Assignment 2: Assess Clustering and Classification Outputs

Kathryn LiPetri

MSDS 453: DL Section 56, Fall 2022

October 23, 2022

Introduction and Problem Statement

The goal of this assignment is to begin to determine terms that will be part of the corpus-wide vocabulary comprised of the class-corpus of movie reviews, using clustering and classification. This assignment is building off work that was performed in Assignment 1 with data wrangling and vectorization. Each member of the class was assigned a movie from one of four movie genres (action, comedy, horror, and sci-fi) to gather ten reviews for (five positive and five negative). These two-hundred documents comprise the class-corpus of movie reviews.

For this assignment, multiple experiments will be performed using clustering, sentiment analysis, multi-class classification, and topic modeling. The results of the experiments will be compared with the goal being to group similar documents together based on movie genre and movie review sentiment (positive or negative).

Data

The dataset of movie reviews contains two-hundred documents total consisting of ten movie reviews (five positive and five negative) for twenty different movies in four different movie genres (action, comedy, horror, and sci-fi). For the twenty movies, six are action, five are comedy, four are horror, and five are sci-fi. Each document is defined as a single movie review and contains at least five-hundred words. Some preliminary work was performed to normalize the documents such as, removing punctuation, putting everything in lower case, removing tags, and removing special characters and digits. The following is a data dictionary that describes the nine columns of the class-corpus data set.

Table 1. Class-Corpus Data Dictionary

Column Name	Definition
DSI_Title	Student initials, document number, and movie title assigned to the movie review; identical to the Submission File Name
Doc_ID	Unique number assigned to each document row (0 – 199)

Column Name	Definition
Text	The actual text of the movie review
Submission File Name	Student initials, document number, and movie title assigned to the movie review; identical to the DSI_Title
Student Name	Initials associated with each individual student
Genre of Movie	One of four movie genres assigned to the movie: Action, Comedy, Horror, or Sci-Fi
Review Type (pos or neg)	Describes whether the movie review was positive or negative
Movie Title	The title of the movie
Descriptor	Movie genre, movie title, N or P for negative or positive, and Doc_ID

Research Design and Modeling Methods

Experiments were performed using clustering, sentiment analysis, multi-class classification, and topic modeling methods. Table 2 summarizes the experiments that were performed.

Table 2. Assignment 2 Experiments Summary

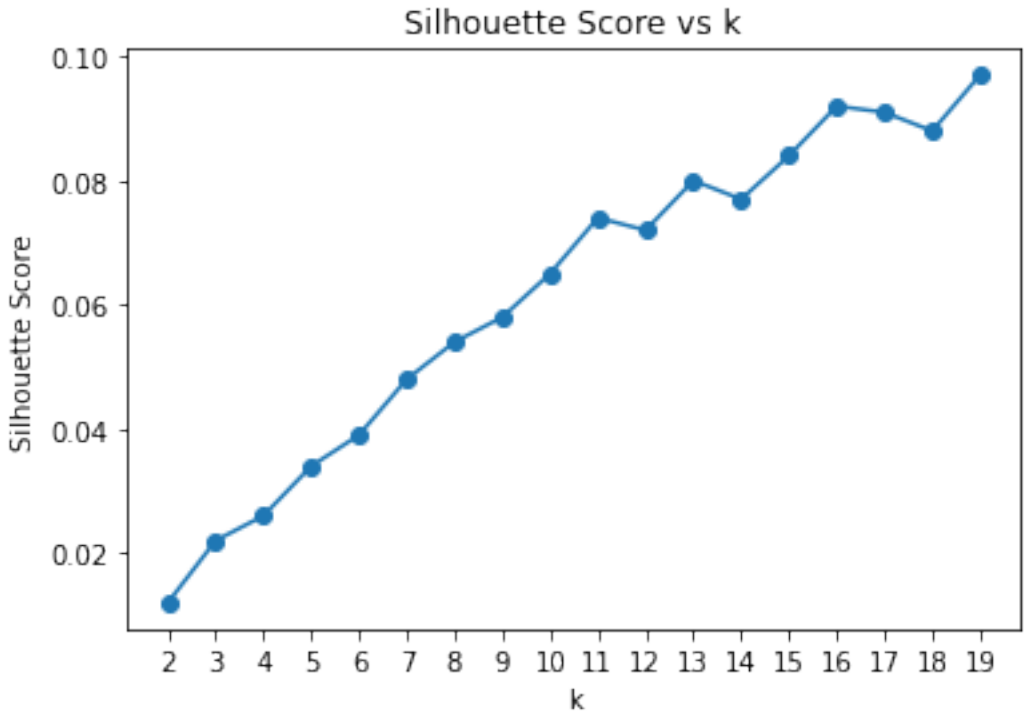
Experiment Type	Experiment Description
Clustering	• K-means clustering produced by tf-idf algorithm. • K values of 2 – 19 were used, and the silhouette score was calculated for each k value. • TF-IDF was used to determine top 10 terms in each cluster.
Sentiment Analysis	• A total of 8 experiments were performed with SVM, Logistic Regression, Naïve Bayes, and Random Forest using TF-IDF and Doc2Vec with embedding level 200.
Multi-Class Classification	• A total of 8 experiments were performed with SVM, Logistic Regression, Naïve Bayes, and Random Forest using TF-IDF and Doc2Vec with embedding level 200.
Topic Modeling	• Four experiments were performed with LSA: 2, 4, 6, 20 topics with 10 words. • Four experiments were performed with LDA: 2, 4, 6, 20 topics with 10 words.

Results

Clustering

Clustering was performed using k-means clustering to group similar movie review documents together by genre. Clusters were produced using the tf-idf algorithm, and the algorithm was performed 18 times, varying k-values from 2 to 19. Silhouette scores were calculated for each k-value, and are shown in Figure 1, below.

Figure 1. Silhouette Score vs. K-value



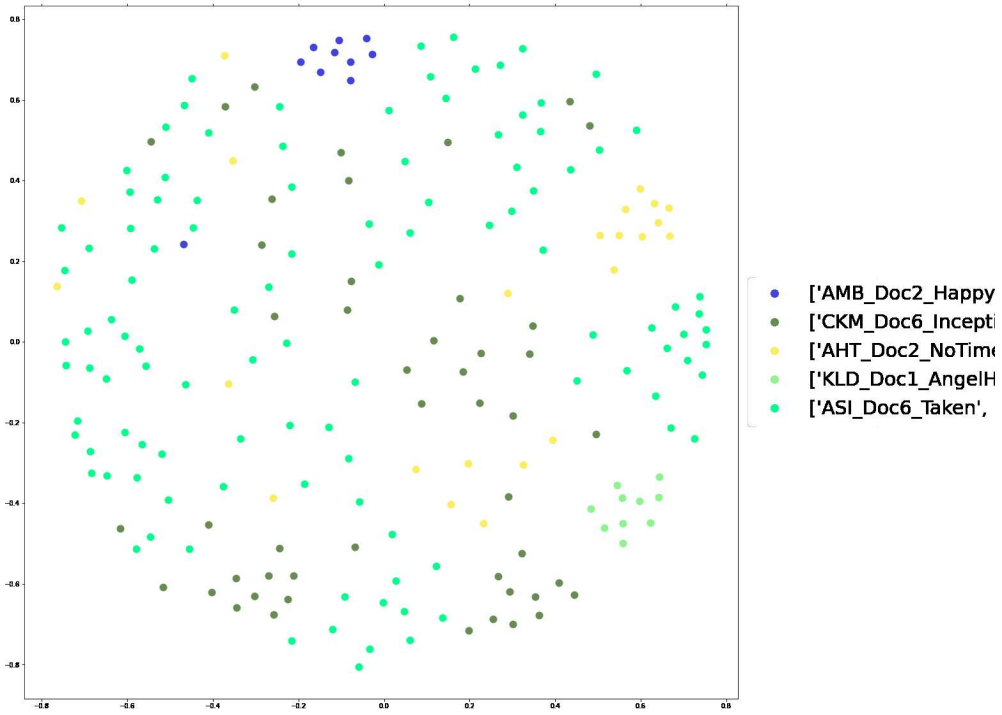
Based on the scores presented in Figure 1, clusters for k-value = 11 were looked at first because after 11, the silhouette score fluctuated instead of steadily increasing. The clusters for k-value = 11 were studied, but 9 of the 11 clusters were found to be from a single movie (See Appendices for more details). This is not very useful for grouping movies by genre. Therefore, a different set of clusters were used for further analysis of k-value = 5, and clusters for this k-value are shown in Table 3.

Table 3. Movie Review Clusters for K-value = 5

Cluster	**Movie Reviews** **(orange = comedy, green = action, blue = horror, purple = sci-fi)**
Cluster 0	Happy Gilmore (Doc 1- 10)
Cluster 1	Inception (Doc 1- 10) Taken (Doc 3, 5, 9, 10) The Revenant (Doc 1 - 10) Despicable Me 3 (Doc 5) Equilibrium (Doc 2, 3, 7, 9) Minority Report (All but 7) Oblivion (All but 5 and 10) Pitch Black (Doc 8, 9)
Cluster 2	No Time to Die (Doc 1 – 10) Despicable Me 3 (Doc 1, 9, 10) Batman (Doc 1 – 10)
Cluster 3	Angel Has Fallen (Doc 1 – 10)
Cluster 4	Taken (Doc 1, 2, 4, 6, 7, 8) The Toxic Avenger (Doc 1 – 10) Despicable Me 3 (Doc 2, 3, 4, 6, 7, 8) Dirty Grandpa (Doc 1 – 10) Legally Blonde (Doc 1- 10) The Lost City (Doc 1 - 10) Drag Me to Hell (Doc 1 - 10) Fresh (Doc 1- 10) It Chapter Two (Doc 1- 10) Us (Doc 1 – 10) Equilibrium (Doc 1, 4, 5, 6, 8, 10) Minority Report (Doc 7) Oblivion (Doc 5, 10) Pitch Black (Doc 1, 2, 3, 4, 5, 6, 7, 10)

A plot was created to visualize the clusters in a two-dimensional plane and is shown in Figure 2.

Figure 2. Cluster Plot in Two-Dimensional Plane, K-value = 5



Sentiment Analysis

Sentiment analysis was performed to determine whether movie reviews gave a positive or negative rating. For this analysis different vectorization techniques and classification models were used to determine which would most accurately model the true sentiment of the movie reviews. The two vectorization techniques used were TF-IDF and Doc2Vec with an embedding dimension of 200. For each vectorization method, 4 different classification models were used, SVM, Logistic Regression, Naïve Bayes, and Random Forest, for a total of 8 experiments. Note that for the experiments using Naïve Bayes, Multinomial Naïve Bayes was used with TF-IDF while Gaussian Naïve Bayes was used with Doc2Vec. The training and testing data was split 67%/33%, and a classification report, including metrics for precision, recall, f1-score, and

accuracy, and a confusion matrix were generated for each experiment. Table 4 shows a summary of the experiments and their respective classification report metrics.

Table 4. Sentiment Analysis Experiments

Experiment	Vectorization Method	Classification Model	Label	Precision	Recall	F1-Score	Accuracy
1	TF-IDF	SVM	Negative	0.48	0.67	0.56	0.47
			Positive	0.45	0.27	0.34	
2	Doc2Vec		Negative	0.50	0.39	0.44	0.50
			Positive	0.50	0.61	0.55	
3	TF-IDF	Logistic Regression	Negative	0.47	0.52	0.49	0.47
			Positive	0.47	0.42	0.44	
4	Doc2Vec		Negative	0.46	0.33	0.39	0.47
			Positive	0.48	0.61	0.53	
5	TF-IDF	Naïve Bayes	Negative	0.36	0.27	0.31	0.39
			Positive	0.41	0.52	0.46	
6	Doc2Vec		Negative	0.56	0.45	0.50	0.55
			Positive	0.54	0.64	0.58	
7	TF-IDF	Random Forest	Negative	0.50	0.30	0.38	0.50
			Positive	0.50	0.70	0.58	
8	Doc2Vec		Negative	0.47	0.42	0.44	0.47
			Positive	0.47	0.52	0.49	

*Highlighted cells have the highest metric for review type.

Multi-Class Classification

Multi-class classification was performed to determine what genre a movie belongs to based on the movie review. For this analysis different vectorization techniques and classification models were used to determine which would most accurately model the genre of the movies. The two vectorization techniques used were TF-IDF and Doc2Vec with an embedding dimension of 200. For each vectorization method, 4 different classification models were used, SVM, Logistic Regression, Naïve Bayes, and Random Forest, for a total of 8 experiments. The training and testing data was split 67%/33%, and a classification report, including metrics for precision, recall, f1-score, and accuracy, and a confusion matrix were generated for each experiment. Table 5 shows a summary of the experiments and their respective classification report metrics.

Table 5. Multi-Class Classification Experiments

Experiment	Vectorization Method	Classification Model	Label	Precision	Recall	F1-Score	Accuracy
1	TF-IDF	SVM	Action	0.80	1.00	0.89	0.92
			Comedy	1.00	0.82	0.90	
			Horror	1.00	0.85	0.92	
			Sci-Fi	1.00	1.00	1.00	
2	Doc2Vec		Action	0.47	0.70	0.56	0.53
			Comedy	0.57	0.71	0.63	
			Horror	0.50	0.08	0.13	
			Sci-Fi	0.62	0.50	0.55	
3	TF-IDF	Logistic Regression	Action	0.80	1.00	0.89	0.92
			Comedy	1.00	0.82	0.90	
			Horror	1.00	0.85	0.92	
			Sci-Fi	1.00	1.00	1.00	
4	Doc2Vec		Action	0.48	0.65	0.55	0.48
			Comedy	0.53	0.59	0.56	
			Horror	0.50	0.08	0.13	
			Sci-Fi	0.44	0.50	0.47	
5	TF-IDF	Naïve Bayes	Action	0.77	1.00	0.87	0.91
			Comedy	1.00	0.82	0.90	
			Horror	1.00	0.77	0.87	
			Sci-Fi	1.00	1.00	1.00	
6	Doc2Vec		Action	0.59	0.50	0.54	0.44
			Comedy	0.48	0.59	0.53	
			Horror	0.08	0.08	0.08	
			Sci-Fi	0.50	0.50	0.50	
7	TF-IDF	Random Forest	Action	1.00	0.90	0.95	0.94
			Comedy	0.81	1.00	0.89	
			Horror	1.00	0.92	0.96	
			Sci-Fi	1.00	0.94	0.97	
8	Doc2Vec		Action	0.36	0.45	0.40	0.39
			Comedy	0.53	0.53	0.53	
			Horror	0.14	0.08	0.10	
			Sci-Fi	0.41	0.44	0.42	

*Highlighted cells have the highest metric for movie genre.

Topic Modeling

Topic modeling was performed using LSA and LDA models to determine topics present in the class corpus movie reviews. A total of 8 different experiments were performed using LSA and LDA methods with different numbers of topics. Table 6 summarizes the details of the

experiments that were performed. LDA was performed with increased iterations and passes in an attempt to improve the results.

Table 6. Topic Modeling Experiments with LSA/LDA

Experiment	Method	Number of Topics	Number of Words	Iterations	Passes	Coherence
1	LSA	2	10	N/A	N/A	0.48
2	LDA	2	10	1000	200	0.26
3	LSA	4	10	N/A	N/A	0.38
4	LDA	4	10	1000	200	0.26
5	LSA	6	10	N/A	N/A	0.51
6	LDA	6	10	1000	200	0.25
7	LSA	20	10	N/A	N/A	0.39
8	LDA	20	10	1000	200	0.28

A cosine similarity matrix was output for each experiment, and the coherence was calculated for the experiments performed with LSA and LDA separately. All cosine similarity matrices are available in the Appendices. The coherence value for each experiment is also presented in Table 6. Each experiment output 10 words associated with each topic. For LSA, the experiment with the highest coherence value was 6 topics and 10 words per topic. For LDA, the highest coherence value was 20 topics and 10 words per topic. Figure 3 below shows the topics and words output for the LSA experiment with 6 topics and 10 words per topic.

Figure 3. LSA, 6 Topics and 10 Words

```
[(0, '0.377*"movie" + 0.283*"batman" + 0.136*"character" + 0.131*"anderton" +
0.130*"would" + 0.129*"story" + 0.124*"characters" + 0.116*"first" + 0.116*"action" +
0.110*"spielberg"')
(1, '-0.454*"batman" + 0.411*"anderton" + 0.279*"spielberg" + 0.257*"minority" +
0.256*"report" + 0.173*"precrime" + 0.146*"future" + 0.135*"technology" + -
0.127*"reeves" + -0.101*"movie"'),
(2, '-0.546*"batman" + 0.263*"movie" + 0.241*"toxic" + -0.229*"anderton" + -
0.154*"reeves" + 0.144*"avenger" + -0.137*"spielberg" + -0.126*"report" + -
0.125*"minority" + 0.123*"action"'),
(3, '-0.513*"toxic" + -0.325*"avenger" + -0.248*"troma" + -0.242*"melvin" +
0.198*"action" + 0.152*"banning" + -0.140*"toxie" + -0.102*"batman" + 0.099*"fallen" +
-0.097*"anderton"'),
(4, '-0.264*"movie" + 0.228*"banning" + -0.213*"equilibrium" + 0.171*"inception" +
0.151*"nolan" + 0.146*"fallen" + -0.139*"future" + -0.123*"wimmer" + -0.108*"action" +
0.102*"angel"'),
(5, '0.456*"banning" + 0.297*"fallen" + -0.271*"inception" + -0.248*"nolan" +
0.207*"angel" + 0.179*"action" + 0.170*"president" + 0.149*"butler" + 0.130*"service"
+ 0.120*"agent"')]
```

Analysis and Interpretation

Clustering

The goal of using k-means clustering experiments was to group the movie review documents logically by genre. Looking at the clusters formed by using a k-value of 5, there are two movies that have their own clusters (Cluster 0 and Cluster 3): “Happy Gilmore” (Comedy) and “Angel Has Fallen” (Action). Per Figure 2, the documents in these respective clusters are plotted close together. Cluster 1, comprised of mostly action and sci-fi movie review documents, has a few different localized groups of dots in Figure 2. Cluster 2, comprised on action, sci-fi, and comedy movie reviews has one tightly plotted group of dots and many dots scattered throughout the plot in Figure 2. Cluster 4 is the largest cluster with many groups of dots all over the plot and contains movie reviews from all four genres in the corpus.

Most of the clusters either contain a single movie or a mix of movies from different genres. Each cluster was analyzed with TF-IDF to determine the 10 words with the highest TF-IDF score. These lists of words for each cluster can be found in the Appendices. Cluster 0 only contains reviews from the movie “Happy Gilmore”, and therefore most of the words are related specifically to that movie, such as actor and character names and themes throughout the movie, like hockey. The only more general term in the list is the word comedy. Since most of the terms are very specific to the movie, it makes sense why “Happy Gilmore” was in its own cluster.

In Cluster 1, it does make some sense that action movie reviews are grouped together with sci-fi movie reviews since many sci-fi films tend to have action components as well. The movie review document 5 from “Despicable Me” was included in this cluster, and it is not exactly apparent why. This movie review is very short, and perhaps some of the words in the

review fit better with the other reviews in this cluster. Once again, the top 10 words are all very specific to certain movies within the cluster.

Cluster 2 is another cluster mixed with one action movie, one sci-fi movie, and some reviews from “Despicable Me 3”. The remaining movie reviews for “Despicable Me 3” are scattered in Clusters 2 and 4. Looking back at the terms for each of the movie reviews, documents 1, 9, and 10 all contain mention of the title of the movie while documents 2, 3, 4, 6, 7, 8 do not, so this might be a reason why those reviews are not clustered together. The top 10 TF-IDF word scores are for words very specific to the movies “Batman” and “No Time to Die” which are included in the cluster. The only general word in this list is the word character.

Cluster 3 contains only movie reviews for the movie “Angel Has Fallen”, and therefore all the words with the top 10 TF-IDF scores are specifically related to characters, actors, and themes in the movie. Cluster 4 contains movies from all genres and is the largest cluster that formed. Looking at the TF-IDF list, there are some more general terms on the list like movie, horror, comedy, and character. The remaining terms are all related to specific movies in the cluster.

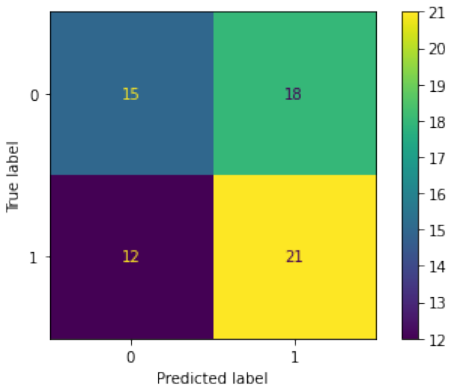
Overall, the k-means clustering is not doing a good job to cluster movies in the same genre together. More clusters just ended up putting more movies into a cluster by themselves, and fewer clusters were not able to group movies of the same genre together. So, it seems to work better for clustering reviews from the same movie. Clustering by genre might improve if specific terms, such as movie characters, actors, etc. were removed from the reviews so that themes related to the genre of the movie remain.

Sentiment Analysis: Positive and Negative Reviews

Based on the data presented in Table 4, for the sentiment analysis experiments the model with the highest accuracy overall is the Naïve Bayes model using Doc2Vec with an embedding dimension of 200. The model with the lowest accuracy over all is the Naïve Bayes model using TF-IDF as the vectorization method. It is interesting that for this classification model, the vectorization method made a large difference in the outcome. The Naïve Bayes model with Doc2Vec also had the highest precision metrics for both negative and positive reviews and the highest f1-score for positive reviews. The SVM model with TF-IDF vectorization had the highest recall and f1-score for negative reviews. The Random Forest model with TF-IDF vectorization had the highest recall and f1-score (tied with Naïve Bayes/Doc2Vec) for positive reviews. TF-IDF vectorization yielded higher metrics overall when looking at the data holistically but did not work well with the Naïve Bayes model. The SVM model with TF-IDF was the best at predicting negative reviews correctly, and the Random Forest model with TF-IDF was the best at predicting positive reviews correctly.

Figure 3 shows the confusion matrix for the most accurate model overall, Naïve Bayes model with Doc2Vec embedding level 200. All confusion matrices for sentiment analysis can be found in the Appendices.

Figure 4. Confusion Matrix Naïve Bayes/Doc2Vec, 200



From the figure, the model predicted 15 of 33 negative reviews correctly and 21 of 31 positive reviews correctly. The model misclassified 18 negative reviews as positive and 12 positive reviews as negative. This model was better at detecting and classifying positive reviews correctly than negative reviews.

Overall, none of the classification models did a fantastic job at predicting the sentiment of the movie reviews, considering that the highest overall accuracy was 0.55. To improve upon this, perhaps different vectorization methods would yield better results. In addition, having a larger dataset of movie reviews would most likely improve the model since there would be a larger amount of data to train the model with. Another factor could be that movie reviews are not showing sentiment strongly enough. Many movie reviewers take a middle of the road stance, discussing the good and bad about the movie.

Multi-Class Classification: Movie Genre

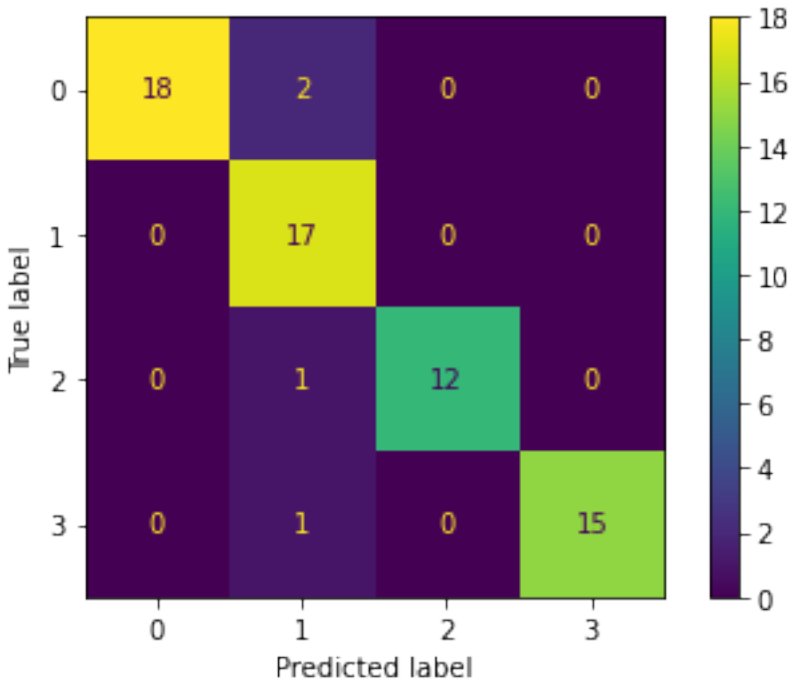
Based on the data presented in Table 5, for the multi-class classification model experiments, the model that performed the best overall was the Random Forest model with TF-IDF vectorization. This model had an accuracy of 0.94. All of the models using TF-IDF vectorization performed very well with accuracy scores above 0.90, while all the models using Doc2Vec vectorization performed poorly with accuracy scores of 0.53 and below. Overall, sci-fi was the movie genre that was best predicted; SVM, Logistic Regression, and Naïve Bayes with TF-IDF were able to predict sci-fi movies with 100% precision and recall, and Random Forest got very close.

For the SVM, Logistic Regression, and Naïve Bayes models with TF-IDF, action movies have 100% recall while comedy and horror movies had 100% precision scores. These three models performed almost identically to one another. However, the Random Forest model with

TF-IDF was able to perform with 100% precision of action, horror, and sci-fi movies and 100% recall comedy movies. The rest of the classification report metrics are very high for this model as well, therefore it got scored the highest on accuracy overall.

Figure 4 shows the confusion matrix for the most accurate model overall, Random Forest model with TF-IDF. All confusion matrices for multi-class classification can be found in the Appendices.

Figure 5. Confusion Matrix Random Forest/TF-IDF



In this figure, the Random Forest model predicted 18 of 20 action movies, 17 of 17 comedy movies, 12 of 13 horror movies, and 15 of 16 sci-fi movies correctly. Two action movies, 1 horror movie, and 1 sci-fi movie were misclassified as comedy movies; no other movies were misclassified. The SVM, Logistic Regression, and Naïve Bayes models with TF-IDF all misclassified a small number of comedy and horror movies as action movies. Some misclassification makes sense because movies often do have elements of multiple genres.

Overall, all the models using TF-IDF vectorization did a good job at predicting movie genre, with the Random Forest model performing the best. The classification of movie genre was modeled well even with the small amount of data in the class corpus likely because there are many mentions of the actual movies' genres within the text of the movie reviews. The model likely would not improve much with any changes.

Topic Modeling

LSA and LDA models were used to determine topics within the class corpus movie reviews. Each experiment output contained separate topics and the 10 words associated with each. Experiments were performed with 2 topics to determine if the topics would be related to positive and negative reviews. In addition, experiments were performed with 4 topics to determine if the topics would be representative of the 4 different movie genres present in the class corpus movie reviews. Experiments with 6 and 20 topics were also performed. All topic and word outputs can be found in the Appendices for each experiment.

The output for LSA and LDA models with 2 topics did not seem to correspond to positive or negative sentiment of movie reviews. Instead, these topics were dominated by words related to movie genre, movie titles, characters, and directors. Similarly, the output for LSA and LDA models with 4 topics did not seem to correspond to different movie genres. In fact, most of the topics contained the word "action" and characters and movie titles related to action and sci-fi movies from the class corpus.

For the LSA model, the experiment with the highest coherence value was 6 topics and 10 words. Looking over the topics, many of them contain the same words. For example, "batman" is near or at the top of the list for 3 of the 6 topics. Other movies represented in these topics include "Minority Report", "Angel Has Fallen", "Toxic Avenger", "Inception", and "Equilibrium". So,

these topics all seem to be related to action and/or sci-fi movies. Some words within the topics are not specifically movie related. In topic 0, the words “movie”, “story”, and “characters” appear, suggesting that the topic could be story or plot related. In topic 1, the words “future” and “technology” appear, suggesting that this topic is more related to science fiction. Overall, it seems the dominating themes in the entire corpus based on this output are related to action and science fiction. It is possible that reviews for these movies were longer than reviews for other types of movies in the corpus, therefore there is more data on these types of movies.

For the LDA model, the experiment with the highest coherence value (although still very low at 0.28), was 20 topics and 10 words. Looking over the topics, some are related to one specific movie. For instance, topic 1 contains words all related to the movie inception, such as “dreams”, “subconscious”, and “inception”. In other topics, it is more difficult to decipher a clear theme. For instance, topic 0 contains words related to both “It: Chapter Two” and “The Revenant” movies.

Overall, since the class corpus is a relatively small set of data, LSA and LDA modeling formed topics with the data that was most prevalent which seemed to be from the action and sci-fi movie reviews.

Conclusions

Multiple experiments were performed using clustering, sentiment analysis, multi-class classification, and topic modeling. The results of the experiments were compared with the goal being to group similar documents together based on movie genre and movie review sentiment.

For the clustering experiments, k-means clustering was performed, and tf-idf was use to determine the top 10 words for each cluster. Overall, the k-means clustering did not do a good job to cluster movies in the same genre together. Performing the clustering with a higher number

of clusters ended up putting more movies into a cluster by themselves, and performing the clustering with fewer clusters did not group movies of the same genre together. So, the k-means clustering worked best for clustering reviews from the same movie together in their own separate clusters. Clustering by genre might improve if specific terms, such as movie characters, actors, etc. were removed from the reviews so that themes related to the genre of the movie remain.

For the sentiment analysis experiments, SVM, Logistic Regression, Naïve Bayes, and Random Forest models were performed with TF-IDF vectorization and Doc2Vec embedding level 200 vectorization. The goal was for the model to correctly identify positive and negative movie reviews. The model that performed the best was Naïve Bayes with Doc2Vec embedding level 200 vectorization. Overall, none of the classification models did a fantastic job at predicting the sentiment of the movie reviews, considering that the highest overall accuracy was 0.55. To improve upon this, perhaps different vectorization methods would yield better results. In addition, having a larger dataset of movie reviews would most likely improve the model since there would be a larger amount of data to train the model with. Another factor could be that movie reviews are not showing sentiment strongly enough. Many movie reviewers take a middle of the road stance, discussing the good and bad about the movie.

For the multi-class classification experiments, SVM, Logistic Regression, Naïve Bayes, and Random Forest models were performed with TF-IDF vectorization and Doc2Vec embedding level 200 vectorization. The goal was for the model to correctly identify the genre of the movie. Overall, all the models using TF-IDF vectorization did a good job at predicting movie genre, with the Random Forest model performing the best. The classification of movie genre was modeled well even with the small amount of data in the class corpus likely because there are

many mentions of the actual movies' genres within the text of the movie reviews. The model likely would not improve much with any changes.

LSA and LDA modeling were used to determine if the similar documents could be clustered together. Experiments were performed with both models for 2 topics and 10 words to determine if the topics could be clustered by sentiment of movie review (positive or negative). In addition, experiments were performed with both models for 4 topics and 10 words to determine if topics could be clustered by genre of the movie (action, comedy, horror, or sci-fi). Experiments were also performed with both models for 6 and 20 topics and 10 words. The LSA and LDA models did not perform well for grouping documents by movie review sentiment or movie genre. Overall, since the class corpus is a relatively small set of data, LSA and LDA modeling formed topics with the data that was most prevalent which seemed to be from the action and sci-fi movie reviews. Having a larger data set would likely improve LSA and LDA modeling for sentiment and movie genre.

Appendices

Table A1. Movie Review Clusters for K-Value = 11

Cluster	Movie Reviews (orange = comedy, green = action, blue = horror, purple = sci-fi)
Cluster 0	Inception (Doc 1- 10)
Cluster 1	Despicable Me 3 (Doc 1- 10) Dirty Grandpa (Doc 1 – 10) Drag Me to Hell (Doc 1- 10) Fresh (Doc 1- 10) Us (Doc 1 – 10) Equilibrium (Doc 1 – 10) Minority Report (Doc 1 – 10) Oblivion (Doc 1 – 10) Pitch Black (Doc 1- 10)
Cluster 2	No Time to Die (Doc 1 – 10) Batman (Doc 1 – 10)
Cluster 3	The Toxic Avenger (Doc 1 – 10)
Cluster 4	Angel Has Fallen (Doc 1 – 10)
Cluster 5	The Lost City (Doc 1- 10)
Cluster 6	It Chapter Two (Doc 1- 10)
Cluster 7	The Revenant (Doc 1 – 10)
Cluster 8	Legally Blonde (Doc 1 – 10)
Cluster 9	Taken (Doc 1- 10)
Cluster 10	Happy Gilmore (Doc 1- 10)

Figure A1. Cluster Plot in Two-Dimensional Plane, K-value = 11

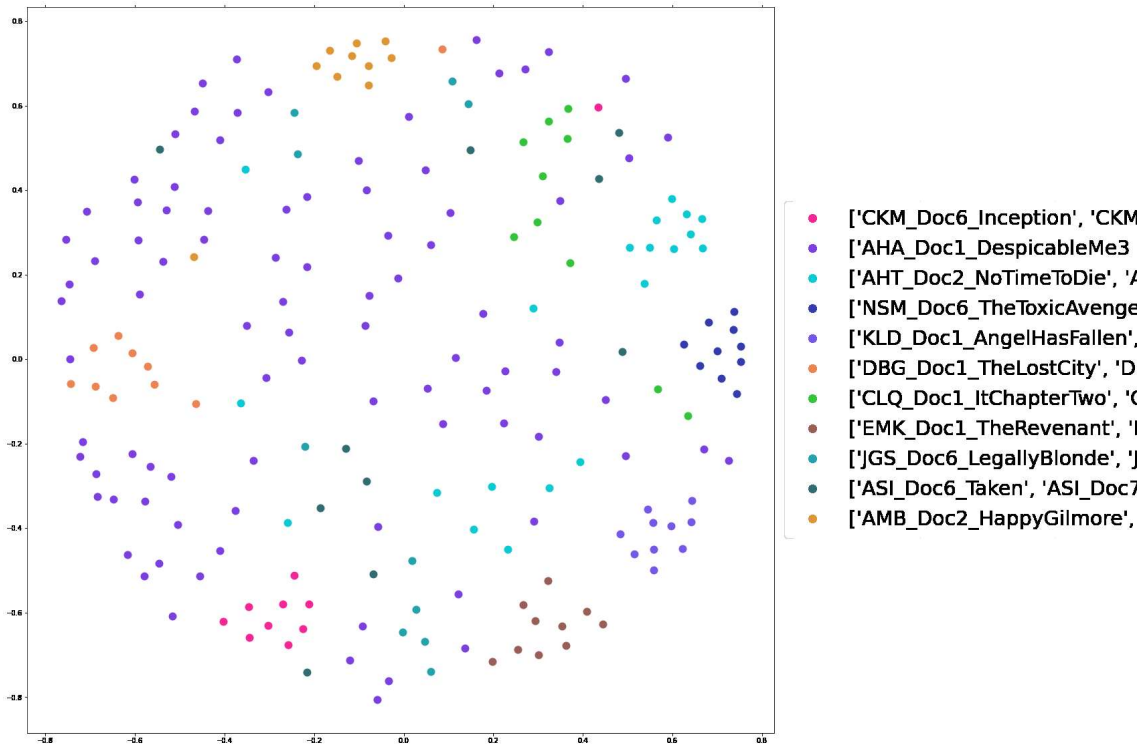


Figure A2. TF-IDF Top 10 Words for Movie Clusters, K-value = 5

Cluster 0:

	term	Average tf-idf score
0	happy	0.3840
1	gilmore	0.2833
2	sandler	0.2726
3	sandlers	0.0995
4	hockey	0.0995
5	chubbs	0.0665
6	shooter	0.0647
7	player	0.0598
8	comedy	0.0578
9	subway	0.0575

Cluster 1:

	term	Average tf-idf score
0	inception	0.0728
1	nolan	0.0653
2	glass	0.0636
3	dream	0.0587
4	spielberg	0.0508
5	cruise	0.0481
6	oblivion	0.0467
7	revenant	0.0465
8	anderton	0.0457
9	report	0.0443

Cluster 2:

	term	Average tf-idf score
0	batman	0.2264
1	craig	0.1195
2	reeves	0.0918
3	james	0.0691
4	gotham	0.0659
5	fukunaga	0.0617
6	riddler	0.0602
7	character	0.0532
8	bruce	0.0503
9	wayne	0.0493

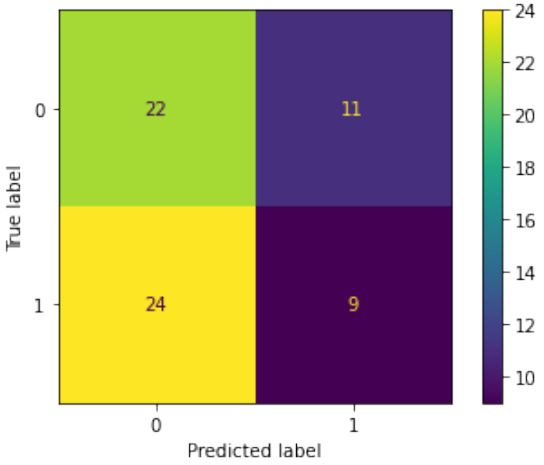
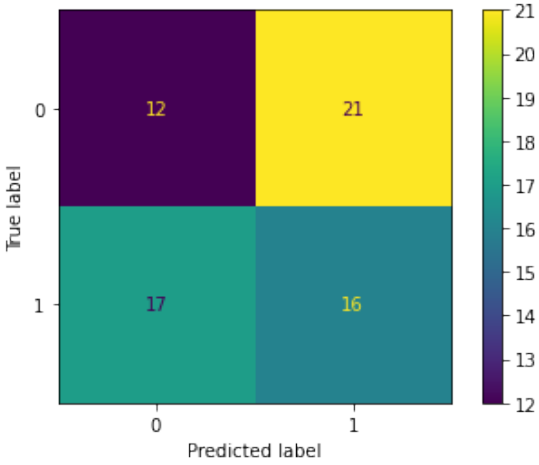
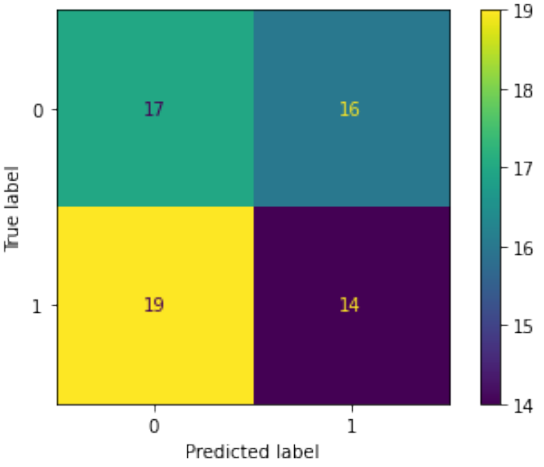
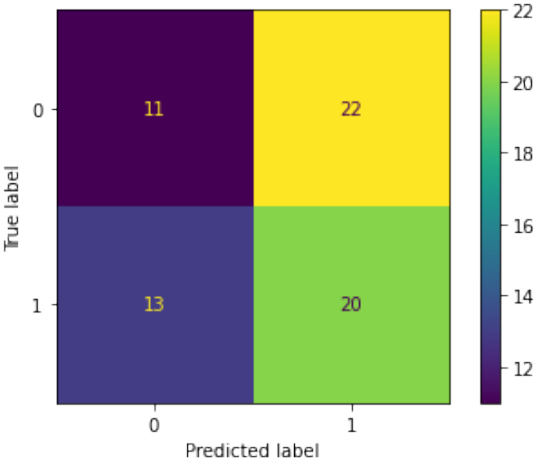
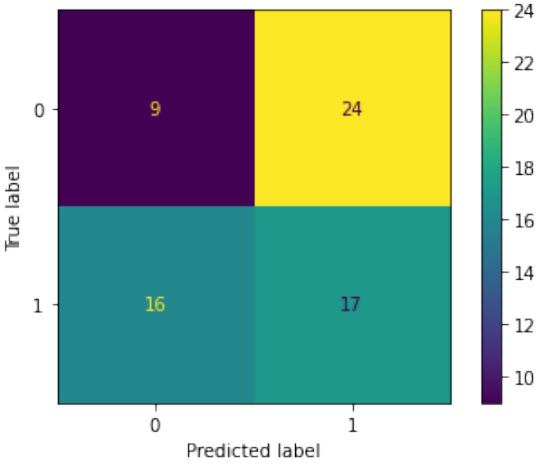
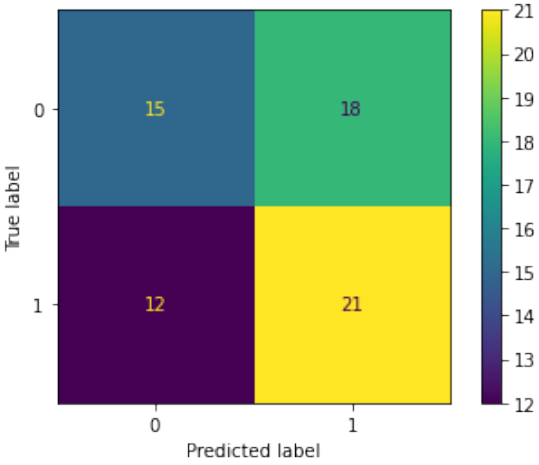
Cluster 3:

	term	Average tf-idf score
0	banning	0.3431
1	fallen	0.2736
2	angel	0.1906
3	president	0.1781
4	butler	0.1153
5	service	0.0991
6	agent	0.0876
7	secret	0.0859
8	freeman	0.0736
9	trumbull	0.0711

Cluster 4:

	term	Average tf-idf score
0	movie	0.0588
1	toxic	0.0352
2	horror	0.0323
3	comedy	0.0279
4	peele	0.0276
5	avenger	0.0258
6	character	0.0236
7	loretta	0.0221
8	troma	0.0208
9	chapter	0.0205

Figure A3. Confusion Matrices for Sentiment Analysis

	TF-IDF	Doc2Vec, 200
SVM		
Logistic Regression		
Naïve Bayes		

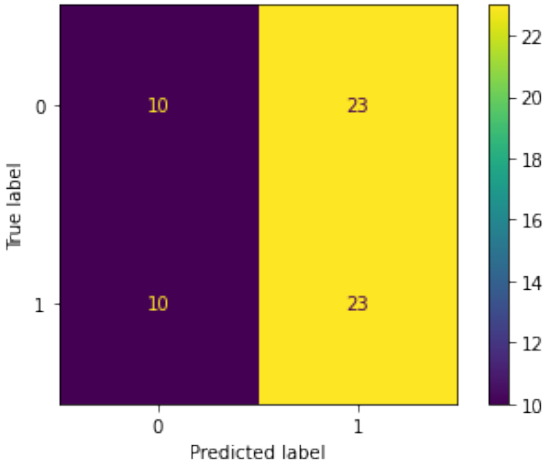
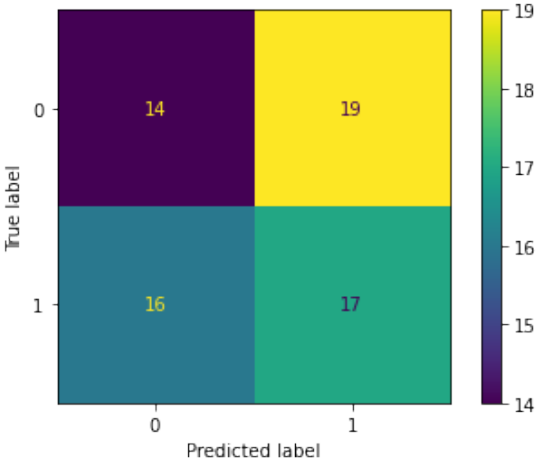
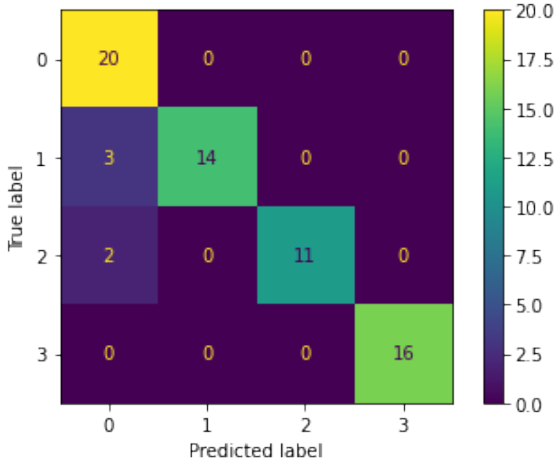
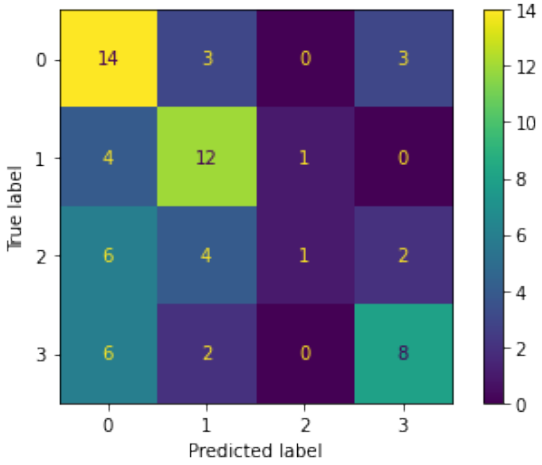
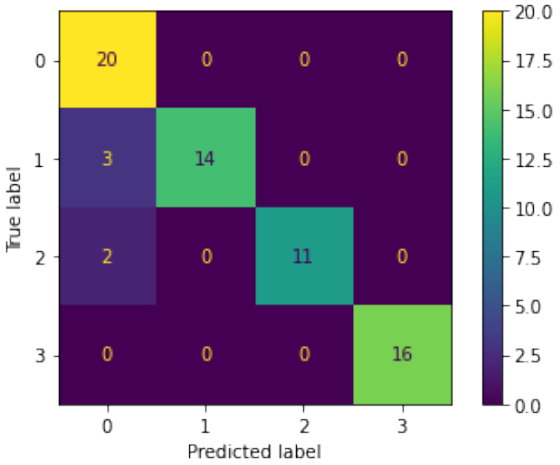
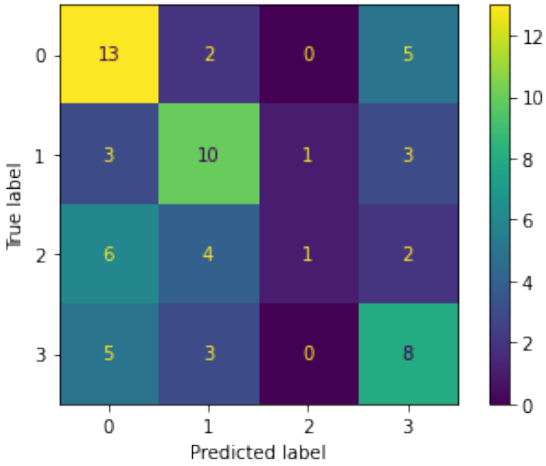
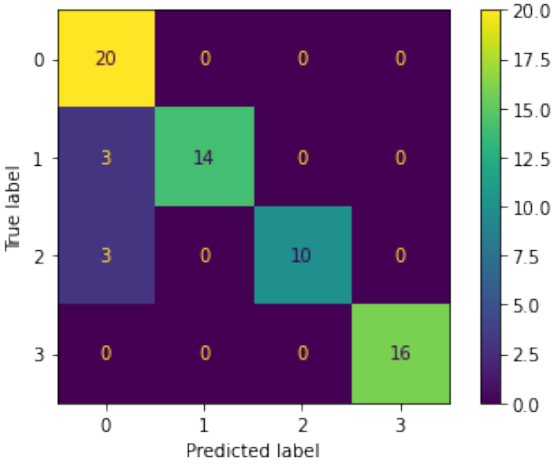
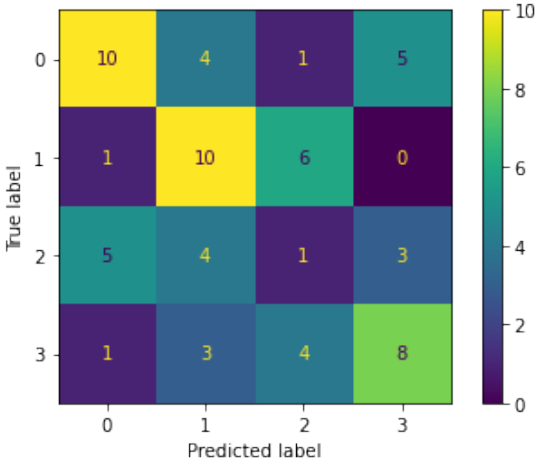
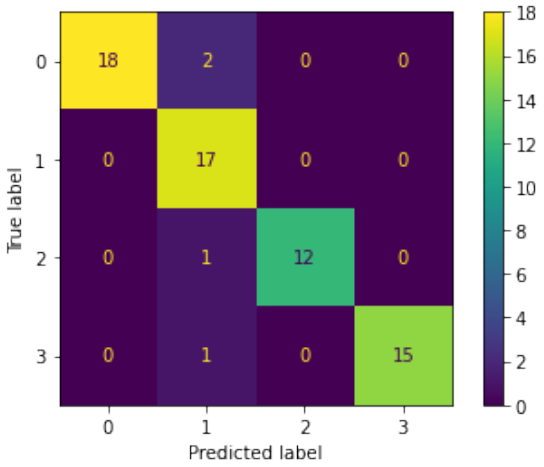
	TF-IDF	Doc2Vec, 200
Random Forest		

Figure A4. Confusion Matrices for Multi-Class Classification

	TF-IDF	Doc2Vec, 200
SVM		
Logistic Regression		
Naïve Bayes		

Random Forest

TF-IDF



Doc2Vec, 200

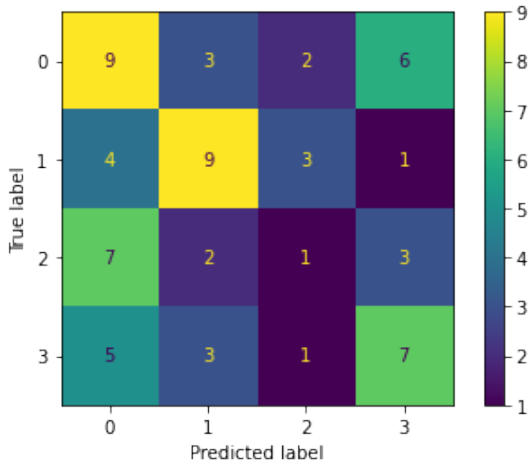


Figure A5. LSA Experiments Topics and Words

2 topics, 10 words

```
[(0, '0.377*"movie" + 0.283*"batman" + 0.136*"character" + 0.131*"anderton" +
0.130*"would" + 0.129*"story" + 0.124*"characters" + 0.116*"first" +
0.116*"action" + 0.110*"spielberg"'),
(1, '-0.454*"batman" + 0.411*"anderton" + 0.279*"spielberg" +
0.257*"minority" + 0.256*"report" + 0.173*"precrime" + 0.146*"future" +
0.135*"technology" + -0.127*"reeves" + -0.101*"movie"')]
```

4 topics, 10 words

```
[(0, '0.377*"movie" + 0.283*"batman" + 0.136*"character" + 0.131*"anderton" +
0.130*"would" + 0.129*"story" + 0.124*"characters" + 0.116*"first" +
0.116*"action" + 0.110*"spielberg"'),
(1, '-0.454*"batman" + 0.411*"anderton" + 0.279*"spielberg" +
0.257*"minority" + 0.256*"report" + 0.173*"precrime" + 0.146*"future" +
0.135*"technology" + -0.127*"reeves" + -0.101*"movie"'),
(2, '0.546*"batman" + -0.263*"movie" + -0.241*"toxic" + 0.229*"anderton" +
0.154*"reeves" + -0.144*"avenger" + 0.137*"spielberg" + 0.126*"report" +
0.125*"minority" + -0.123*"action"'),
(3, '-0.513*"toxic" + -0.325*"avenger" + -0.248*"troma" + -0.242*"melvin" +
0.198*"action" + 0.152*"banning" + -0.140*"toxie" + -0.102*"batman" +
0.099*"fallen" + -0.097*"anderton"')]
```

6 topics, 10 words

```
[(0, '0.377*"movie" + 0.283*"batman" + 0.136*"character" + 0.131*"anderton" +
0.130*"would" + 0.129*"story" + 0.124*"characters" + 0.116*"first" +
0.116*"action" + 0.110*"spielberg"'),
(1, '-0.454*"batman" + 0.411*"anderton" + 0.279*"spielberg" +
0.257*"minority" + 0.256*"report" + 0.173*"precrime" + 0.146*"future" +
0.135*"technology" + -0.127*"reeves" + -0.101*"movie"'),
(2, '-0.546*"batman" + 0.263*"movie" + 0.241*"toxic" + -0.229*"anderton" + -
0.154*"reeves" + 0.144*"avenger" + -0.137*"spielberg" + -0.126*"report" + -
0.125*"minority" + 0.123*"action"'),
(3, '-0.513*"toxic" + -0.325*"avenger" + -0.248*"troma" + -0.242*"melvin" +
0.198*"action" + 0.152*"banning" + -0.140*"toxie" + -0.102*"batman" +
0.099*"fallen" + -0.097*"anderton"'),
(4, '-0.264*"movie" + 0.228*"banning" + -0.213*"equilibrium" +
0.171*"inception" + 0.151*"nolan" + 0.146*"fallen" + -0.139*"future" + -
0.123*"wimmer" + -0.108*"action" + 0.102*"angel"'),
(5, '0.456*"banning" + 0.297*"fallen" + -0.271*"inception" + -0.248*"nolan" +
0.207*"angel" + 0.179*"action" + 0.170*"president" + 0.149*"butler" +
0.130*"service" + 0.120*"agent"')]
```

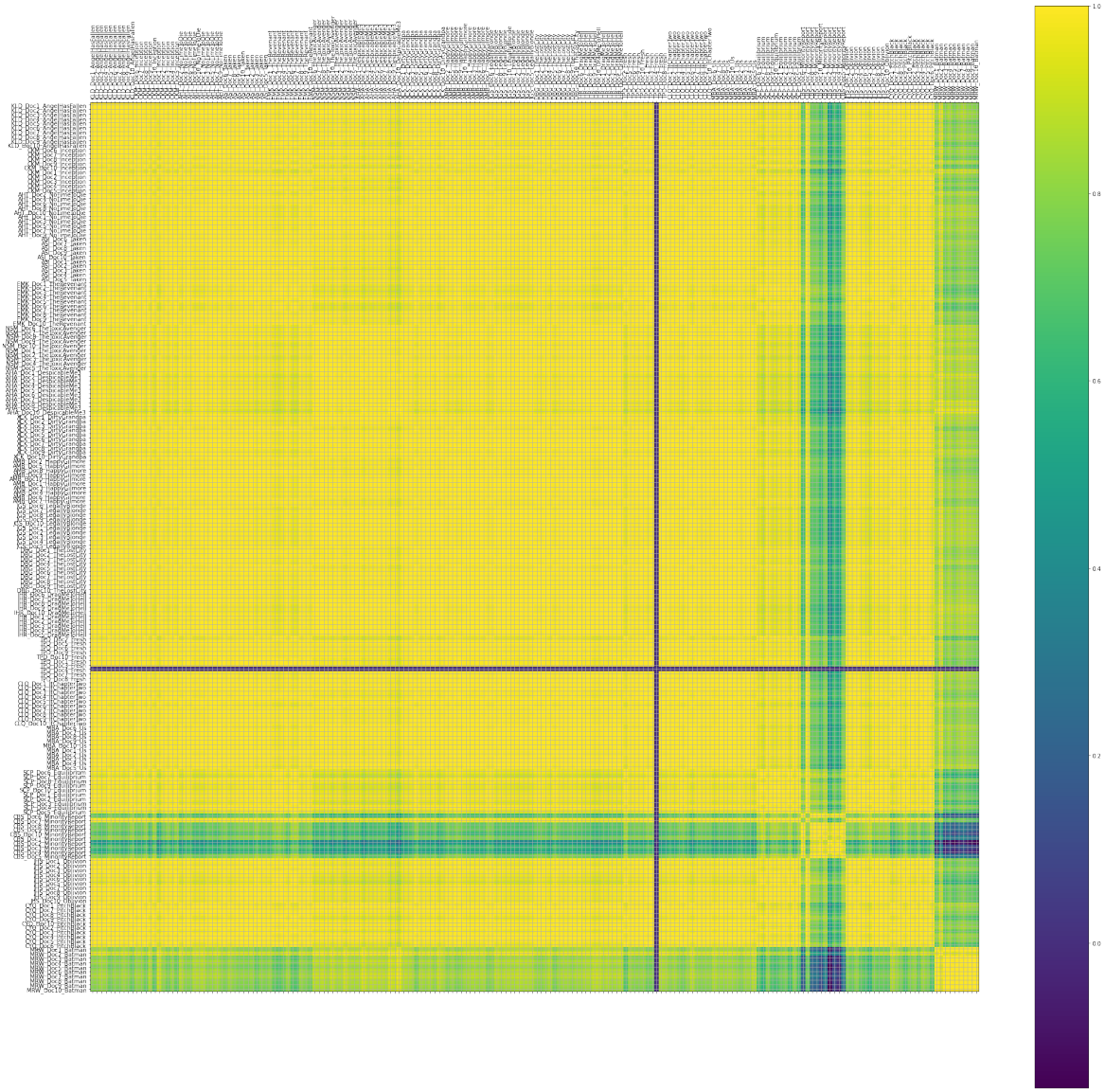
20 topics, 10 words

```
[(0, '-0.377*"movie" + -0.283*"batman" + -0.136*"character" + -
0.131*"anderton" + -0.130*"would" + -0.129*"story" + -0.124*"characters" + -
0.116*"first" + -0.116*"action" + -0.110*"spielberg"'),
```

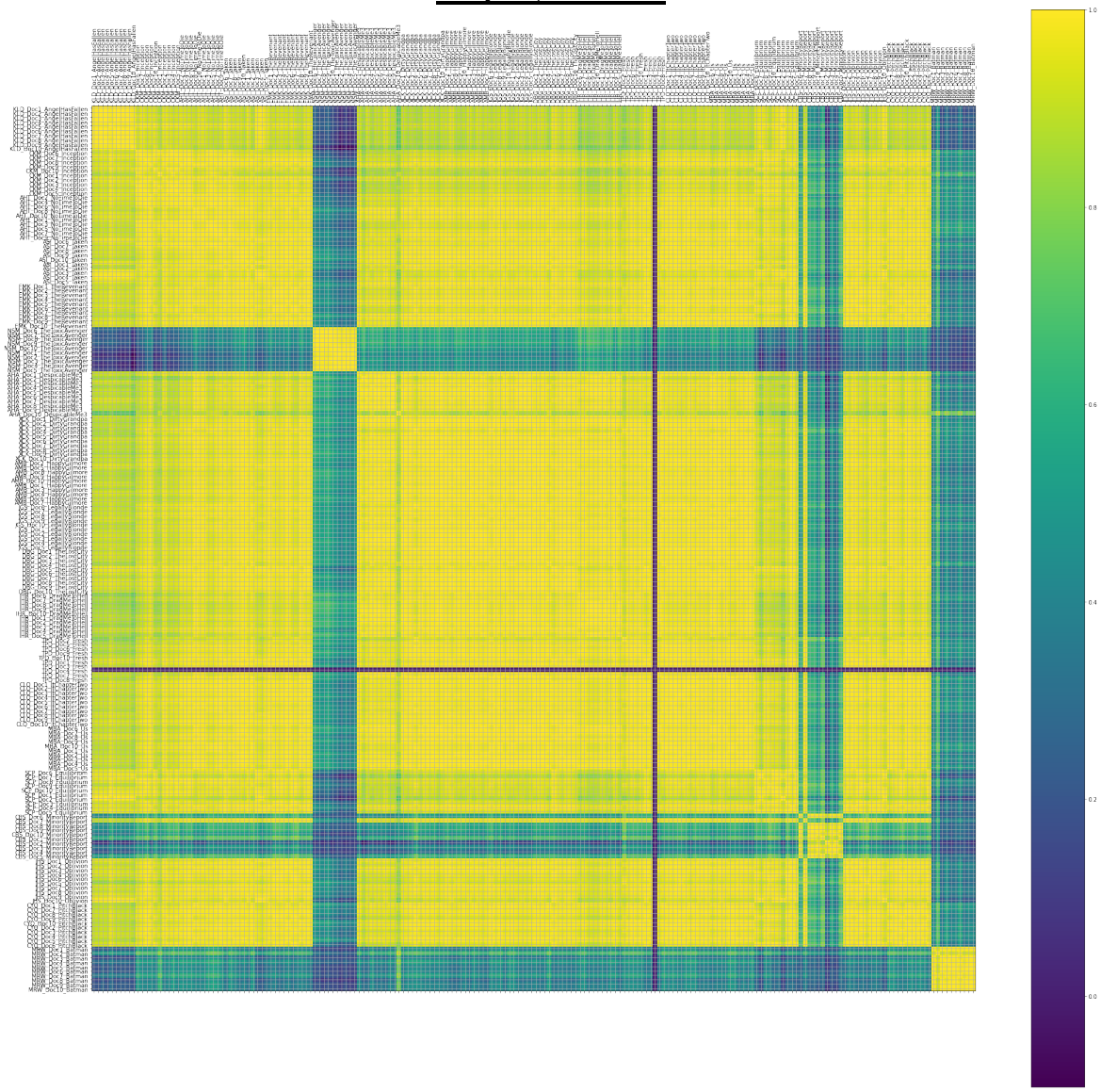
```
(1, '0.454*"batman" + -0.411*"anderton" + -0.279*"spielberg" + -
0.257*"minority" + -0.256*"report" + -0.173*"precrime" + -0.146*"future" + -
0.135*"technology" + 0.127*"reeves" + 0.101*"movie"'),
(2, '0.546*"batman" + -0.263*"movie" + -0.241*"toxic" + 0.229*"anderton" +
0.154*"reeves" + -0.144*"avenger" + 0.137*"spielberg" + 0.126*"report" +
0.125*"minority" + -0.123*"action"'),
(3, '0.513*"toxic" + 0.325*"avenger" + 0.248*"troma" + 0.242*"melvin" + -
0.198*"action" + -0.152*"banning" + 0.140*"toxie" + 0.102*"batman" + -
0.099*"fallen" + 0.097*"anderton"'),
(4, '-0.264*"movie" + 0.228*"banning" + -0.213*"equilibrium" +
0.171*"inception" + 0.151*"nolan" + 0.146*"fallen" + -0.139*"future" + -
0.123*"wimmer" + -0.108*"action" + 0.102*"angel"'),
(5, '0.456*"banning" + 0.297*"fallen" + -0.271*"inception" + -0.248*"nolan" +
0.207*"angel" + 0.179*"action" + 0.170*"president" + 0.149*"butler" +
0.130*"service" + 0.120*"agent"'),
(6, '-0.457*"inception" + -0.416*"nolan" + -0.178*"action" + 0.158*"happy" +
-0.155*"dreams" + -0.117*"toxic" + -0.113*"dream" + 0.109*"movie" +
0.103*"horror" + 0.102*"chapter"'),
(7, '0.528*"happy" + 0.309*"gilmore" + 0.259*"sandler" + -0.206*"chapter" + -
0.177*"pennywise" + -0.129*"horror" + -0.122*"muschietti" + -0.119*"first" +
0.117*"movie" + 0.108*"hockey"'),
(8, '0.318*"happy" + -0.268*"movie" + 0.236*"chapter" + 0.215*"pennywise" +
0.184*"gilmore" + 0.160*"sandler" + 0.141*"muschietti" + 0.131*"first" +
0.114*"losers" + -0.109*"people"'),
(9, '-0.250*"movie" + 0.207*"glass" + 0.178*"peele" + -0.143*"chapter" +
0.143*"black" + 0.142*"american" + -0.135*"pennywise" + -0.126*"scene" + -
0.125*"blonde" + 0.119*"happy"'),
(10, '0.330*"peele" + -0.309*"glass" + 0.241*"horror" + 0.188*"black" +
0.162*"family" + -0.162*"revenant" + -0.147*"story" + 0.135*"adelaide" + -
0.117*"rritu" + -0.109*"american"'),
(11, '0.237*"glass" + 0.235*"peele" + 0.184*"horror" + 0.126*"family" + -
0.124*"chapter" + 0.124*"anderton" + 0.121*"revenant" + -0.120*"planet" + -
0.119*"pitch" + -0.118*"riddick"'),
(12, '0.175*"pitch" + 0.173*"riddick" + -0.154*"people" + -0.145*"spielberg"
+ 0.142*"black" + -0.136*"american" + 0.130*"loretta" + 0.127*"planet" +
0.126*"bullock" + -0.116*"glass"'),
(13, '0.230*"glass" + 0.156*"riddick" + -0.151*"loretta" + 0.149*"pitch" +
0.147*"planet" + -0.143*"bullock" + 0.138*"black" + -0.132*"tatum" +
0.127*"revenant" + -0.125*"spielberg"'),
(14, '-0.249*"woman" + 0.229*"movie" + -0.202*"women" + 0.145*"glass" +
0.123*"loretta" + -0.120*"neeson" + 0.119*"story" + -0.111*"first" +
0.110*"bullock" + -0.108*"bryan"'),
(15, '-0.250*"blonde" + 0.233*"loretta" + -0.221*"legally" + 0.221*"bullock"
+ 0.202*"tatum" + -0.155*"character" + 0.142*"oblivion" + -
0.140*"witherspoon" + -0.138*"harvard" + -0.127*"doesnt"'),
(16, '-0.191*"oblivion" + -0.172*"taste" + 0.162*"melvin" + -0.146*"cruise" +
-0.134*"troma" + -0.133*"kosinski" + 0.131*"people" + -0.131*"blonde" + -
0.118*"legally" + 0.113*"craig"'),
(17, '0.194*"oblivion" + 0.178*"craig" + 0.146*"cruise" + -0.138*"black" + -
0.136*"pitch" + -0.133*"loretta" + -0.133*"riddick" + 0.129*"fukunaga" +
0.128*"kosinski" + -0.126*"bullock"'),
(18, '0.282*"neeson" + 0.253*"bryan" + 0.198*"taken" + 0.159*"daughter" +
0.135*"jason" + -0.134*"horror" + -0.127*"story" + 0.116*"mills" +
0.111*"paris" + -0.107*"woman"'),
(19, '-0.181*"craig" + 0.147*"paycheck" + -0.136*"blonde" + -0.129*"oblivion"
+ -0.127*"legally" + -0.122*" ukunaga" + -0.110*"cruise" + 0.106*"actually" +
0.105*"story" + -0.100*"james"')]
```

Figure A6. LSA Experiments Cosine Similarity Matrices

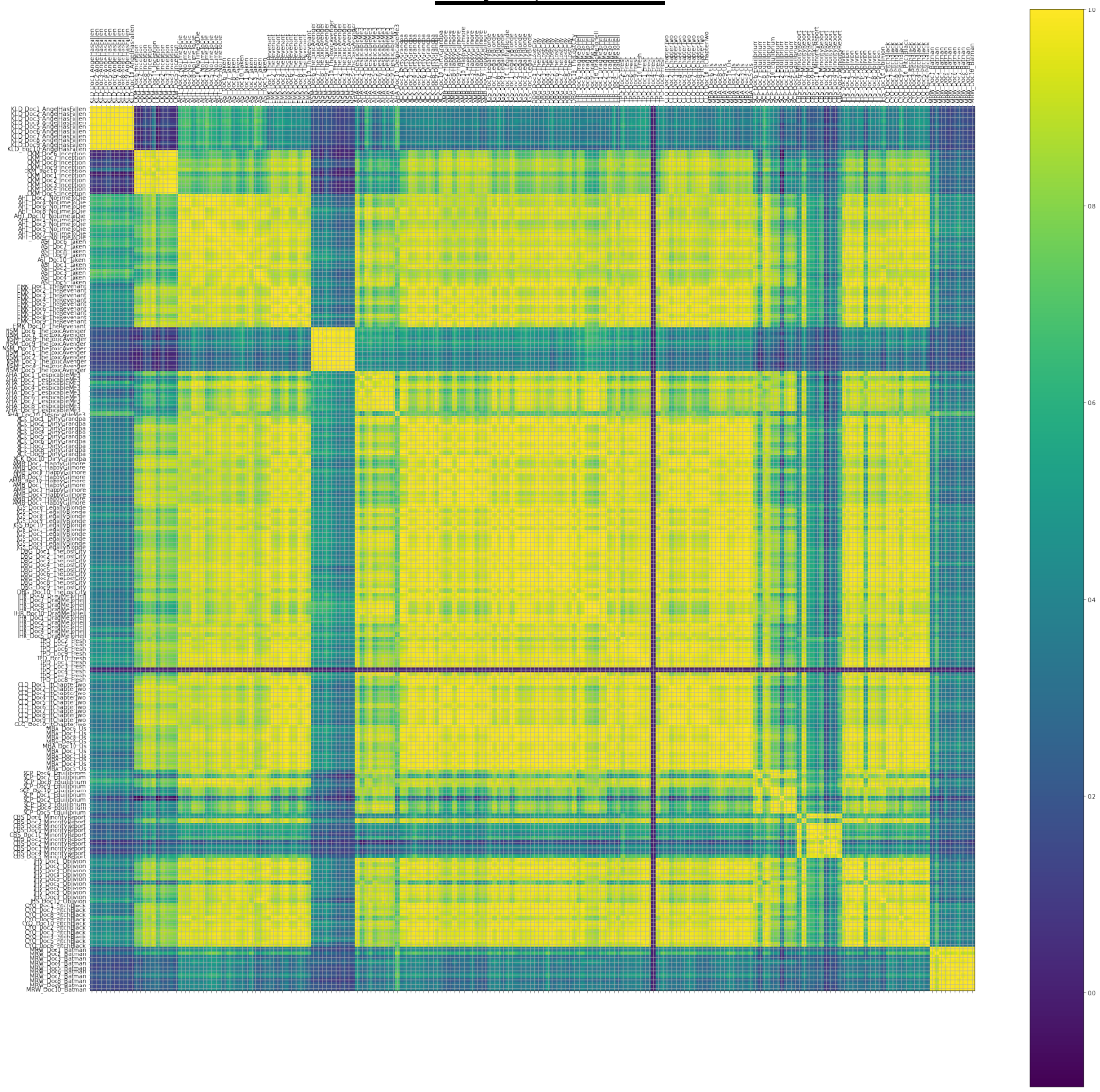
2 topics, 10 words



4 topics, 10 words



6 topics, 10 words



20 topics, 10 words

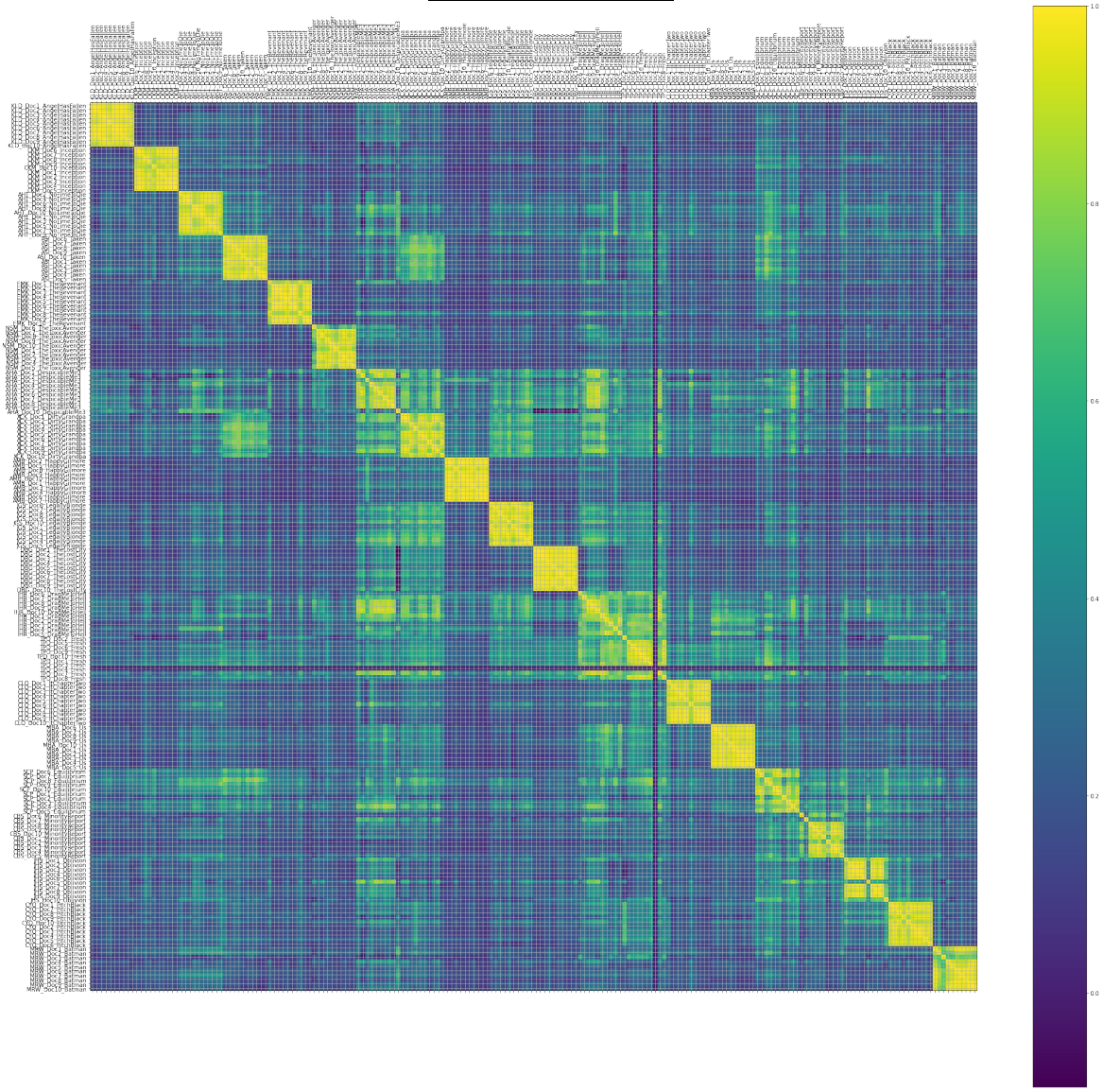


Figure A7. LDA Experiments Topics and Words

2 topics, 10 words

```
[(0, '0.008*"movie" + 0.003*"first" + 0.003*"batman" + 0.003*"character" +
0.003*"story" + 0.003*"action" + 0.003*"would" + 0.003*"characters" +
0.003*"films" + 0.002*"could"'),
(1, '0.008*"movie" + 0.003*"action" + 0.003*"character" + 0.003*"story" +
0.003*"first" + 0.003*"would" + 0.003*"batman" + 0.003*"films" +
0.003*"characters" + 0.002*"could"')]
```

4 topics, 10 words

```
[(0, '0.008*"movie" + 0.003*"first" + 0.003*"action" + 0.003*"story" +
0.003*"character" + 0.003*"characters" + 0.003*"would" + 0.003*"films" +
0.003*"horror" + 0.002*"movies"'),
(1, '0.008*"movie" + 0.003*"story" + 0.003*"would" + 0.003*"action" +
0.003*"character" + 0.003*"first" + 0.003*"films" + 0.003*"world" +
0.003*"characters" + 0.003*"people"'),
(2, '0.009*"movie" + 0.003*"action" + 0.003*"character" + 0.003*"would" +
0.003*"first" + 0.003*"story" + 0.002*"films" + 0.002*"characters" +
0.002*"could" + 0.002*"movies"'),
(3, '0.009*"movie" + 0.004*"batman" + 0.003*"character" + 0.003*"story" +
0.003*"first" + 0.003*"action" + 0.003*"characters" + 0.003*"would" +
0.003*"films" + 0.002*"could"')]
```

6 topics, 10 words

```
[(0, '0.008*"movie" + 0.004*"first" + 0.003*"story" + 0.003*"character" +
0.003*"action" + 0.003*"characters" + 0.002*"could" + 0.002*"films" +
0.002*"never" + 0.002*"would"'),
(1, '0.007*"movie" + 0.003*"story" + 0.003*"action" + 0.003*"would" +
0.003*"character" + 0.003*"anderton" + 0.003*"first" + 0.003*"films" +
0.003*"world" + 0.003*"nolan"'),
(2, '0.008*"movie" + 0.003*"character" + 0.003*"would" + 0.003*"action" +
0.003*"first" + 0.003*"story" + 0.003*"characters" + 0.002*"films" +
0.002*"could" + 0.002*"years"'),
(3, '0.008*"movie" + 0.003*"character" + 0.003*"action" + 0.003*"first" +
0.003*"story" + 0.003*"films" + 0.002*"characters" + 0.002*"would" +
0.002*"could" + 0.002*"still"'),
(4, '0.010*"movie" + 0.004*"batman" + 0.003*"toxic" + 0.003*"would" +
0.003*"first" + 0.003*"action" + 0.003*"films" + 0.003*"character" +
0.003*"story" + 0.003*"movies"'),
(5, '0.009*"movie" + 0.004*"batman" + 0.004*"character" + 0.003*"first" +
0.003*"action" + 0.003*"happy" + 0.003*"would" + 0.003*"characters" +
0.003*"films" + 0.002*"story"')]
```

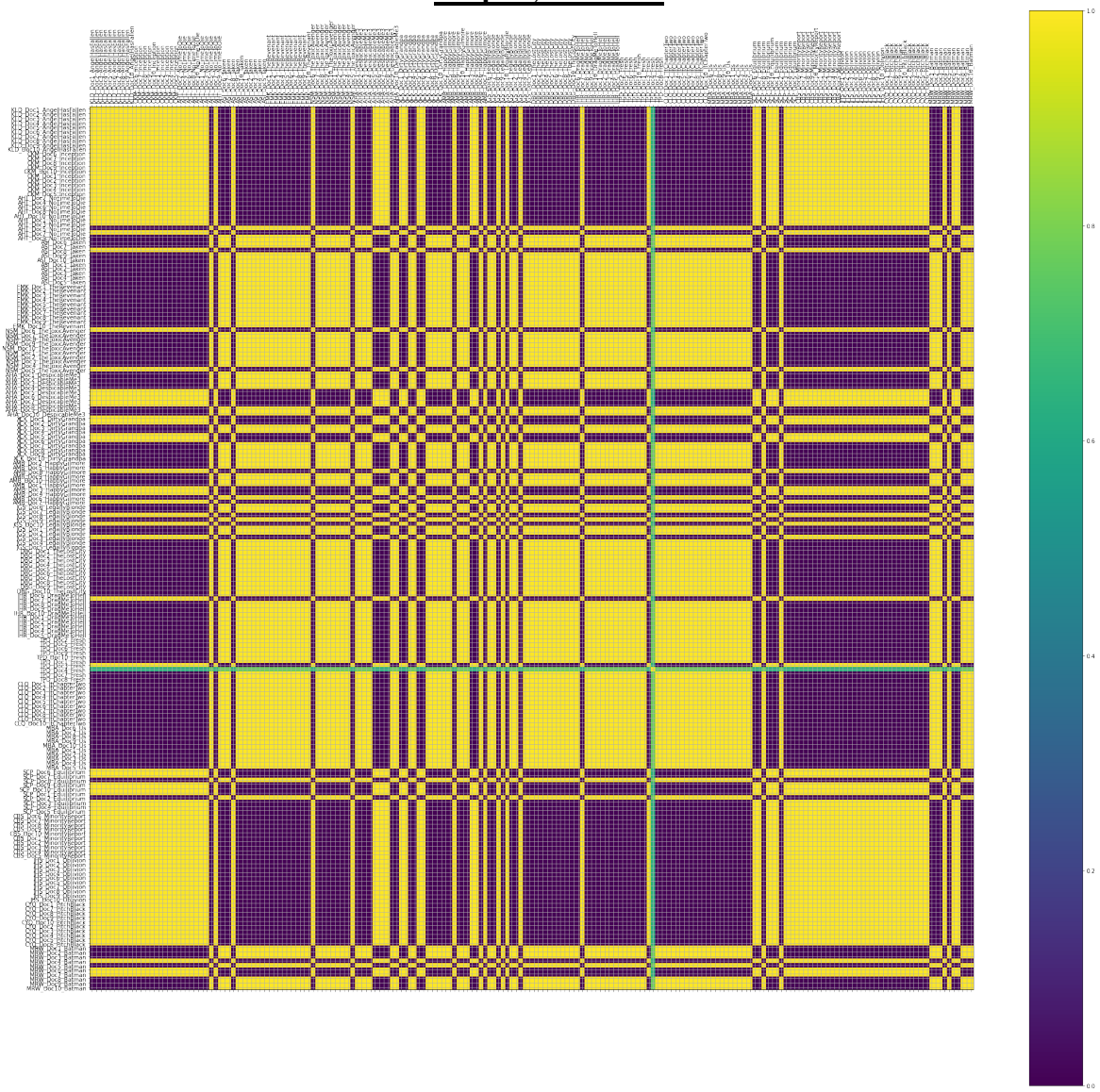
20 topics, 10 words

```
[(0, '0.007*"glass" + 0.006*"pennywise" + 0.006*"movie" + 0.006*"chapter" +
0.005*"story" + 0.005*"first" + 0.004*"characters" + 0.003*"muschietti" +
0.003*"revenant" + 0.003*"another"'),
```

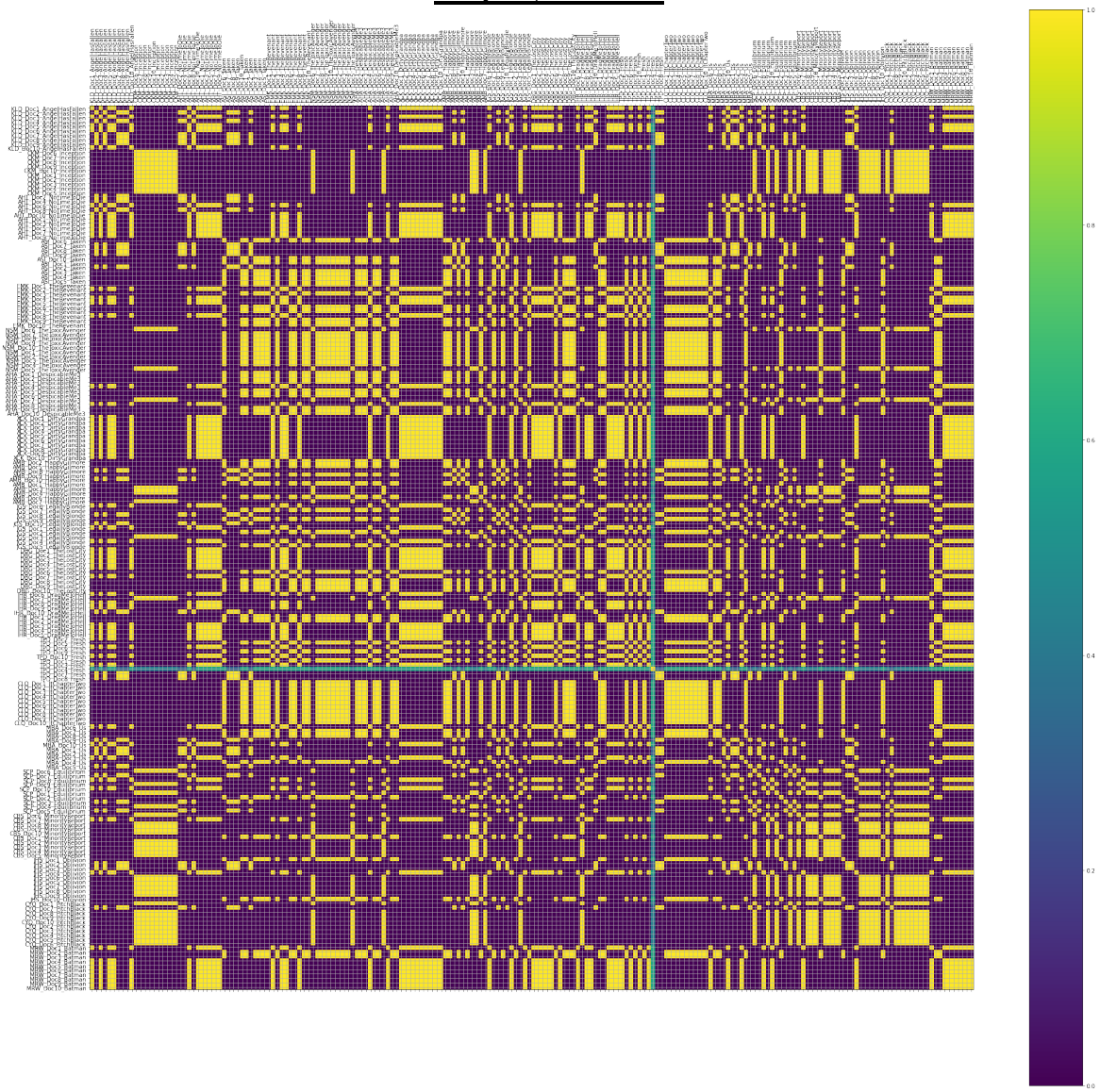
```
(1, '0.010*"inception" + 0.009*"nolan" + 0.008*"movie" + 0.005*"dreams" +
0.004*"dream" + 0.003*"subconscious" + 0.003*"could" + 0.003*"story" +
0.003*"world" + 0.003*"dicaprio"'),
(2, '0.007*"movie" + 0.003*"wendy" + 0.003*"character" + 0.003*"fallen" +
0.003*"would" + 0.003*"first" + 0.002*"story" + 0.002*"action" +
0.002*"characters" + 0.002*"people"'),
(3, '0.007*"movie" + 0.003*"story" + 0.003*"character" + 0.003*"though" +
0.003*"action" + 0.003*"still" + 0.002*"glass" + 0.002*"first" +
0.002*"banning" + 0.002*"could"'),
(4, '0.009*"toxic" + 0.009*"batman" + 0.009*"movie" + 0.006*"avenger" +
0.005*"troma" + 0.004*"melvin" + 0.004*"people" + 0.004*"films" +
0.004*"scene" + 0.003*"story"'),
(5, '0.010*"movie" + 0.006*"peele" + 0.005*"action" + 0.004*"first" +
0.003*"horror" + 0.003*"family" + 0.003*"though" + 0.003*"world" +
0.003*"equilibrium" + 0.002*"feels"'),
(6, '0.008*"movie" + 0.005*"black" + 0.005*"riddick" + 0.005*"pitch" +
0.004*"planet" + 0.004*"characters" + 0.003*"character" + 0.003*"action" +
0.003*"would" + 0.003*"diesel"'),
(7, '0.008*"anderton" + 0.007*"spielberg" + 0.005*"report" + 0.005*"minority"
+ 0.005*"movie" + 0.003*"precrime" + 0.003*"story" + 0.003*"scene" +
0.003*"future" + 0.003*"characters"'),
(8, '0.011*"movie" + 0.009*"happy" + 0.005*"gilmore" + 0.004*"character" +
0.004*"sandler" + 0.003*"films" + 0.003*"movies" + 0.003*"people" +
0.003*"would" + 0.003*"story"'),
(9, '0.008*"movie" + 0.004*"action" + 0.004*"would" + 0.004*"character" +
0.004*"alison" + 0.004*"horror" + 0.003*"first" + 0.003*"could" +
0.003*"movies" + 0.003*"story"'),
(10, '0.011*"movie" + 0.007*"batman" + 0.004*"would" + 0.004*"character" +
0.003*"characters" + 0.003*"years" + 0.003*"fallen" + 0.003*"films" +
0.003*"people" + 0.002*"story"'),
(11, '0.009*"movie" + 0.004*"character" + 0.004*"story" + 0.004*"horror" +
0.004*"first" + 0.003*"would" + 0.003*"little" + 0.003*"something" +
0.003*"films" + 0.002*"better"'),
(12, '0.009*"movie" + 0.004*"first" + 0.004*"character" + 0.003*"banning" +
0.003*"steve" + 0.003*"action" + 0.003*"little" + 0.003*"would" +
0.003*"though" + 0.002*"fresh"'),
(13, '0.007*"movie" + 0.004*"future" + 0.004*"story" + 0.004*"spielberg" +
0.003*"report" + 0.003*"character" + 0.003*"minority" + 0.003*"anderton" +
0.003*"years" + 0.003*"precrime"'),
(14, '0.008*"movie" + 0.004*"happy" + 0.004*"character" + 0.003*"little" +
0.003*"would" + 0.003*"films" + 0.003*"first" + 0.003*"action" +
0.003*"gilmore" + 0.003*"sandler"'),
(15, '0.008*"movie" + 0.004*"action" + 0.004*"craig" + 0.003*"james" +
0.003*"character" + 0.003*"films" + 0.003*"first" + 0.003*"characters" +
0.003*"movies" + 0.003*"story"'),
(16, '0.008*"movie" + 0.005*"action" + 0.004*"world" + 0.003*"banning" +
0.003*"first" + 0.003*"story" + 0.003*"sense" + 0.003*"could" + 0.003*"films"
+ 0.002*"would"'),
(17, '0.008*"movie" + 0.004*"character" + 0.003*"blonde" + 0.003*"legally" +
0.003*"first" + 0.003*"scene" + 0.003*"movies" + 0.003*"could" +
0.003*"happy" + 0.003*"characters"'),
(18, '0.007*"movie" + 0.004*"first" + 0.004*"horror" + 0.003*"nolan" +
0.003*"novel" + 0.003*"chapter" + 0.003*"would" + 0.003*"story" +
0.003*"another" + 0.003*"inception"'),
(19, '0.008*"movie" + 0.003*"peele" + 0.003*"horror" + 0.003*"toxic" +
0.003*"films" + 0.003*"taste" + 0.003*"first" + 0.003*"avenger" +
0.003*"would" + 0.002*"character"')]
```

Figure A8. LDA Experiments Cosine Similarity Matrices

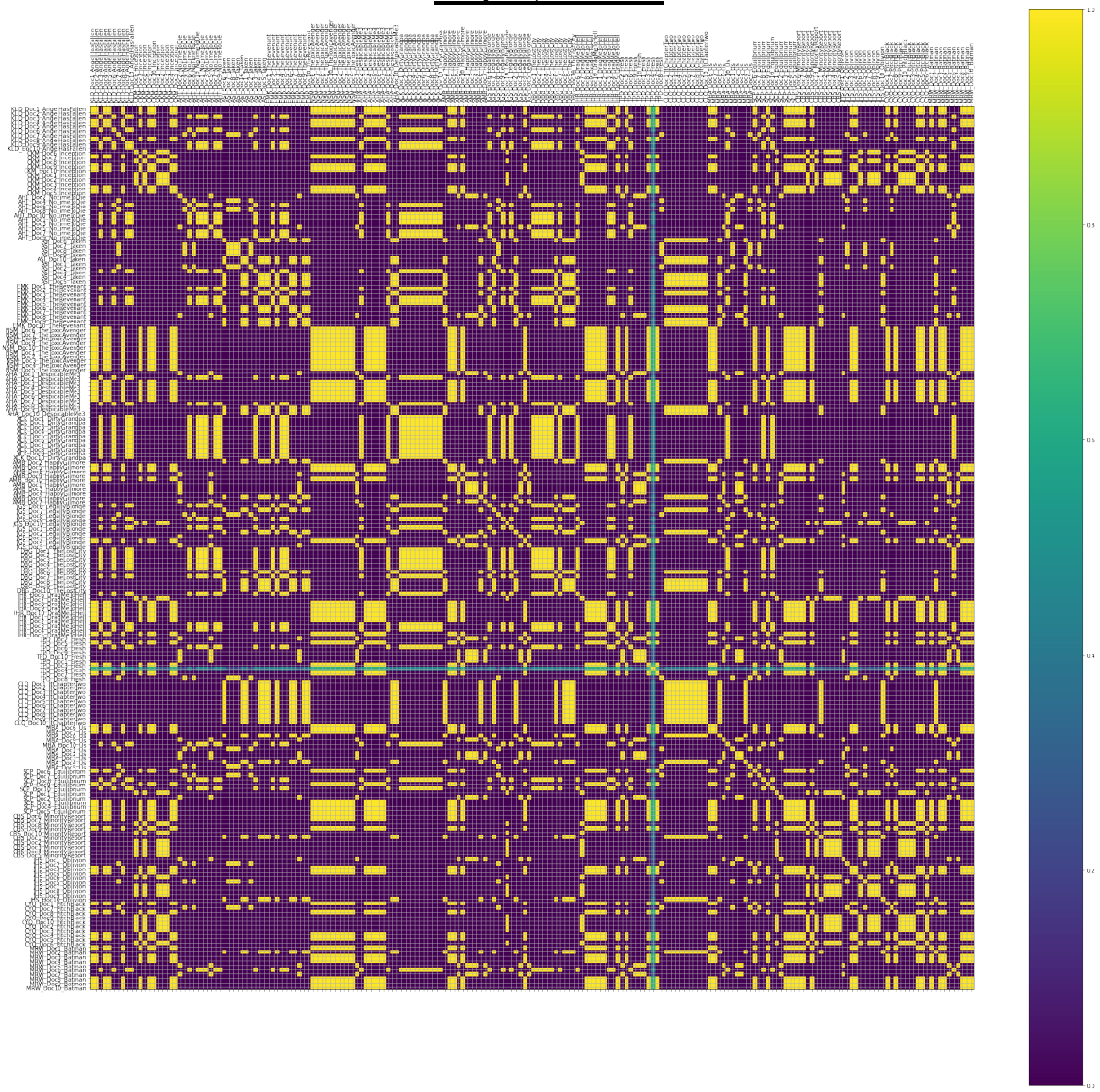
2 topics, 10 words



4 topics, 10 words



6 topics, 10 words



20 topics, 10 words