Text clustering Assignment

Data Science Applications DTI5125[EG]

Group name: DSA_202101_ 8

Amr Ashraf Mahmoud Elsherbiny

Ahmed Fares Saad Eldin Khalifa

Amira EssamEldin Ibrahim Ibrahim Elgebaly

Abdallah Medhat Mohamed Rashed

Contents

1. Objectives 1
2. Requirements 2
3. Methodology 2
3.1. Data Preparation 2
3.2. Data Exploration 2
3.3. Data Preprocessing 3
3.4. Feature Engineering 3
a) Bag of Words 3
b) Term Frequency-Inverse Document Frequency 8
c) Doc2Vec 11
d) Latent Dirichlet Allocation 13
3.5. Performance Evaluation 14
3.6. Champion Model 15
3.7. Error Analysis 15

Figure 1: Reading books and generate partitions of 150 words ..... **Error! Bookmark not defined.**
Figure 2: Dale-Chall Readability of the 5 Books 3
Figure 3: Sentiment Scores of the 5 Books 3
Figure 4: word count per partition after cleaning ..... **Error! Bookmark not defined.**
Figure 5: BOW with Hierarchical Clustering 5
Figure 6: BOW with K-Means Clustering 6
Figure 7: BOW with EM Clustering 7
Figure 8: TF-IDF with hierarchical Clustering 8
Figure 9: TF-IDF with K-means Clustering 9
Figure 10: TF-IDF with EM Clustering 10
Figure 11: doc2vec with hierarchical Clustering 11
Figure 12: doc2vec with K-means Clustering 12
Figure 13: doc2vec with EM Clustering 13
Figure 14: Misclassified Examples 15
Figure 15: Composition of Keyword Frequencies in Misclassified Examples from Household Management Book 16
Figure 16: Composition of Keyword Frequencies in Misclassified Examples from Household Management Book 16
Figure 17: Keyword Frequencies in Household Management Correct Predictions 17
Figure 18: Keyword Frequencies in Household Management Correct Predictions 17
Figure 19: Knowledge Graph of Shared Words 18

1. Objectives

The overall objective is to cluster partitions of different books from the different genres (different topics), use different transformations and models to cluster these partitions, analyze the pros and cons of algorithms, and finally generate and communicate insights.

2. Requirements

Using a random sample of size 5 of different books written by different authors from Gutenberg Digital Library, split the data into training partitions. After that, try different transformation techniques and clustering methods to accurately cluster these partitions.

3. Methodology

3.1. Data Preparation

We start by loading the books from URLs in Gutenberg library website. After that, we proceed by removing the header and footer of Gutenberg library from each book. We then create random samples of 200 partitions from each book. As we sample before cleaning to avoid losing important features, we prepare the records of 150 words records for each document, label them by book name genre. as per the book they belong to.

3.2. Data Exploration

After loading and sampling from the 5 books, we explore various features of the raw data to get a better understanding of the structure of each book and the author's writing style. First, we compute the Dale-Chall readability of each of the 5 books. Dale-Chall is a measure of how comprehensible a text is, by checking how many difficult words it contains. It then computes a score that can be used to tell what level of education would be needed to understand this text. Figure 2 shows how readable are the partitions of the 5 books. We can observe that "The Rat Race Book" contains many partitions that can be comprehended by a 9-10th grader, making it the simplest text. In addition to that, we employ a sentiment analysis model to calculate how positive or negative partitions are. As shown in Figure 3, we can observe that most partitions from the Music book are positive sentiments.

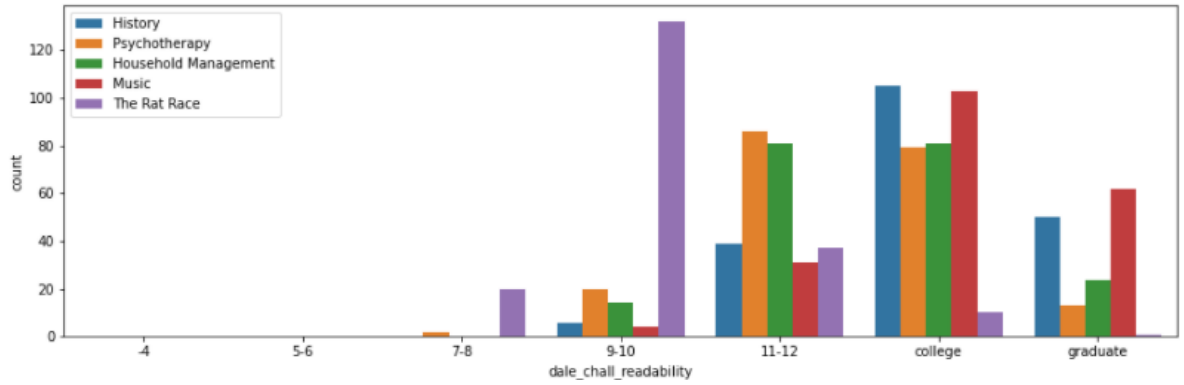


Figure 1: Dale-Chall Readability of the 5 Books

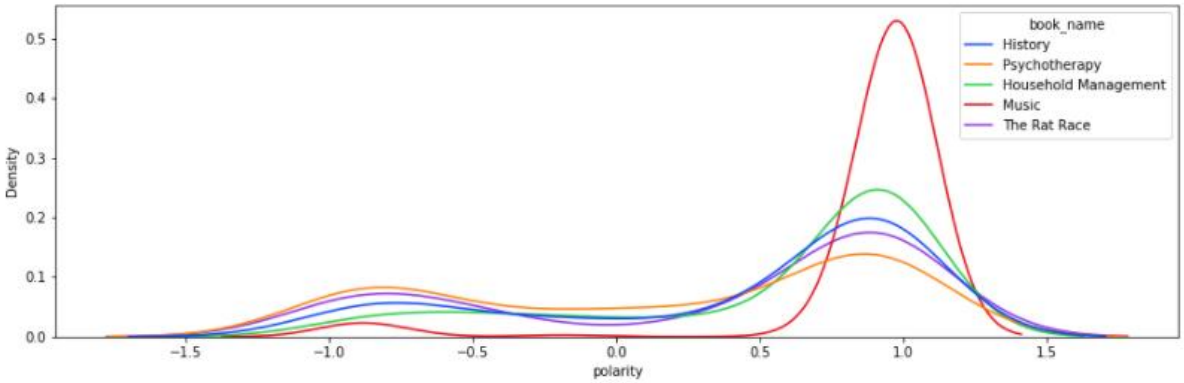


Figure 2: Sentiment Scores of the 5 Books

3.3. Data Preprocessing

We start by removing stop words and punctuations by regular expression, which are the noise in the text. In addition to that, we normalize text to words by using Stemming which reduces words to their word root.

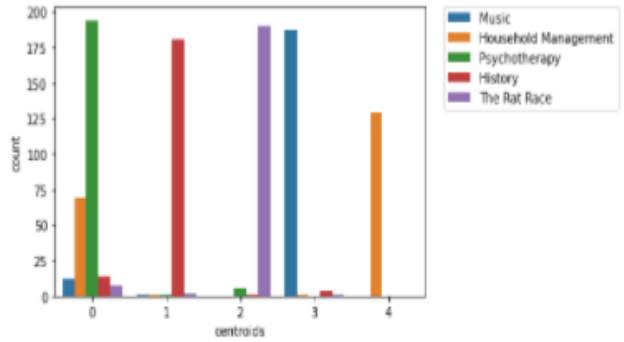
3.4. Feature Engineering

Feature Engineering is important because in the problem of clustering of texts, we have a series of texts and their corresponding labels. But we directly can't use text for our model. We must convert that text into some numbers or number vectors. And we applied algorithms on every transformation model with variant features which affect the accuracy, so when we choose low number of max features, we get low accuracy.

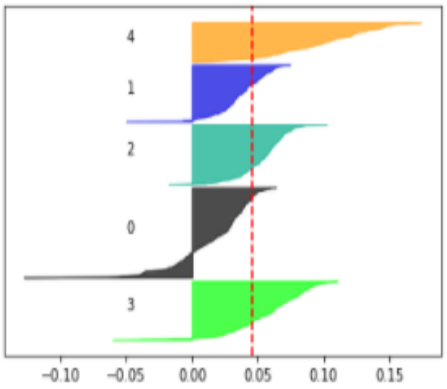
a) Bag of Words

The first transformation we apply is bag of words. It basically runs over the whole dataset, constructs its vocabulary of predefined size, and then transforms each sentence to

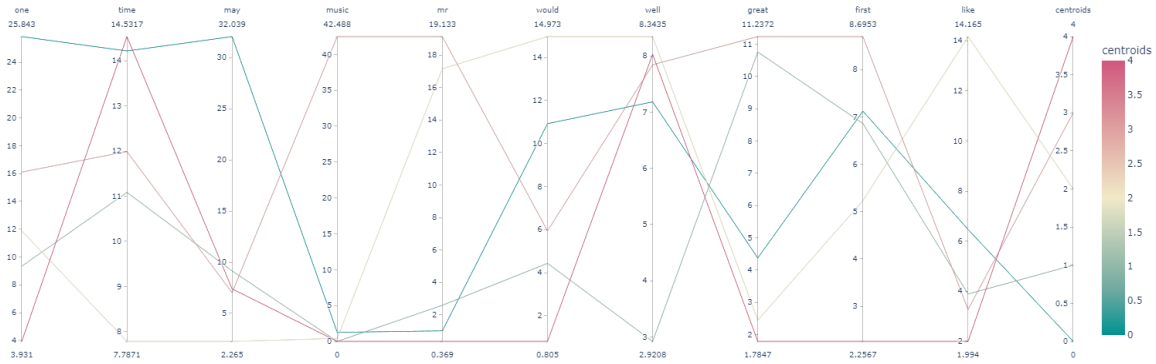
number of occurrences of each word in it. To account for varying size sentence, we normalize the vector by the number of words in the sentence. After that, we pass the transformed vector to 3 different classifiers, namely K-means, Hierarchical clustering, and EM.




```
Kappa Score = 0.8512500000000001
Silhouette Score = 0.04591679305083734
Homogeneity Score = 0.7632828440942648
```




```
<Figure size 720x432 with 0 Axes>
```



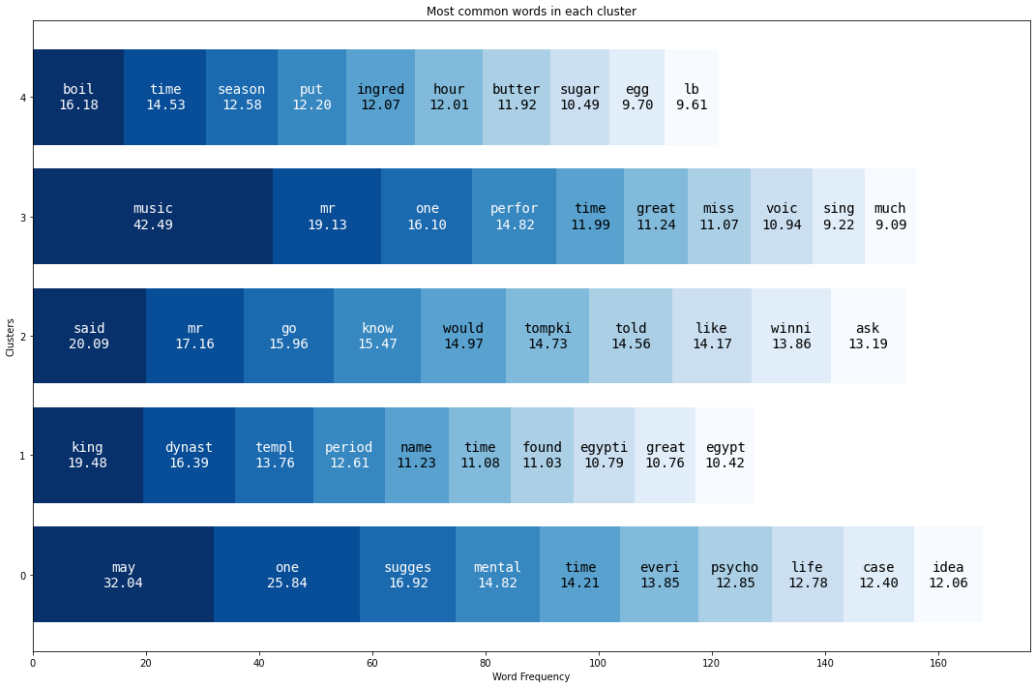
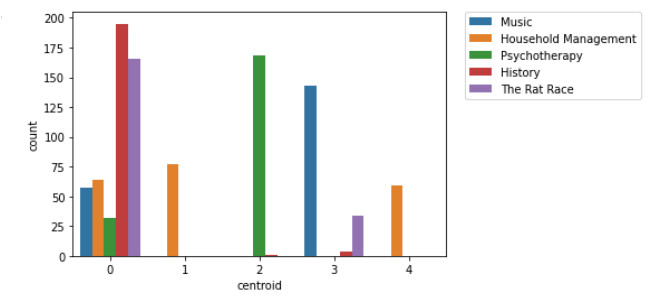
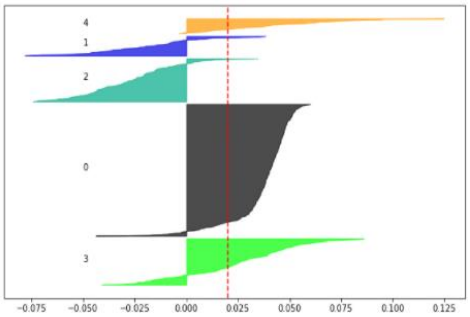
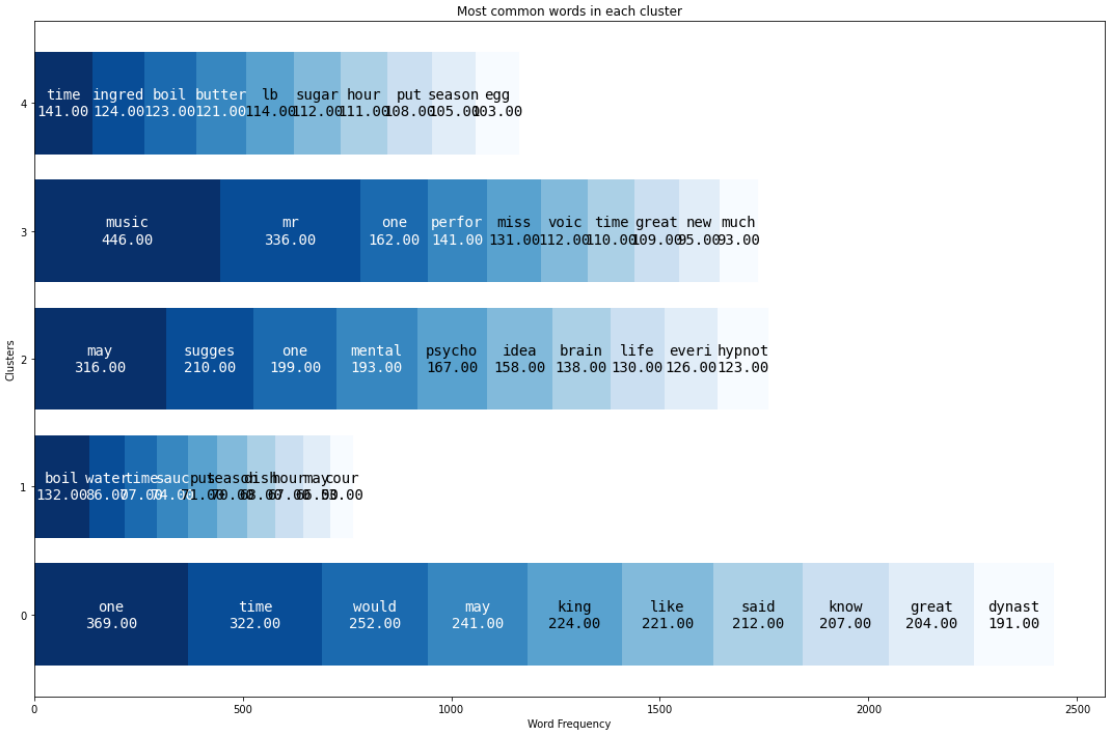


Figure 3: BOW with Hierarchical Clustering







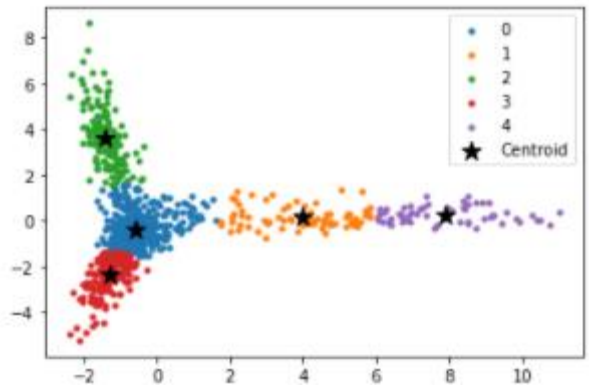


Figure 4: BOW with K-Means Clustering

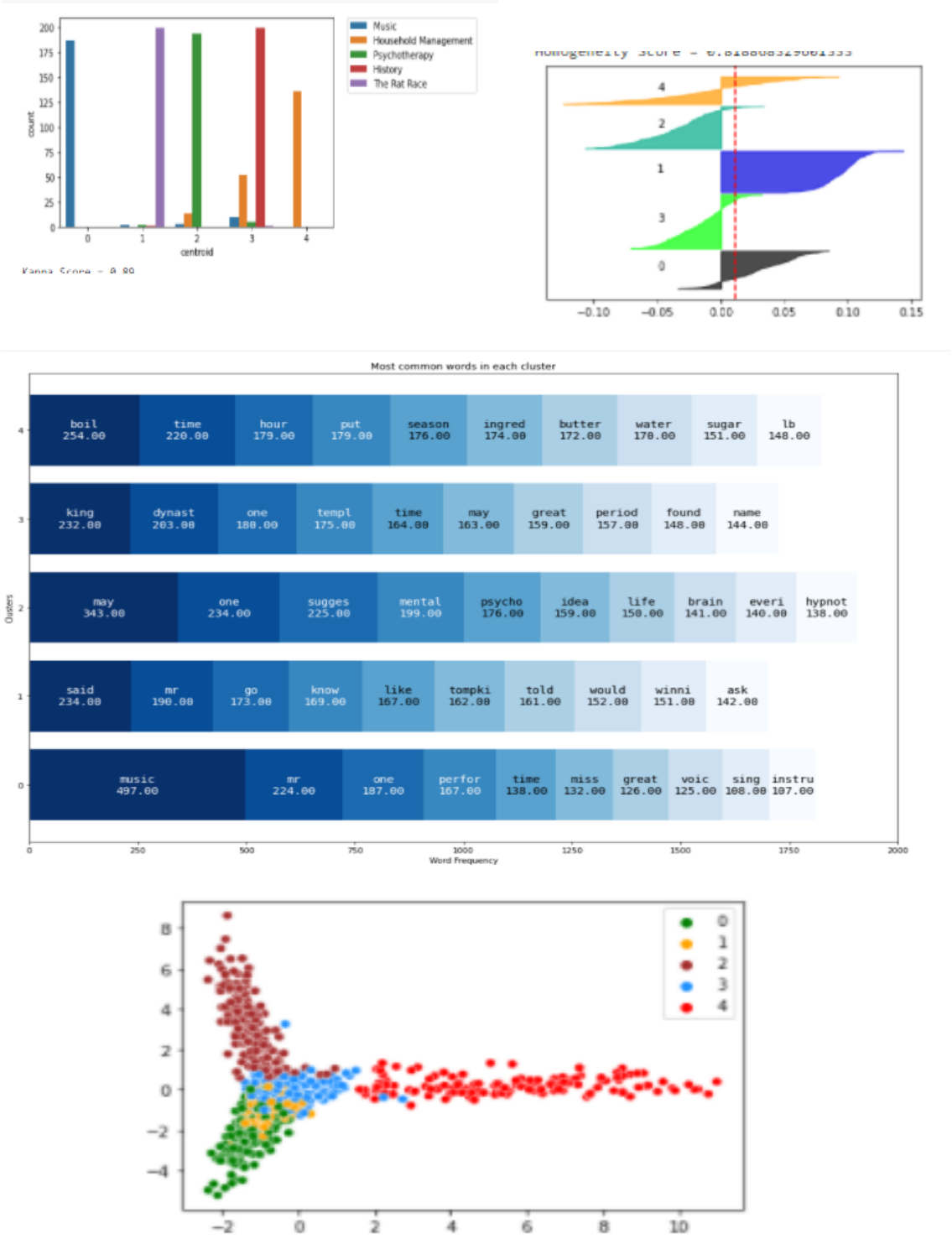


Figure 5: BOW with EM Clustering

b) Term Frequency-Inverse Document Frequency

A clear drawback of the simple Bag of Words transformation is that it does not consider very common words that are present in many documents. These frequent words are likely to dominate the vocabulary, carrying little information about the differences between documents. To address that, we use Term Frequency-Inverse Document Frequency transformation with K-means, Hierarchical clustering, and EM. which assigns a lower weight to the too frequent words.

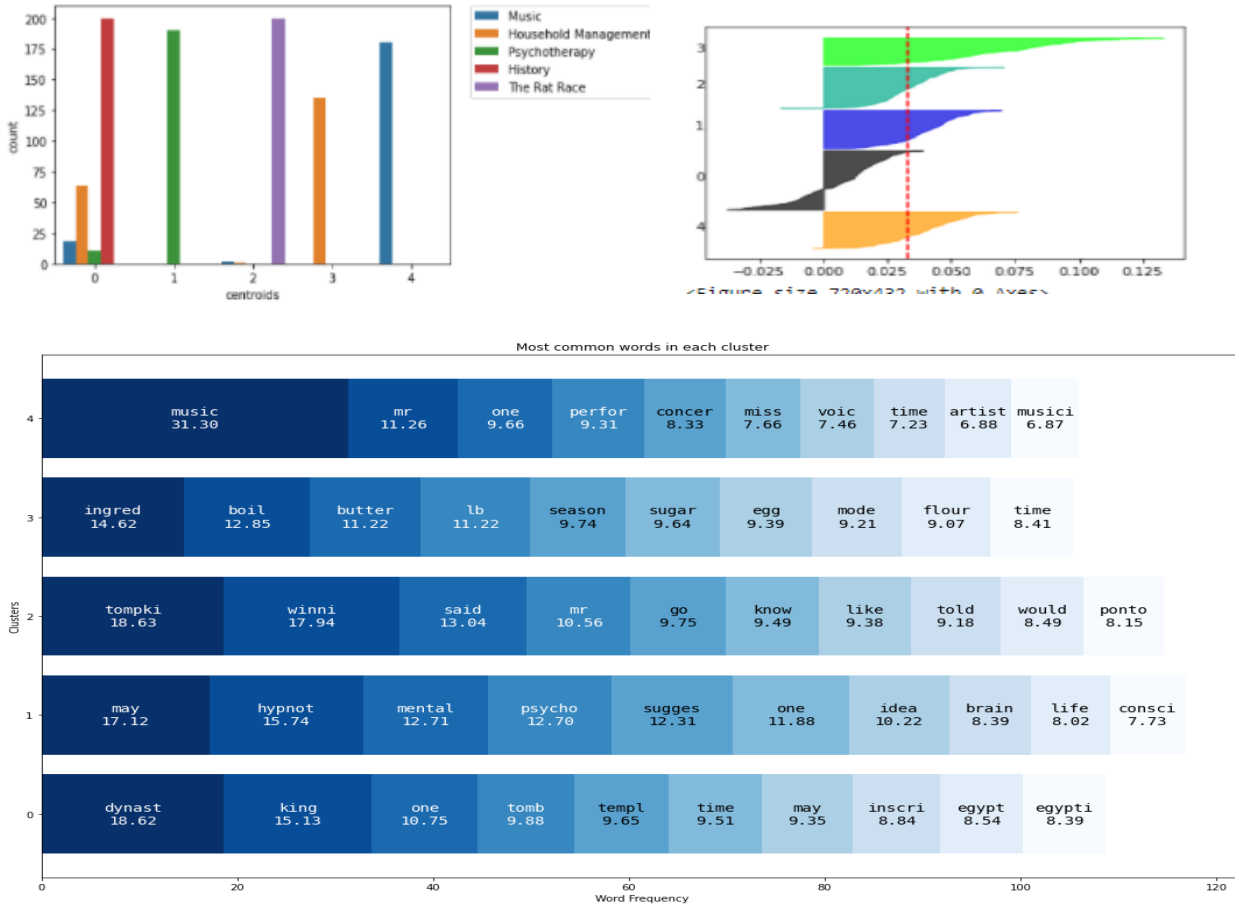


Figure 6: TF-IDF with hierarchical Clustering

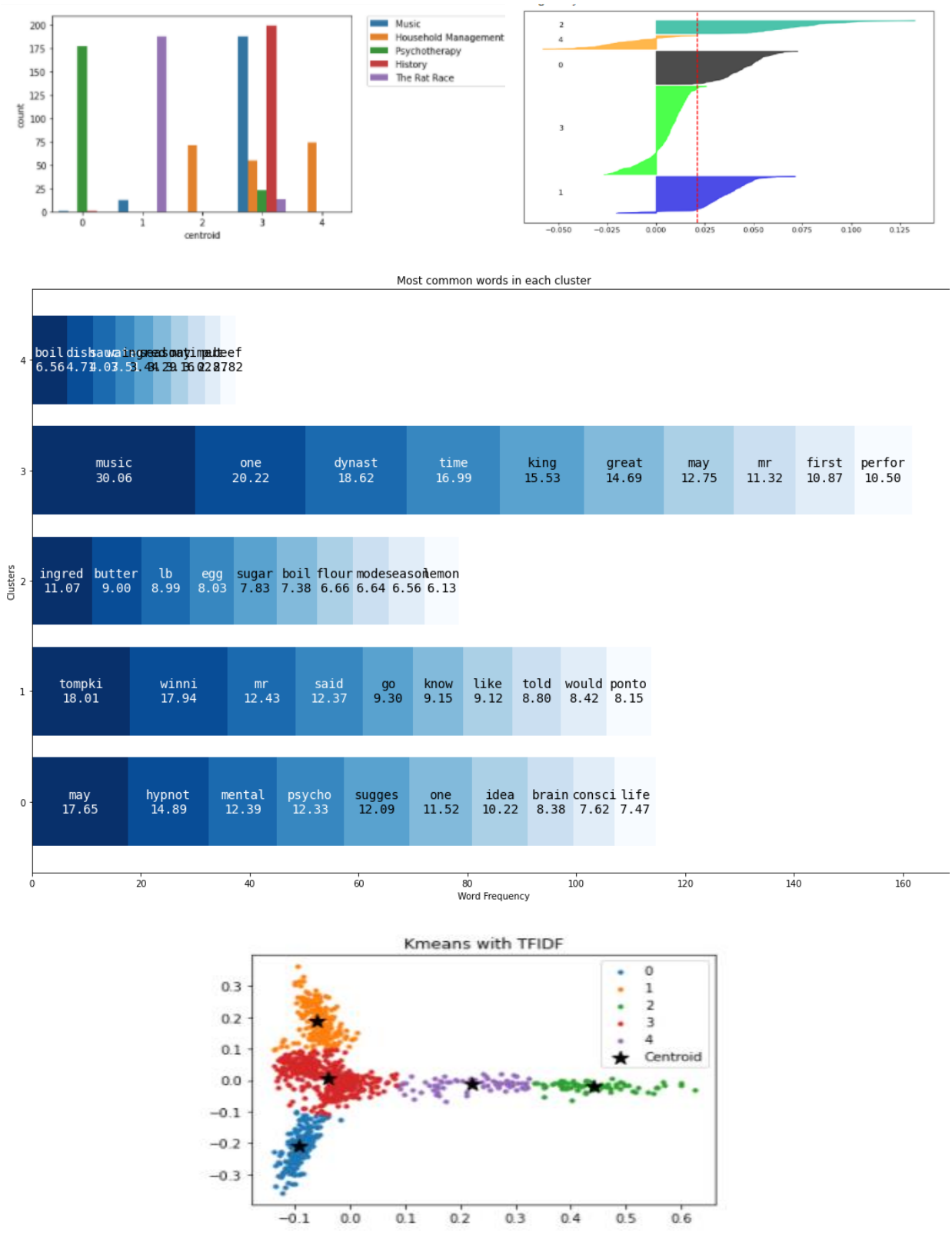
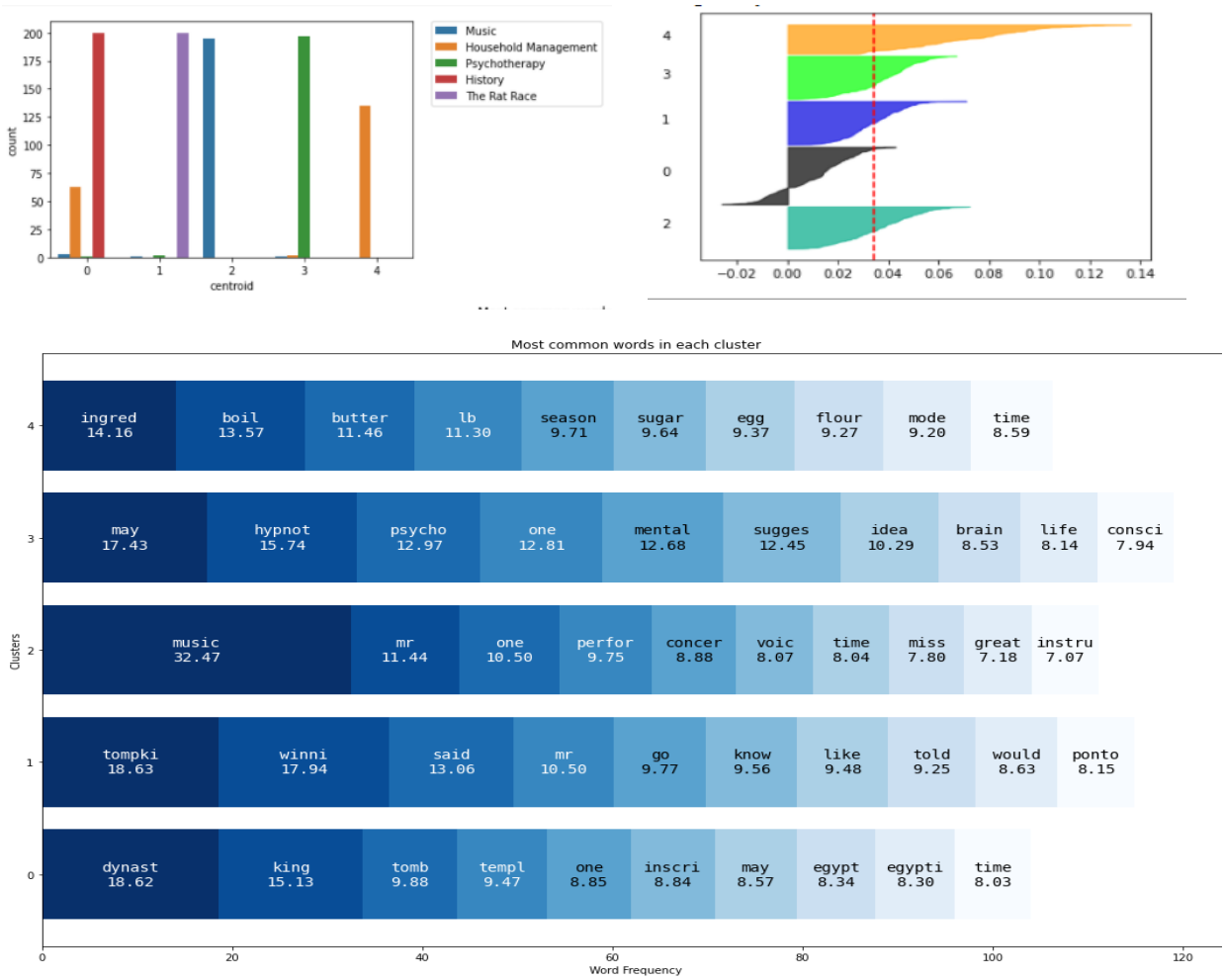


Figure 7: TF-IDF with K-means Clustering



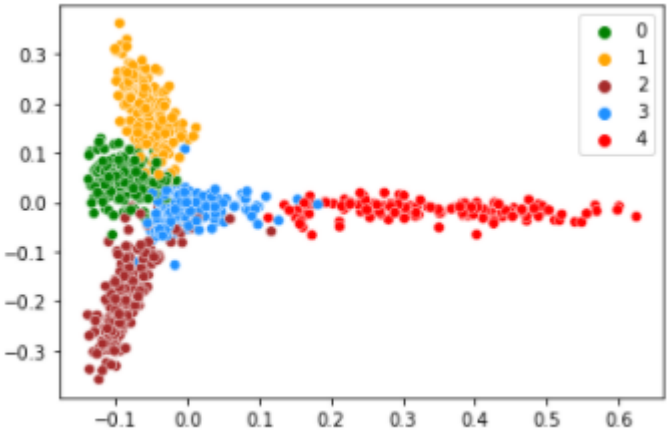


Figure 8: TF-IDF with EM Clustering

c) Doc2Vec

A more sophisticated transformation that we use here is Doc2Vec with K-means, Hierarchical clustering, and EM. It represents the whole partition of text as a fixed length vector and allows computing similarities.

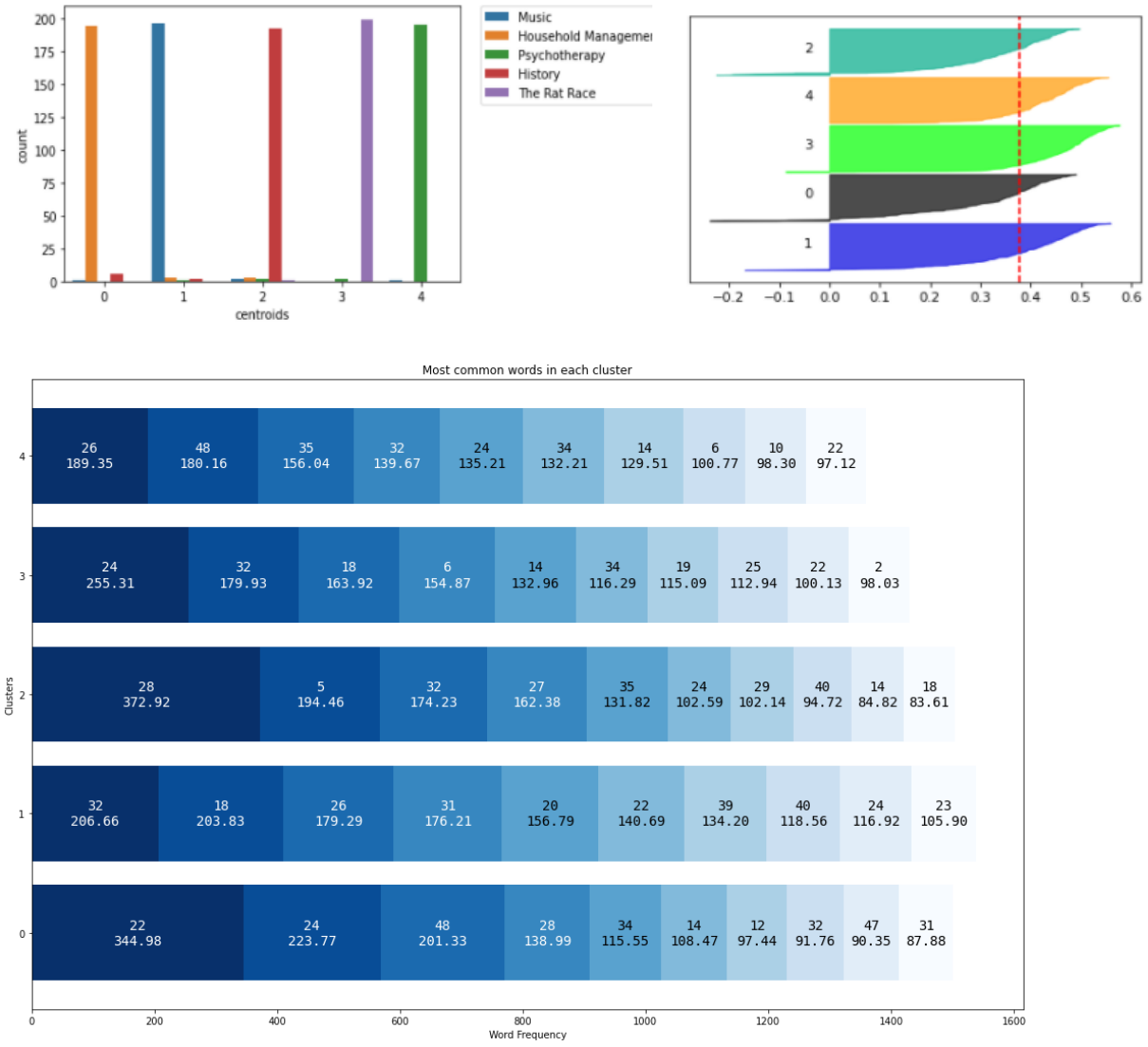


Figure 9: doc2vec with hierarchical Clustering

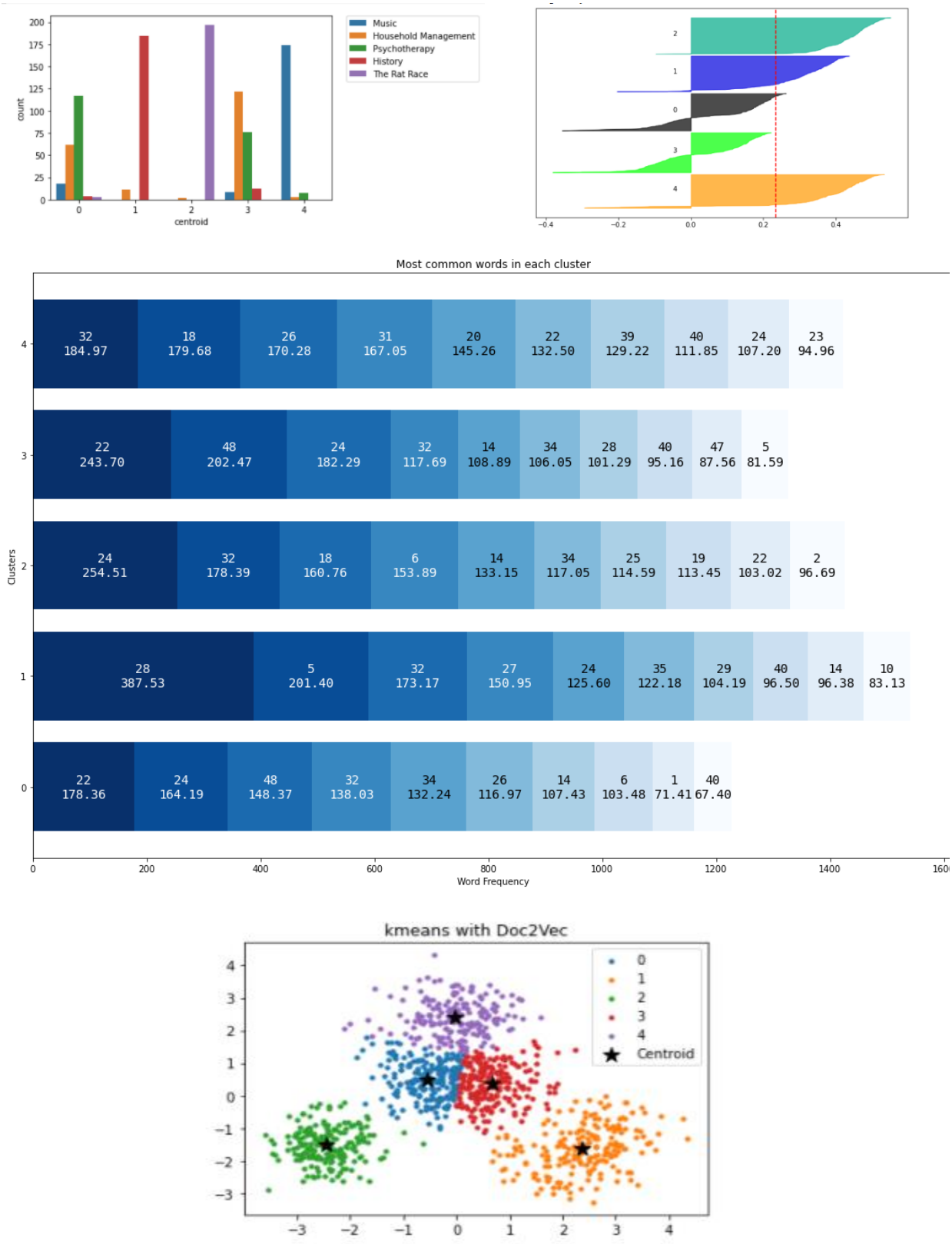


Figure 10: doc2vec with K-means Clustering

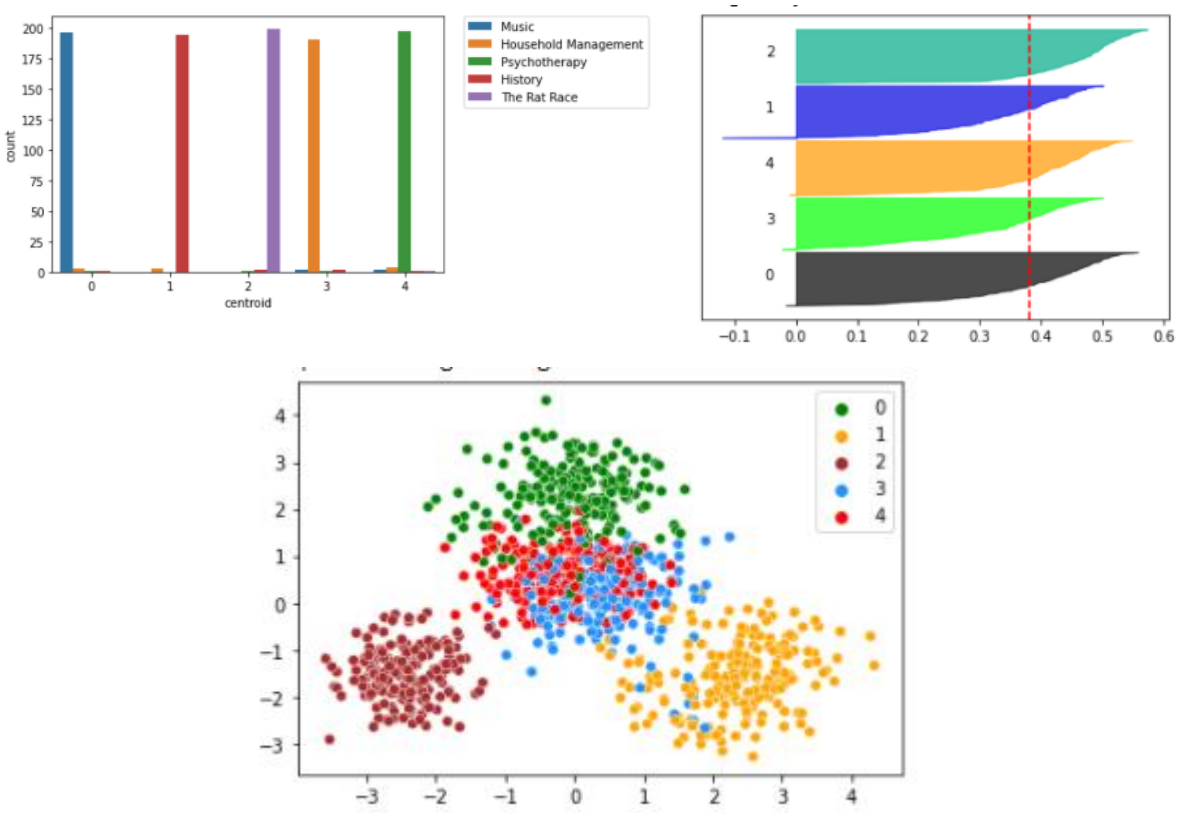
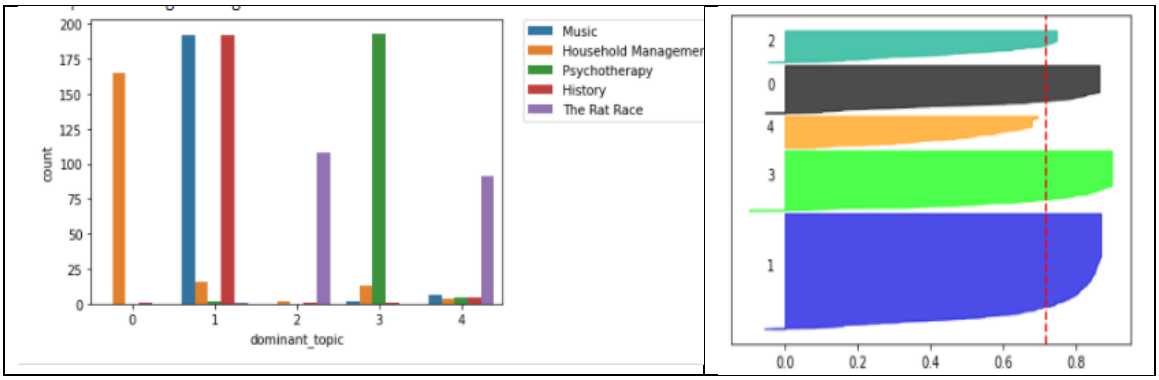


Figure 11: doc2vec with EM Clustering

d) Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a form of unsupervised learning that views documents as bags of words. define it as a generative probabilistic model of a corpus with the idea of representing each document as a random mixture over latent topics, each topic being characterize by a distribution over words.



3.5. Performance Evaluation

- **Cohen's Kappa:** Cohen's kappa is a metric often used to assess the agreement between two raters. we could use Cohen's kappa to compare the machine learning model predictions with the real data. It ranges from -1 to 1. Since we need to map cluster predictions to topics in order to be able to compute Kappa score, we construct a dictionary of cluster to book names, where we choose the dominant book in every cluster as the book prediction to any point mapped to this cluster.
- **Silhouette:** Silhouette score is a measure used to assess the quality of the clustering. Optimally, clustering would locate similar points in the same cluster (close to each other), and far from different points in other clusters. Silhouette score ranges from -1 to 1, where a higher score indicates better cluster, a score of 0 indicates that clusters are close to each other, and a negative score indicates an overlap between clusters.
- **Consistency:** Facts are said to be consistent if they support each other. In our setting, we define consistency to be that similar words are assigned to the same cluster. To assess this, we train a Word2Vec model, and compute the average pairwise similarity of the top 20 words of each cluster.

Algorithms	**K-means**		**EM**		**Hierarchical clustering**	
BOW	kappa	0.541	kappa	0.69	kappa	0.876
	silhouette	0.007	silhouette	0.011	silhouette	0.047
	Homogeneity	0.469	Homogeneity	0.697	Homogeneity	0.783
	Coherence	0.865	Coherence	0.893	Coherence	0.914
	Consistency within clusters	6677603.4	Consistency within clusters	6237014.6	Consistency within clusters	34758.78
	Consistency between clusters	2049.615	Consistency between clusters	2240.92	Consistency between clusters	16.44
TF-IDF	kappa	0.601	kappa	0.722	kappa	0.882
	silhouette	0.018	silhouette	0.026	silhouette	0.033
	Homogeneity	0.548	Homogeneity	0.750	Homogeneity	0.833
	Coherence	0.902	Coherence	0.894	Coherence	0.929
	Consistency within clusters	50478.23	Consistency within clusters	47401.4	Consistency within clusters	38762.80
	Consistency between clusters	9.40	Consistency between clusters	10.49	Consistency between clusters	12.12
Doc2Vec	kappa	0.807	kappa	0.97	kappa	0.961
	silhouette	0.262	silhouette	0.372	Silhouette	0.363
	Homogeneity	0.682	Homogeneity	0.918	Homogeneity	0.905
	Coherence	0.879	Coherence	0.898	Coherence	0.909
	Consistency within clusters	253328.2	Consistency within clusters	198589.48	Consistency within clusters	204578.11
	Consistency between clusters	1278.5	Consistency between clusters	1528.81	Consistency between clusters	204578.11

Algorithms	**Kappa**	**Silhouette**	**Homogeneity**	**Coherence**	**Consistency within clusters**	**Consistency between clusters**

LDA	0.94	0.779	0.868	0.921	1807.38	112.32

3.6. Champion Model

After training all these models and trying different transformations, it is time to choose the best performing model. We choose Kappa score to be our primary measure, since we already have the true labels, and high Kappa score implies that our clustering is agreeing with the true labels. The top performing model-transformation combination according to this criterion was Doc2Vec transformation, with EM clustering. It achieved a Kappa score of 0.97.

3.7. Error Analysis

Although our champion model performed really well on most examples, it still missed a lot. In order to understand the reasons that lie behind misclassifications, and identify areas of possible improvements, we perform some error analysis. First, lets see from which books our misclassifications come, and to which cluster where they predicted. This is shown in the figure below.

	book_name	cluster_book	count
0	History	Household Management	4
1	History	Psychotherapy	1
2	History	The Rat Race	3
3	Household Management	History	2
4	Household Management	Music	3
5	Household Management	Psychotherapy	3
6	Music	History	1
7	Music	Household Management	2
8	Music	Psychotherapy	2
9	Psychotherapy	Household Management	1
10	Psychotherapy	The Rat Race	1
11	The Rat Race	Psychotherapy	1

Figure 12: Misclassified Examples

As shown above, we have 24 misclassified examples. The majority (8) of these misclassified examples come from Household Management book, followed by 8 from history book. Now, we dig deeper into these wrong examples. First, extract the top 20 frequent words of each book.

Next, we compare the how many of these keywords, were present in the misclassified examples of the Household management book. The figure below shows the composition of these keywords.

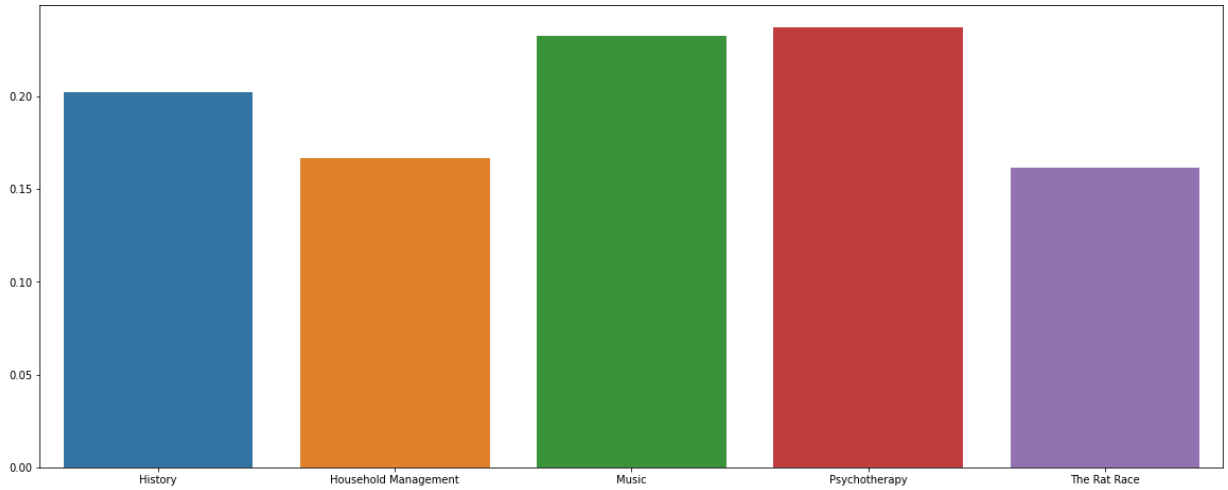


Figure 13: Composition of Keyword Frequencies in Misclassified Examples from Household Management Book

Although these examples belong to Household Management, they contain more keywords from all other classes. To get a better understanding of what this should look like, we plot the same figure again, but for the correctly classified examples from Household Management.

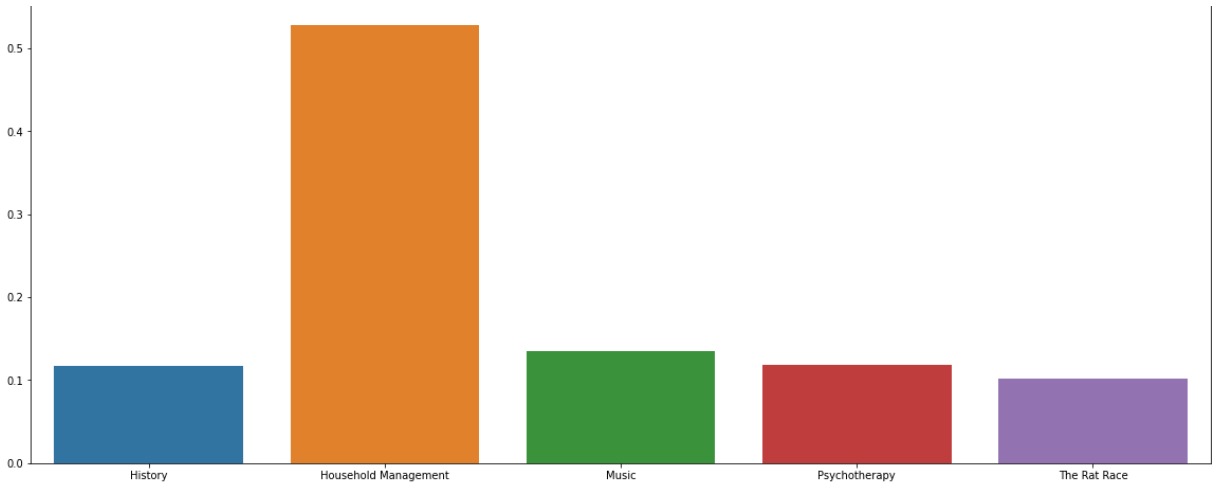


Figure 14: Composition of Keyword Frequencies in Misclassified Examples from Household Management Book

Here, we can see that keywords from Household Management book are significantly more present than keywords from any other category. However, keywords from other categories seem to be equally present in the correct predictions. This may imply that there are common words between all books, and there are specific words unique to each book. The absence of these words causes the misclassifications. To test this hypothesis, let's visualize the frequency of the keywords, extracted from Household Management book, in the correct and wrong predictions.

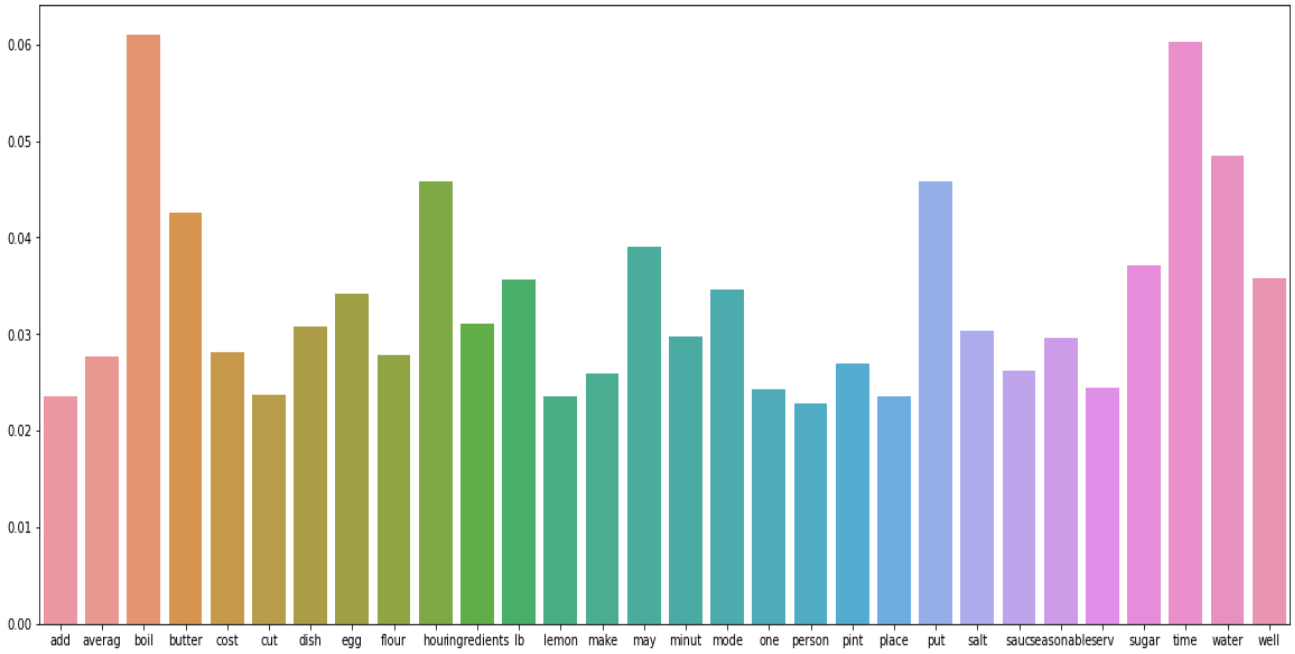


Figure 15: Keyword Frequencies in Household Management Correct Predictions

In the figure above, we can see how frequent the top 20 words are in the correctly classified examples. We can observe some keywords unique to household management topic, like butter, egg, ingredients, sugar, etc. It is very unlikely to observe these keywords, say in a music book. After that, we plot the same figure again, but for the wrongly classified examples.

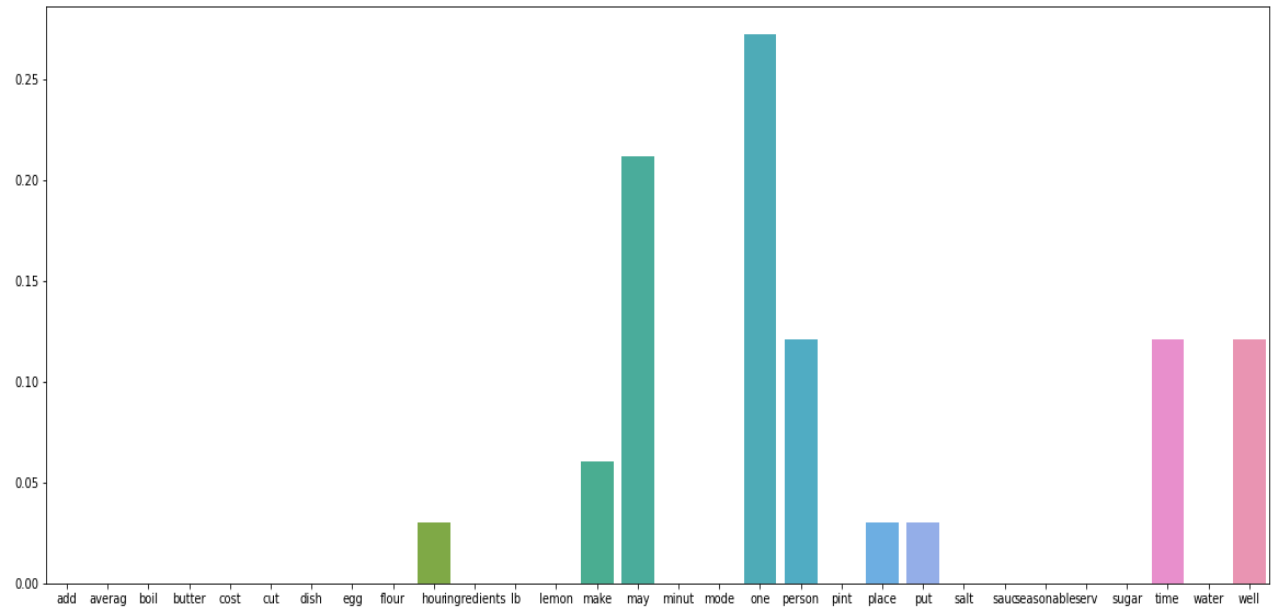


Figure 16: Keyword Frequencies in Household Management Correct Predictions

As shown in the figure above, the keywords we mentioned earlier are not present. This led to the confusion of the model and had it misclassify examples from Household Management to other classes.

To better understand the words common between different books, we construct a knowledge graph of the shared words, and visualize it below.

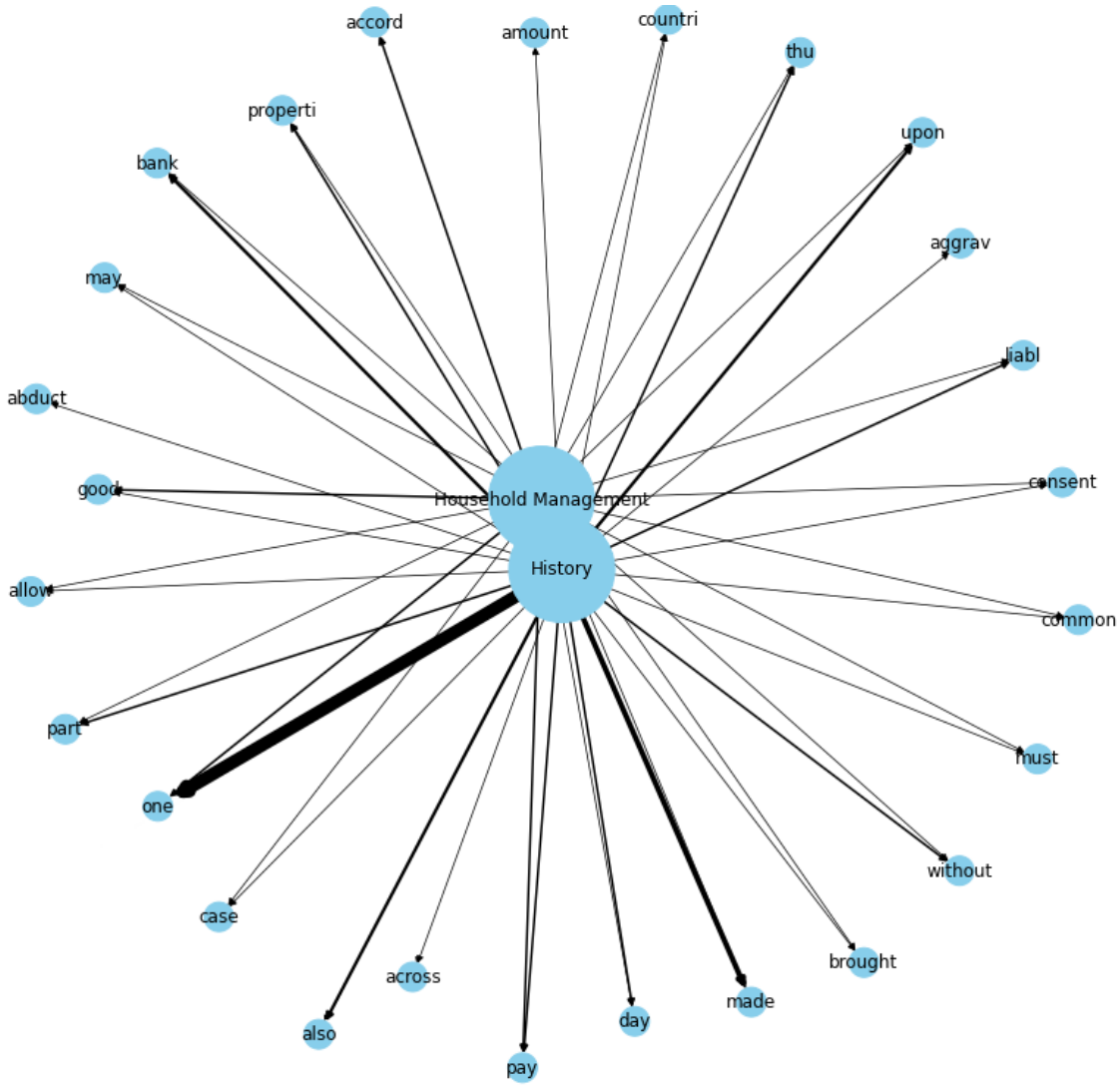


Figure 17: Knowledge Graph of Shared Words

The thickness of each lines shows how frequent a word is in a certain book. We can observe that the word one is very frequent for example. Most of the other words are regular words common in the language that bear no distinctive features for classifying the books.