frequent Unigrams and Bigrams for the 1st book partitions



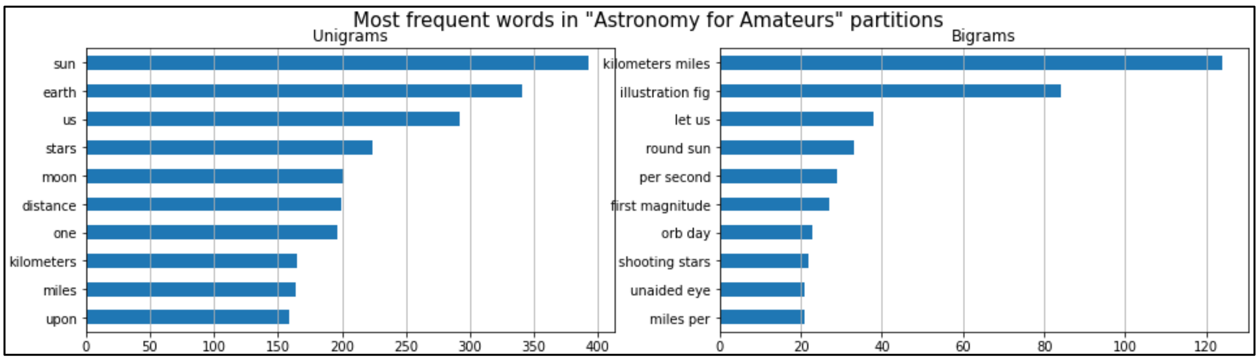
Book 2 partitions

Most Frequent Unigrams and Bigrams for the 2nd book partitions



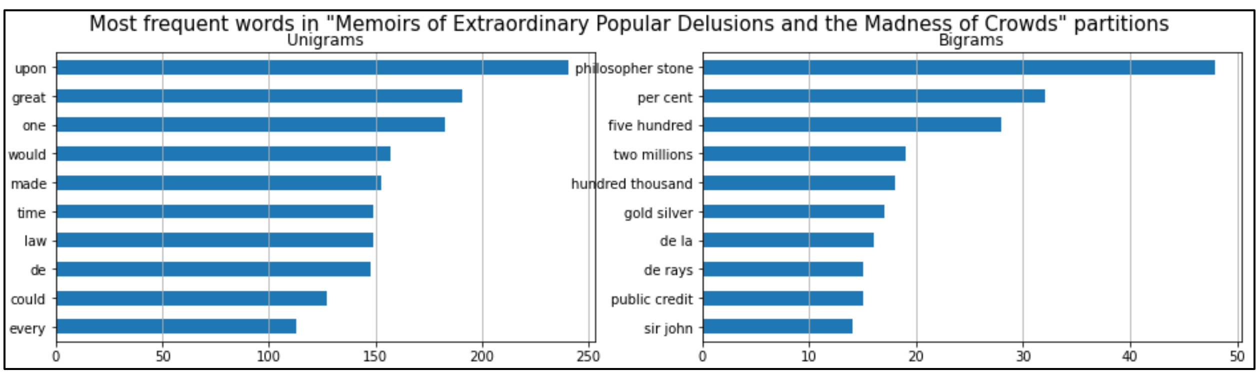
Book 3 partitions

Most Frequent Unigrams and Bigrams for the 3rd book partitions



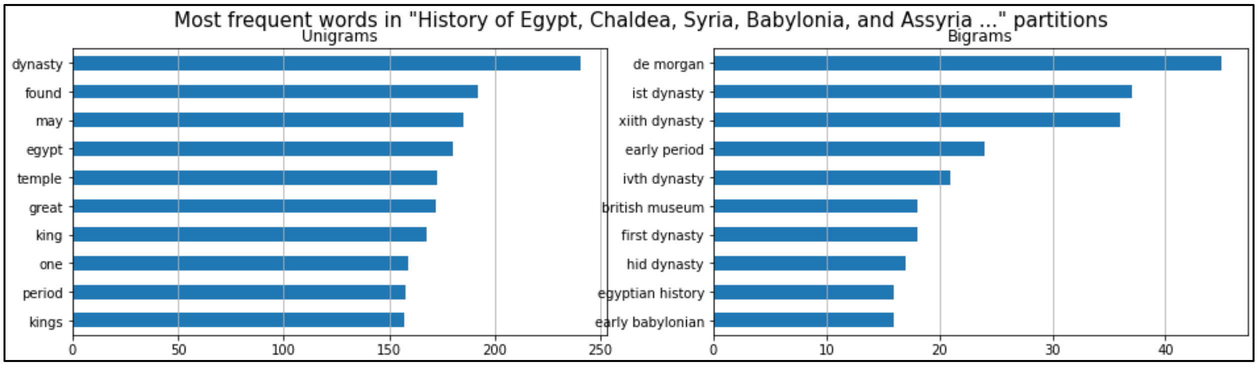
Book 4 partitions

Most Frequent Unigrams and Bigrams for the 4th book partitions



Book 5 partitions

Most 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.

The 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

	Partition	Label	Partition Text	Topic
0	[race, time, romans, pliny, made, curious, dis...	c	race time romans pliny made curious distinctio...	2
1	[government, various, parts, egypt, course, la...	e	government various parts egypt course large nu...	15
2	[detects, three, others, times, ancient, greek...	c	detects three others times ancient greeks seve...	17
3	[pigeon, including, two, three, geographical, ...	b	pigeon including two three geographical races ...	4
4	[becomes, obvious, domestic, races, show, adap...	b	becomes obvious domestic races show adaptation...	4
5	[another, proportion, trenches, cut, deeper, m...	e	another proportion trenches cut deeper mound s...	15
6	[distinguished, species, firstly, discovery, i...	b	distinguished species firstly discovery interm...	4
7	[flints, could, found, desert, surface, beadne...	e	flints could found desert surface beadnell geo...	11
8	[inhabitants, thus, jostle, closely, shall, ge...	b	inhabitants thus jostle closely shall general ...	4
9	[dare, conceive, things, really, mere, commonp...	a	dare conceive things really mere commonplaces ...	12

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

	Topic	0	1	2	3	4	5	6	7	8	9
0	0	de	hybrids	great	john	upon	house	sterility	caricature	print	time
1	1	one	times	would	distance	upon	planet	moon	us	earth	sun
2	2	may	comet	might	angle	like	two	year	degree	great	one
3	3	much	beak	great	tumbler	carrier	marks	fantail	tail	several	breeds
4	4	would	many	animals	plants	natural	may	one	varieties	selection	species
5	5	second	distance	us	first	kilometers	magnitude	miles	one	star	stars
6	6	salesman	les	de	new	much	upon	witness	geese	would	time
7	7	stock	made	great	time	hundred	company	one	law	would	upon
8	8	well	see	little	could	would	man	one	upon	said	holmes
9	9	albert	basil	relative	rule	applies	left	light	marks	view	length
10	10	law	several	still	man	like	praying	day	petition	every	time
11	11	tombs	two	great	tomb	royal	buried	found	abydos	kings	dynasty
12	12	hand	away	carried	goddess	like	god	one	man	temple	upon
13	13	gold	would	man	philosopher	one	could	made	great	de	upon
14	14	two	north	origin	semitic	king	time	may	egyptian	egypt	dynasty
15	15	egypt	babylonia	babylonian	later	prehistoric	de	history	period	found	early
16	16	might	modifications	one	thus	structure	many	could	pollen	flower	would
17	17	elamite	babylon	upon	life	reign	may	time	one	king	elam
18	18	like	king	raymond	upon	england	man	said	make	great	time
19	19	mound	may	gods	great	shirpurla	god	city	gudea	ningirsu	temple

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.

```
warnings.warn(msg, category=FutureWarning)
```

	ab	abadiya	abandon	abandoned	abandoning	abandons	abated	abbey	abbot	abbots	...	zèle	âge	æsculapius	égal	égypte	équipage	étoiles	êtes	ûn	œuvre
Part0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Part1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Part2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Part3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Part4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Part995	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	0	0	0	0
Part996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Part997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Part998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Part999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

1000 rows × 16876 columns

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.

```
abadiya          7.215608
abandon          7.215608
abandoning       7.215608
abated           7.215608
abbots           7.215608
                   ...
champignelle     7.215608
chancellorship   7.215608
chandeliers      7.215608
chandos          7.215608
channels         7.215608
Name: idf_weights, Length: 1000, dtype: float64
```

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 less than 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 less than 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.

```
documents = [TaggedDocument(doc, [i]) for i, doc in enumerate (books_df['Partition'])]
doc2vec_model = Doc2Vec(documents, min_count=1)
fname = get_tmpfile("my_doc2vec_model")
doc2vec_model.save(fname)
doc2vec_model = Doc2Vec.load(fname)
```

3. Use the model to generate vectors.

```
doc2vec_vectors = []
for i in range(len(books_df['Partition'])):
 vector = doc2vec_model.infer_vector(books_df['Partition'].iloc[i])
 doc2vec_vectors.append(vector)
```

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.

```
doc2vec_vectors[0]
```

```
array([ 0.1397138 ,  0.08637278,  0.1534169 , -0.02689473,  0.37527543,
       -0.01685197,  0.04189474, -0.41312414,  0.1763254 ,  0.02248875,
       -0.08367461, -0.00385659, -0.21243724,  0.10165562, -0.00554188,
        0.34253156,  0.10246495,  0.11122139,  0.07921384,  0.06346048,
       -0.18544436, -0.06609705, -0.01641821,  0.1211511 ,  0.35922837,
        0.10716776, -0.06577618, -0.03069227,  0.12740114,  0.21664979,
       -0.17482258, -0.20687525,  0.37692708,  0.04827308, -0.00648702,
        0.0350808 , -0.51152986,  0.2604146 ,  0.03183403,  0.3246265 ,
        0.17696972,  0.00060876,  0.27790335,  0.2563983 ,  0.23618615,
        0.04593148,  0.14874554,  0.12717517,  0.12120364, -0.02909256,
        0.299997  ,  0.04510668,  0.17727232,  0.08787717, -0.16179189,
        0.01306718,  0.07835363, -0.29701355,  0.06348787,  0.32631257,
        0.13043071, -0.2563765 , -0.14969836,  0.15814084,  0.19442333,
        0.07983616, -0.01286518,  0.18156892,  0.14005491, -0.06711028,
       -0.08630536, -0.19704445, -0.23880199, -0.0424545 , -0.16171768,
       -0.37920815, -0.39778635,  0.14389579, -0.07072361, -0.15644681,
        0.18002489,  0.11170616, -0.52616155,  0.04355256,  0.07171201,
        0.34777144, -0.17383263,  0.11914935, -0.00851282, -0.26481962,
        0.12999554,  0.21416828, -0.10922162,  0.11713382, -0.05063432,
       -0.17773424,  0.15133905, -0.5085488 , -0.0704578 , -0.06746372],
      dtype=float32)
```

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.

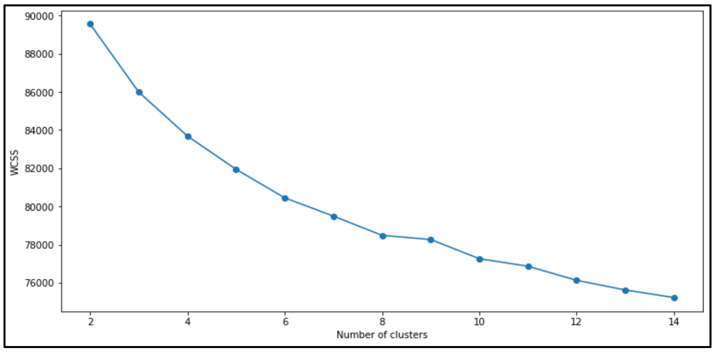
K-Means Model

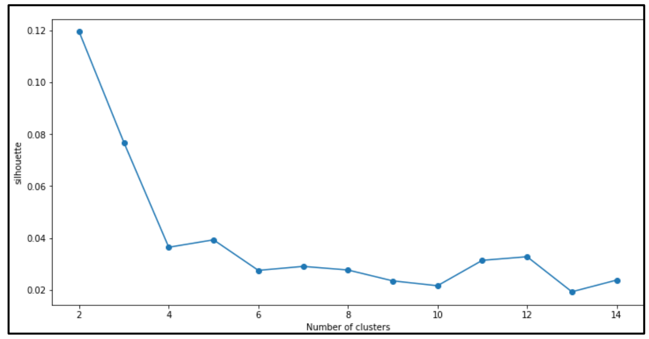
K-Means Clustering is an unsupervised learning technique that divides an unlabeled dataset into clusters.

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

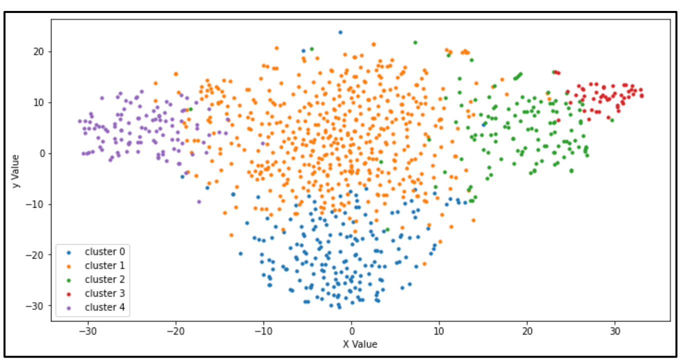
A pipeline is created. First, apply BOW transformer, then add the result to our estimator which is k means. After that, we fit the pipeline, apply evaluation by calculating the kappa, coherence, and silhouette.

Evaluation

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

Cluster Quality

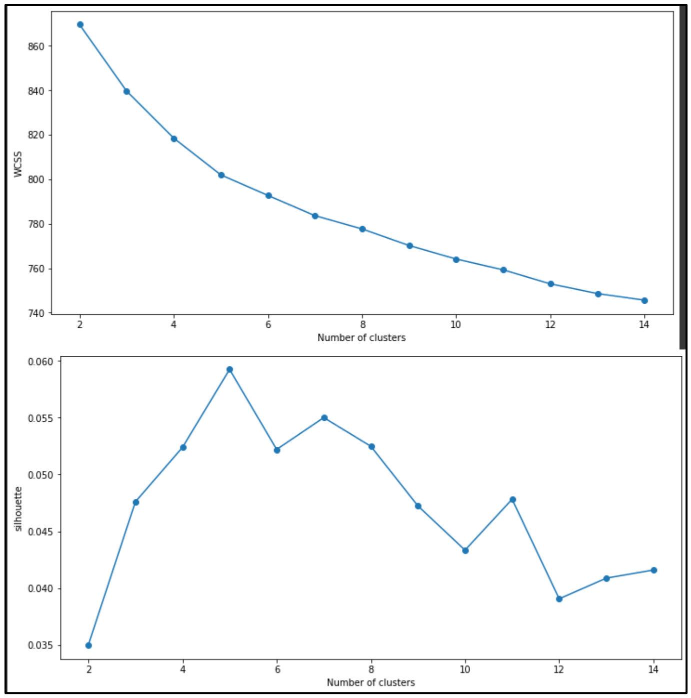
Clusters scatter plot using T-SNE.



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.



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

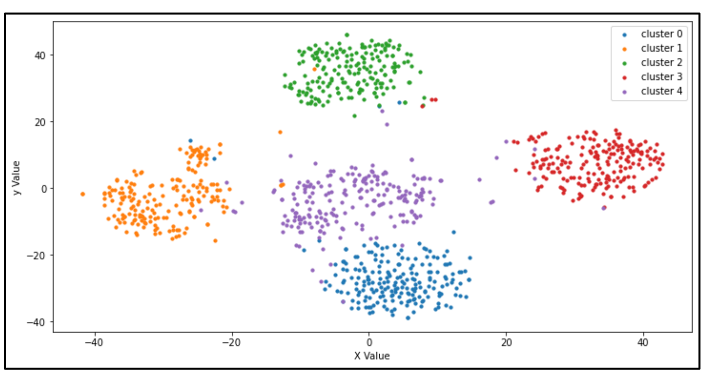
Model

Evaluation

Metric	Score
Kappa	0.96125
Coherence	881.95255
silhouette	0.05923

Cluster Quality

Clusters scatter plot using T-SNE.

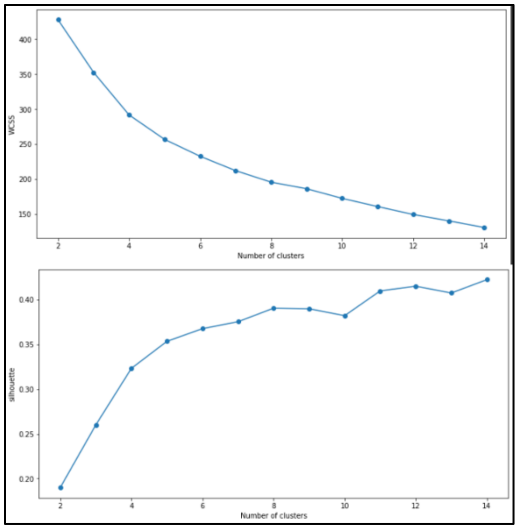


LDA

Selecting the best number of clusters

Elbow, Silhouette methods for LDA

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



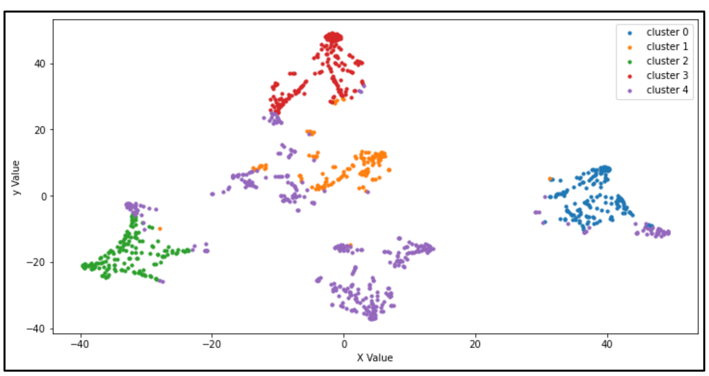
Model

Evaluation

Metric	Score
kappa	0.72875
coherence	245.96424
silhouette	0.34222

Cluster Quality

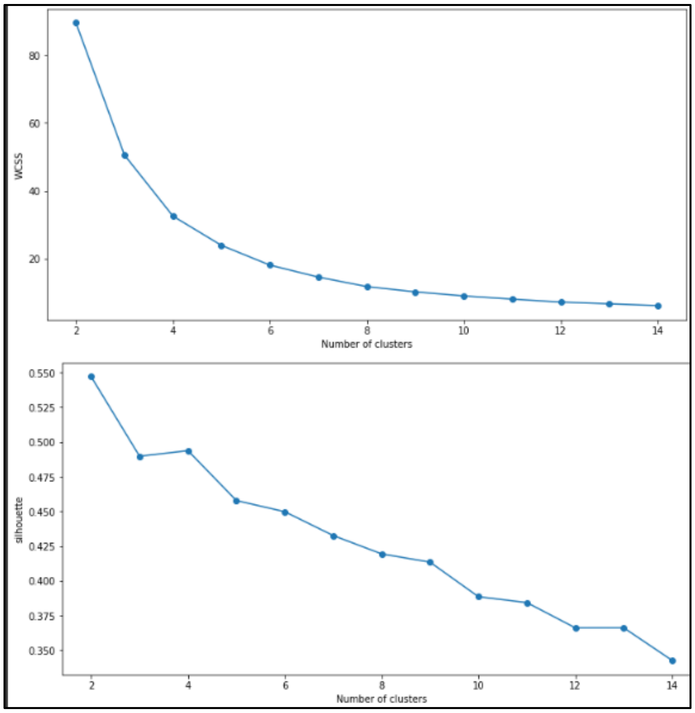
Clusters scatter plot using T-SNE.



word-embedding

Selecting the best number of clusters

Elbow, Silhouette methods for word-embedding



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

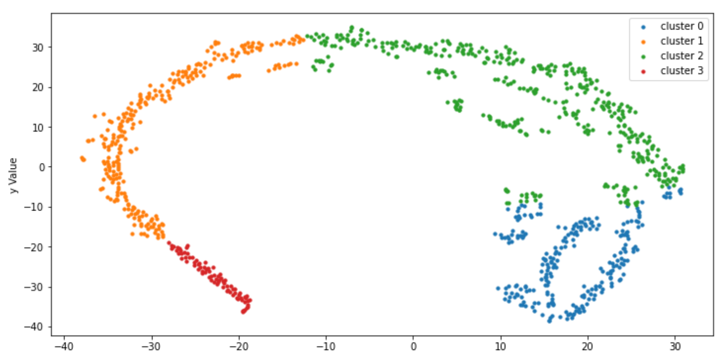
Model

Evaluation

Metric	Score
kappa	0.180000
coherence	32.461126
silhouette	0.470511

Cluster Quality

Clusters scatter plot using T-SNE.



Gaussian Mixture Model

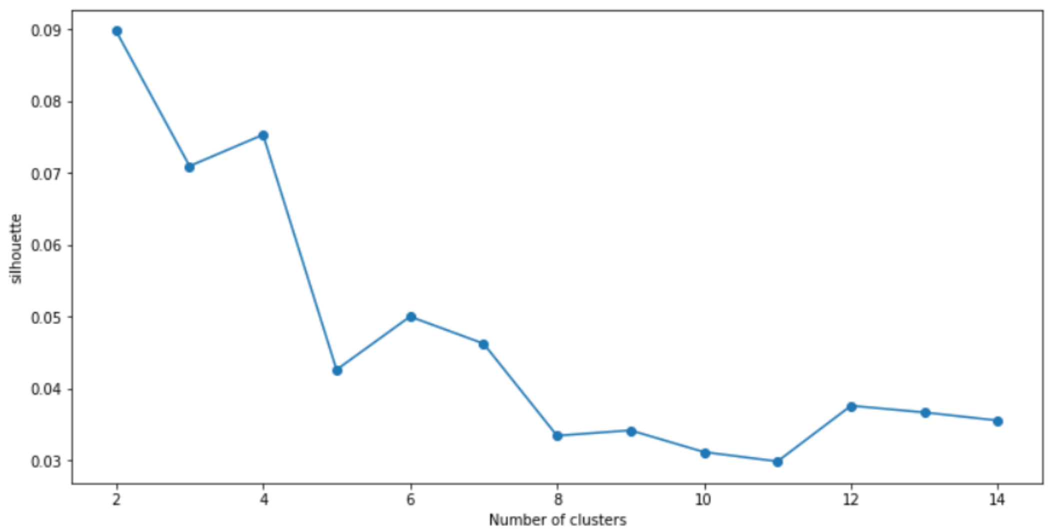
A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. it works based on an algorithm called Expectation-Maximization, or EM.

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.



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

Model

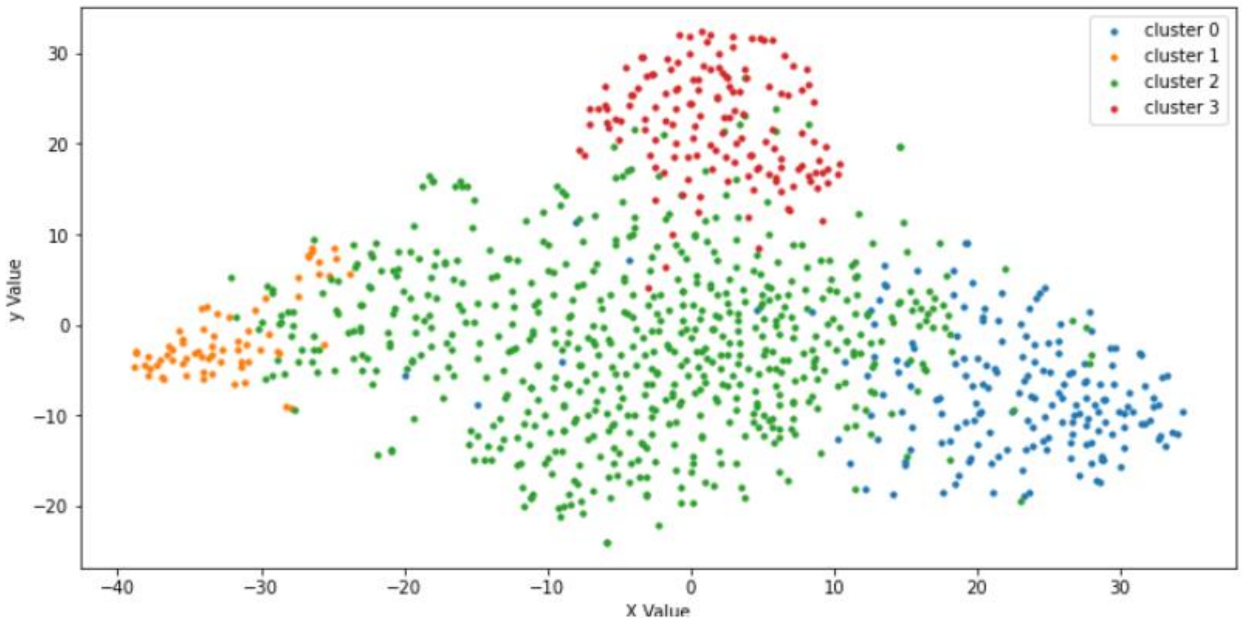
A gaussian mixture model was trained on features from BOW and TruncatedSVD transformations yielding the following clustering results:

Evaluation

Metric	**Score**
Kappa	0.47875
Silhouette	0.0753171425476114

Clusters quality

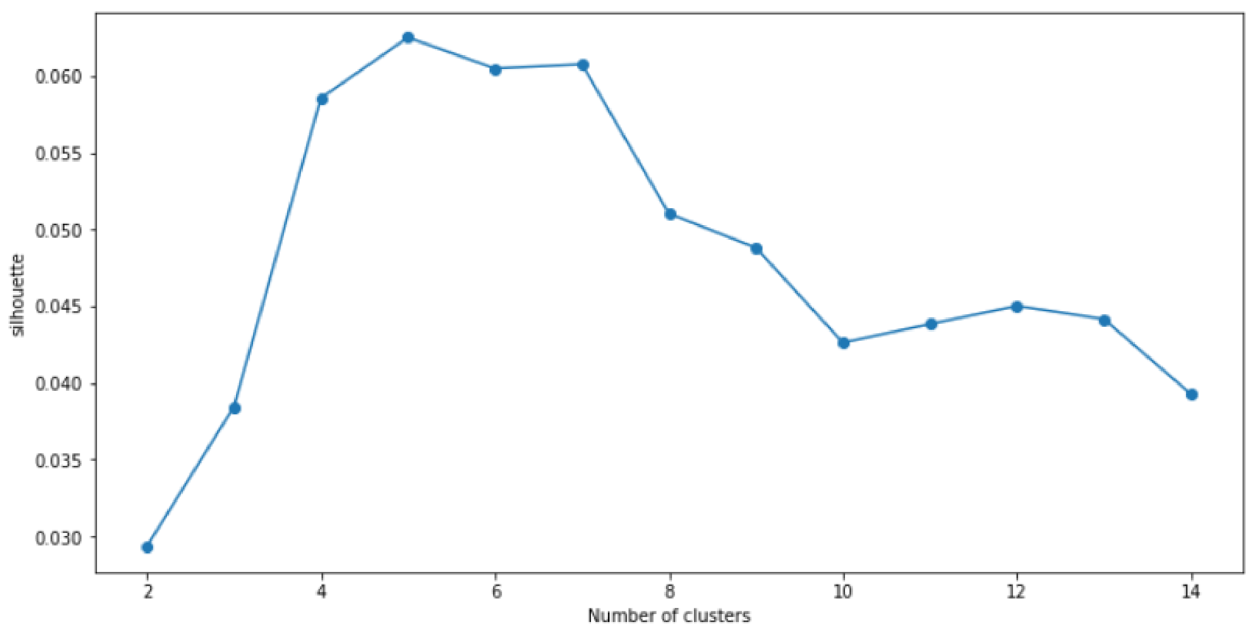
Clusters scatter plot using T-SNE.



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.



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

Model

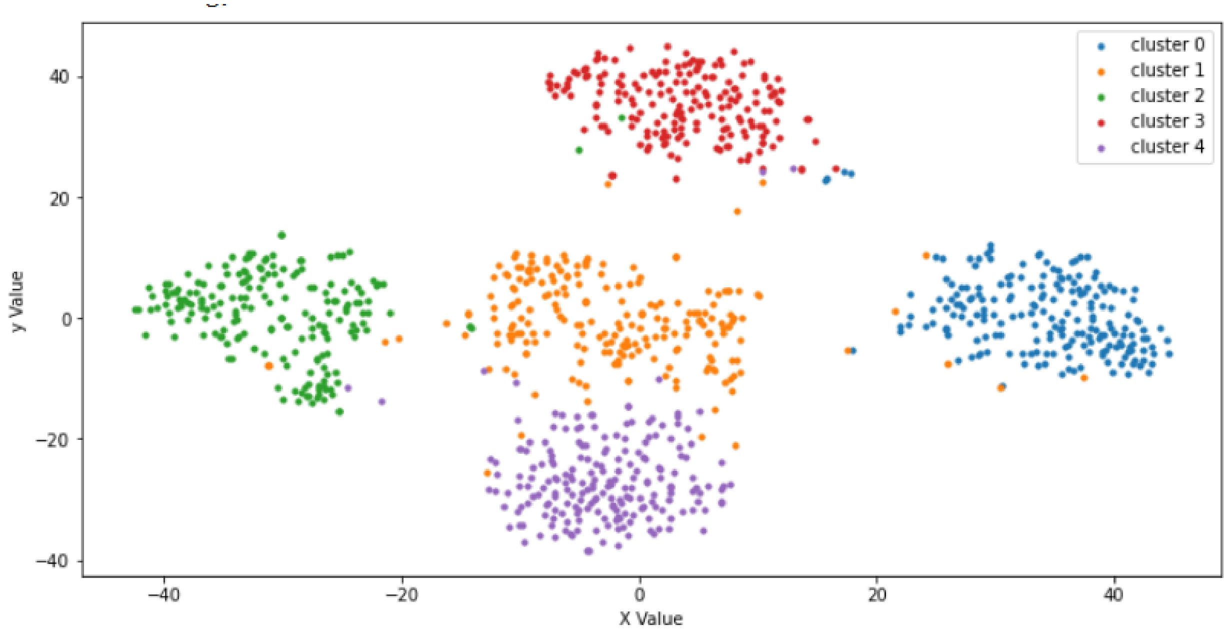
A gaussian mixture model was trained on features from TF-IDF and TruncatedSVD transformations yielding the following clustering results:

Evaluation

Metric	Score
Kappa	0.96125
Silhouette	0.06249725400943369

Clusters quality

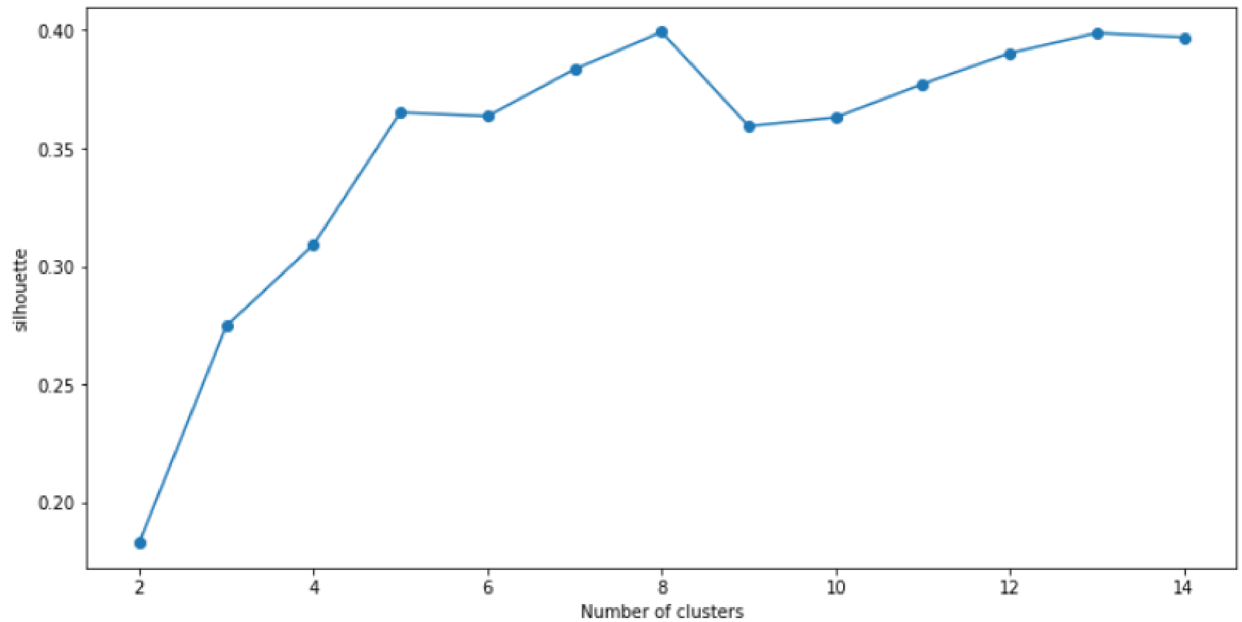
Clusters scatter plot using T-SNE.



LDA

Selecting the best number of clusters

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



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

Model

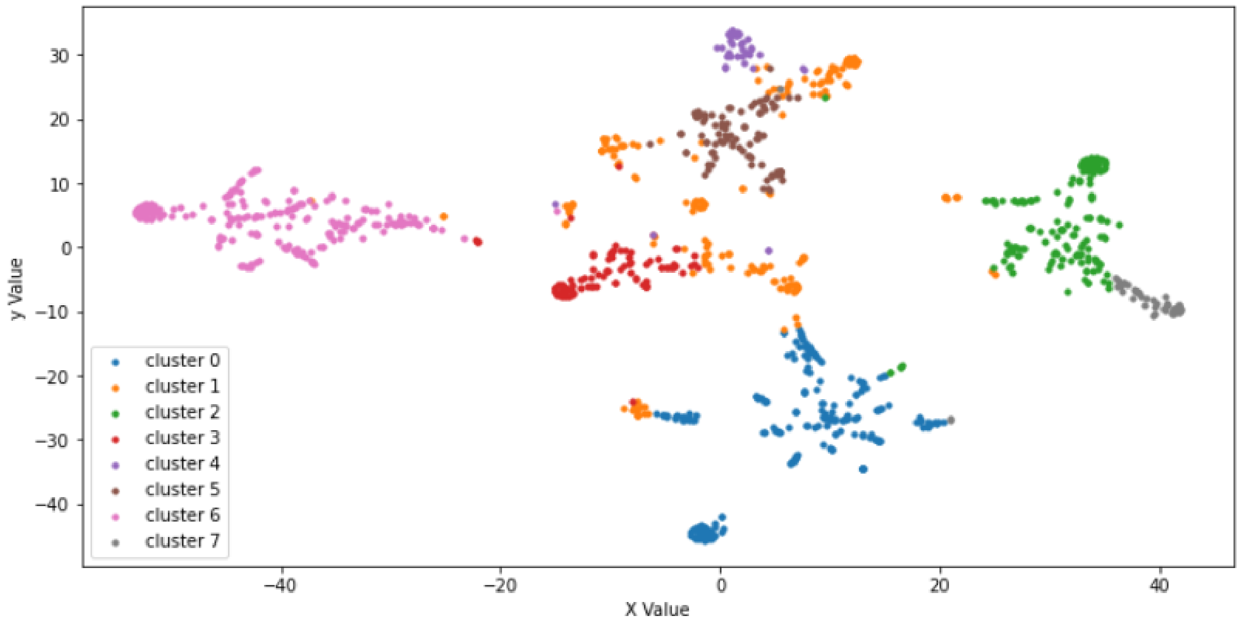
A gaussian mixture model was trained on features from LDA and TruncatedSVD transformations yielding the following clustering results:

Evaluation

Metric	**Score**
Kappa	0.8362499999999999
Silhouette	0.399243497533306

Clusters quality

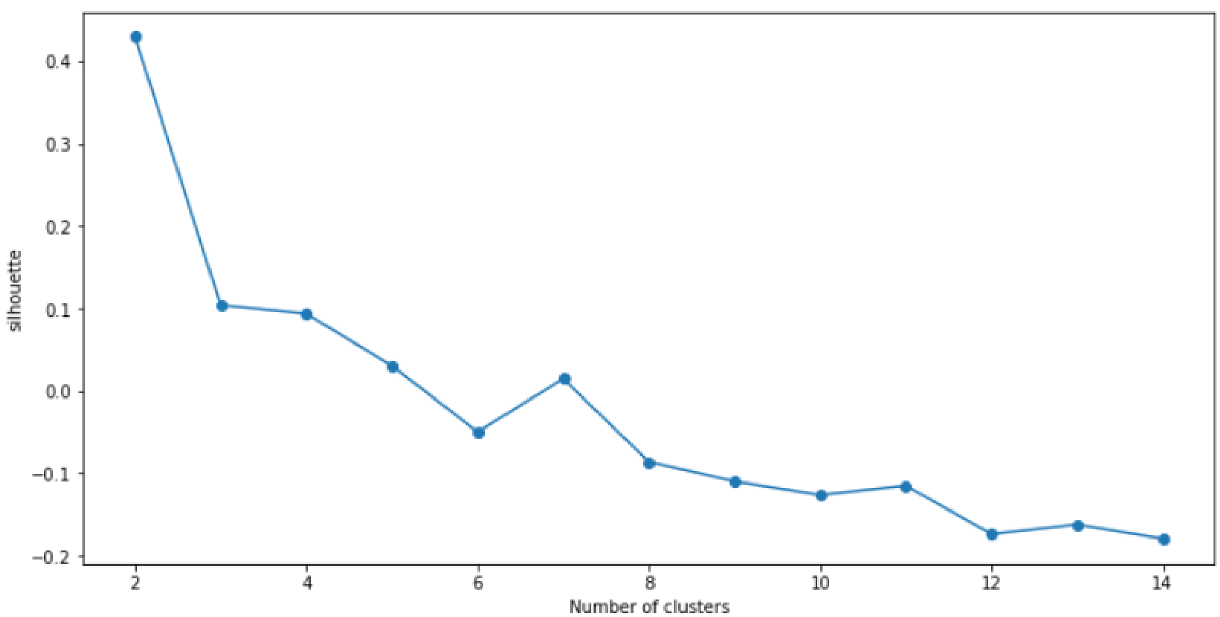
Clusters scatter plot using T-SNE.



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.



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

Model

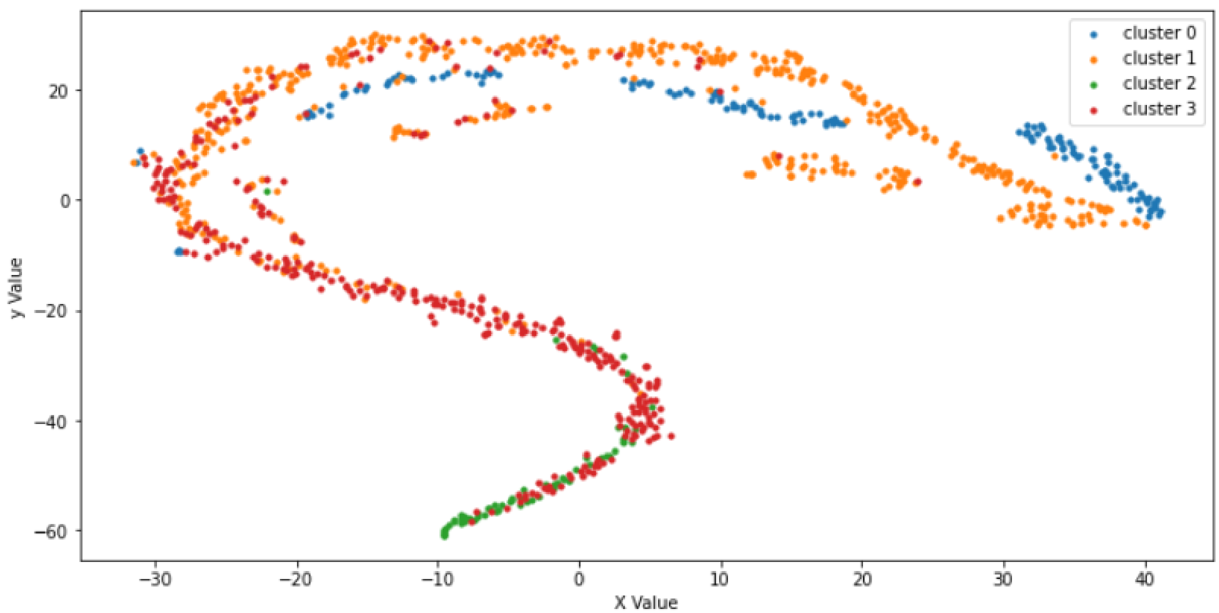
A gaussian mixture model was trained on features from word-embedding and TruncatedSVD transformations yielding the following clustering results:

Evaluation

Metric	Score
Kappa	0.31375
Silhouette	0.09379908442497253

Clusters quality

Clusters scatter plot using T-SNE.



Hierarchical Agglomerative Model

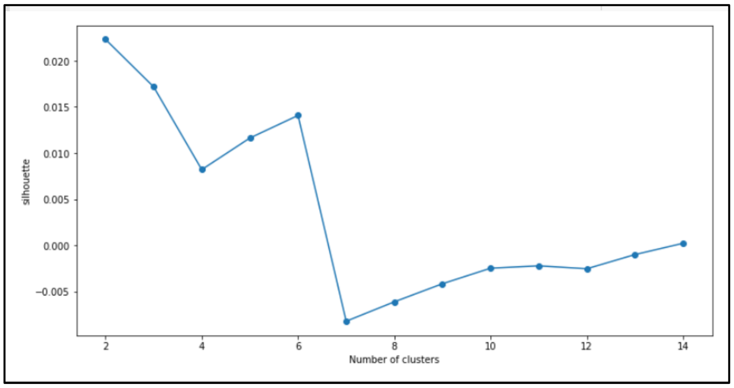
Hierarchical Agglomerative is a model in which lower levels are sorted under a hierarchy of successively higher-level units and the data is grouped into clusters at more levels.

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



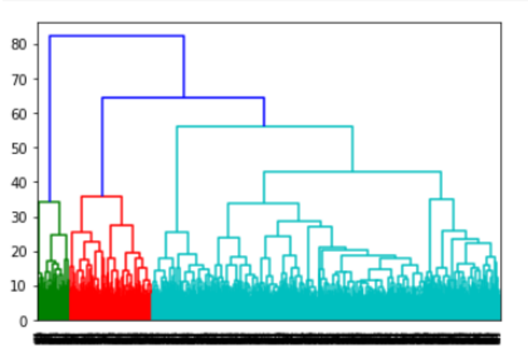
Model

Evaluation

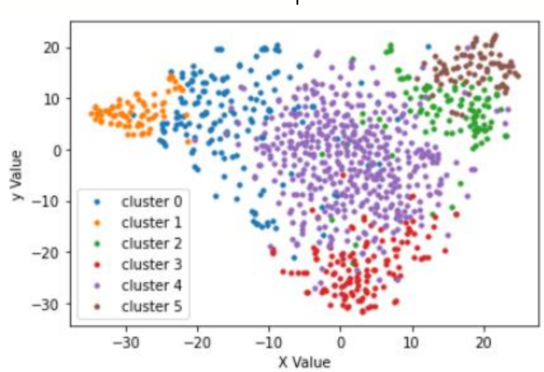
Metric	Score
Kappa	0.57
Silhouette	0.04271552890500441

Clusters quality

The Hierarchical graph:



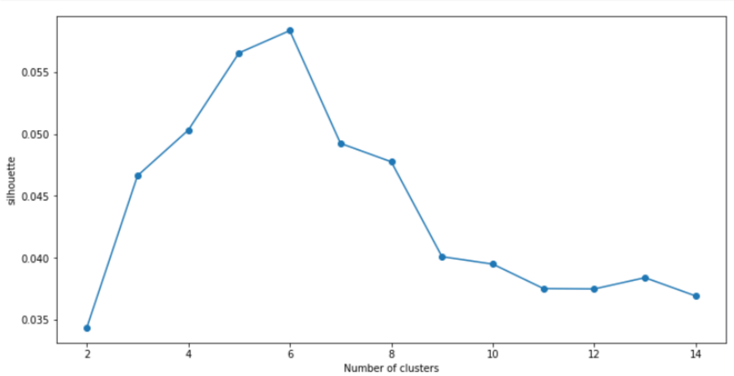
the scatter plot:



TF_IDF:

Selecting the best number of clusters

The best number of clusters = 6



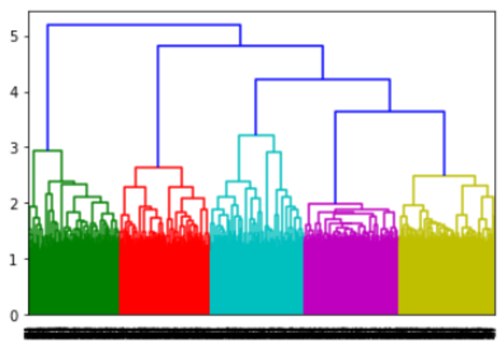
Model

Evaluation

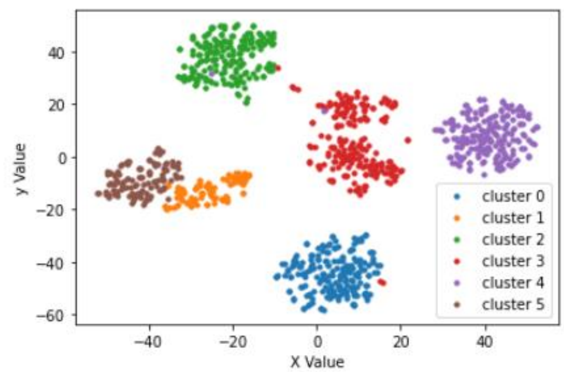
Metric	Score
Kappa	0.98875
Silhouette	0.02250727234524136

Clusters quality

The Hierarchical graph:



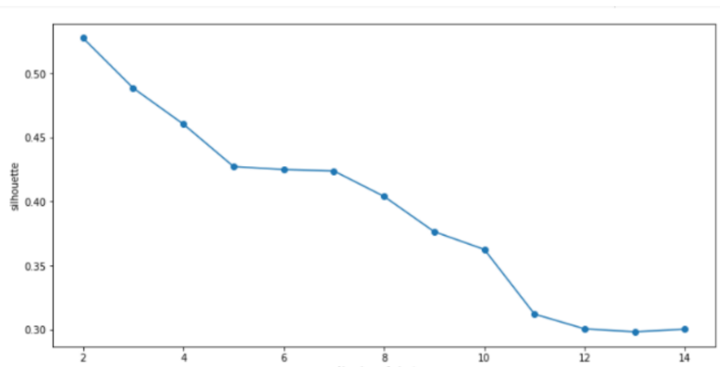
The scatter plot:



word Embedding

Selecting the best number of clusters

The best number of clusters = 8



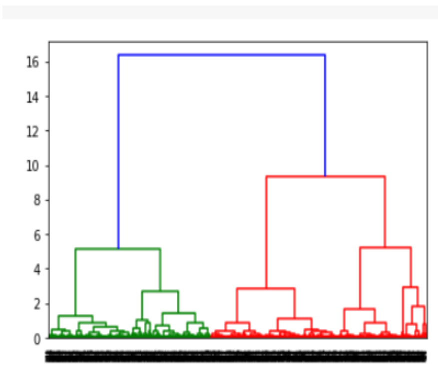
Model

Evaluation

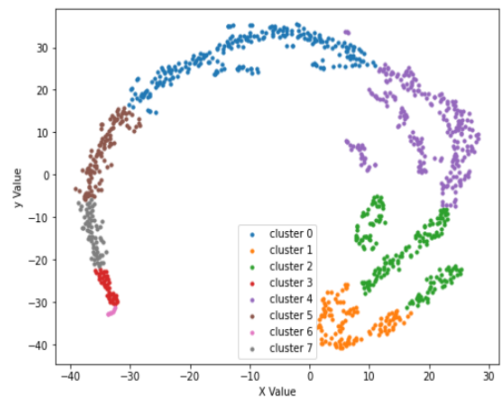
Metric	Score
Kappa	0.17625000000000002
Silhouette	0.4040016233921051

Clusters quality

The Hierarchical graph:



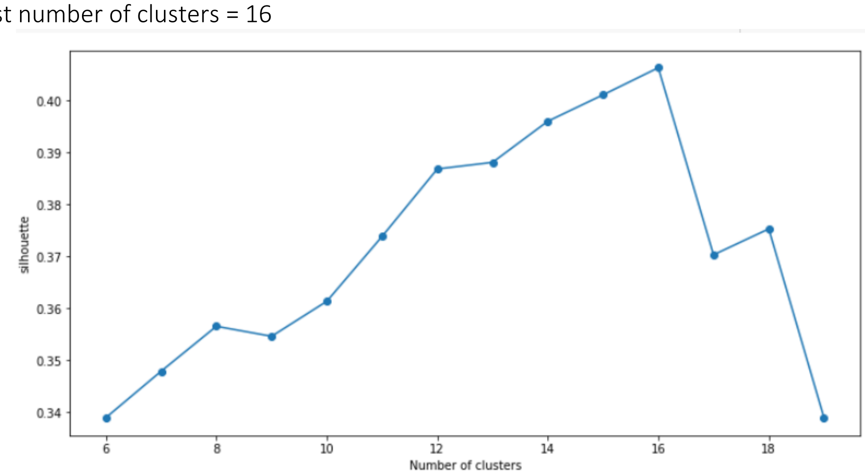
The scatter plot:



LDA

Selecting the best number of clusters

The best number of clusters = 16



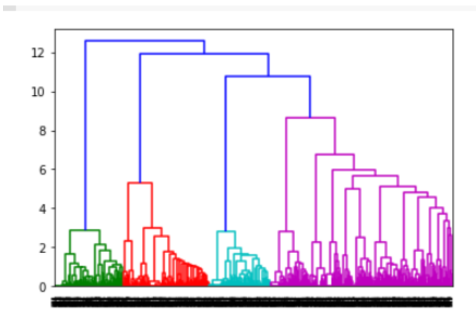
Model

Evaluation

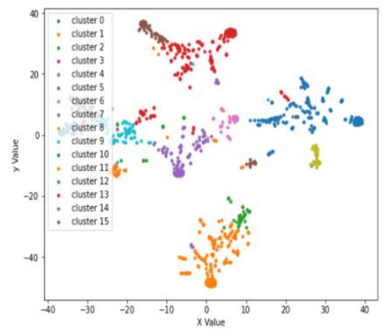
Metric	Score
Kappa	0.86625
Silhouette	0.4063031226347956

Clusters quality

The Hierarchical graph:



The Scatter Plot:



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.

	Partition	Label	Partition Text	Topic	Cluster Label
34	[geometry, algebra, astronomy, author, three, ...	c	geometry algebra astronomy author three works ...	2	d
117	[darkens, atmosphere, becomes, sad, dull, anti...	c	darkens atmosphere becomes sad dull anticipati...	7	a
161	[general, conclusions, origin, species, last, ...	b	general conclusions origin species last year s...	19	d
448	[individual, embryology, laws, explained, vari...	b	individual embryology laws explained variation...	4	d
624	[sleep, better, seen, obvious, indisputable, t...	c	sleep better seen obvious indisputable truths ...	1	d
682	[college, time, ecclesiastical, establishment,...	c	college time ecclesiastical establishment clos...	19	d
714	[bosom, neckerchief, blown, aside, wind, fit, ...	d	bosom neckerchief blown aside wind fit inspira...	13	a
810	[permission, copy, order, preserve, model, bes...	c	permission copy order preserve model best natu...	6	d
900	[astronomy, diffuses, light, truth, within, us...	c	astronomy diffuses light truth within us poeti...	6	d

Figure 6 - Falsely labeled text partitions

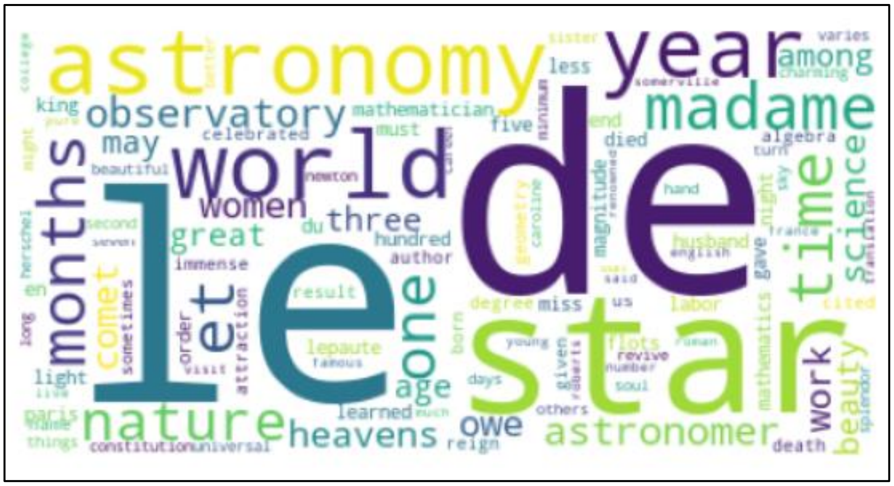
It is noticed that many partitions that were originally taken from book with label c, ended up in clusters that contained many partitions from book d. So, the error analysis was performed on those partitions specifically.

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.

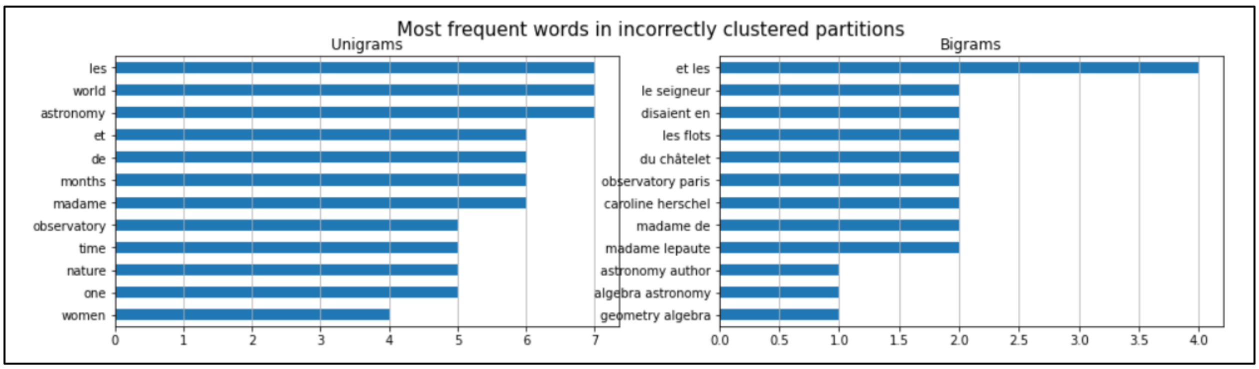
Visualizing incorrect predictions

The following visualization are of book c only.

Wordcloud for incorrect labeled partitions from book c



Ngrams for incorrect labeled partitions from book c

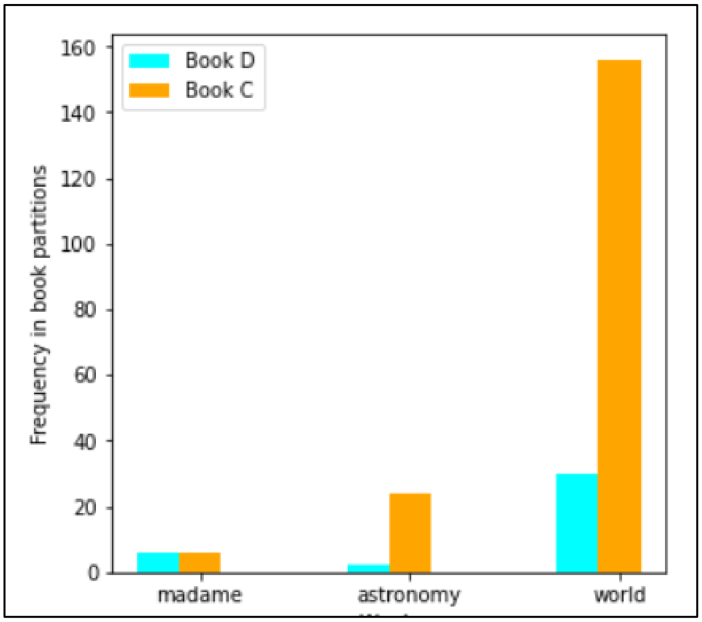


From the previous graphs, the following words stood out the most ("madame" , "astronomy" , "world")

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.

	idf_weights
one	1.467808
time	2.017111
nature	2.778856
world	2.858899
de	3.129631
months	3.976929
astronomy	4.730701
women	4.864232
observatory	5.136166
madame	5.423848
et	5.606170
les	6.116995

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

	idf_weights
world	3.007468
madame	4.357395
astronomy	5.204693

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

	idf_weights
world	1.985817
astronomy	3.258782
madame	4.693867

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.