DATA WRANGLING AND VECTORIZATION TECHNIQUES FOR NLP CLUSTERING

Betzalel Moskowitz
MSDS453-DL: Natural Language Processing
October 31, 2023

Introduction

While strong consideration of different model architectures is necessary to achieve sufficient performance on many natural language processing (NLP) business problems, performance is often better improved through the use of higher-quality data. Effective data wrangling and exploration should always be undertaken in any natural language processing task to ensure that the data is clean and represented in a manner most conducive to clustering, semantic segmentation, classification, and generative models. This report aims to define the most effective data wrangling and vectorization techniques for clustering and grouping similar documents together. The particular effectiveness of TF-IDF vectorization, Word2Vec embeddings, and data wrangling techniques is illustrated in the report through applications to the business problem of clustering and grouping together similar movie reviews from a class-assembled corpus.

Methods

Before any experiments were conducted, a qualitative review of terms extracted from the movie review of interest ("*Equilibrium")* was undertaken. This involved using human intuition and experience based on reading movie reviews to identify terms that were believed to be most useful for distinguishing reviews of "*Equilibrium*" from other movie reviews. To validate these qualitative judgements, TF-IDF scores were calculated using the frequency of term appearances over the ten movie reviews. In checking that these terms are clustered together, these "prevalent terms" were used to assess the performance of varying techniques during experimentation.

As such, experiments were conducted to determine the most effective data wrangling steps to cluster together similar movie reviews. Three different data wrangling steps were assessed:

I. *Tokenization + normalization (Baseline)*
II. Baseline + *lemmatization + remove stop words*
III. Baseline + lemmatization + remove stop words + *remove non-alphabetic tokens + remove custom stop words*

Tokenization was conducted using NLTK's word_tokenize. Normalization was conducted by removing punctuation, tags, and non-alphabetic tokens (only for Data Wrangling III). The text was also converted to lowercase. Any lemmatization was applied using NLTK's WordNetLemmatizer.

Custom stop words selected for removal were derived from a list of tokens that had high mean TF-IDF scores among all movie reviews. These were terms like – but not limited to – 'film', 'movie', 'ha', 'wa', 'like', 'get', and 'character'. They can be seen in *A1* in the 'Data Wrangling II' column of row '*Top Terms By TF-IDF'.*

In order to represent tokens and documents as a vector, three vectorization techniques were tested: sklearn's TfidfVectorizer (Scikit-Learn Developers) , gensim's Word2Vec (Řehůřek, 2022)., and gensim's Doc2Vec (Řehůřek, 2022). The latter two vectorizers produce dense embeddings and therefore require embedding-size parameters. Experiments were carried out using embedding-sizes of 100, 200, and 300 for both Word2Vec and Doc2Vec vectorizers.

To assess the effectiveness of the various data wrangling methods among TF-IDF vectors, cosine similarity matrices were computed and visualized in the 'TF-IDF Embedding Heatmaps' in *A1*. For the Word2Vec models, three visualizations were generated for each combination of embedding size (100, 200, 300) and data wrangling method to compare the ability to determine the similarity of word tokens across the corpus. The three visualizations were similarity matrices, similarity hierarchical cluster maps, and t-Stochastic Neighbor Embedding (t-SNE) plots. While the tokens visualized were selected using random choice (with the addition of the identified prevalent terms), the same random seed was used across all experiments to ensure that the experiments visualizations could be accurately compared to each other. The same visualizations were run for each of the Doc2Vec experiments in order to visualize the similarity of documents among Doc2Vec experiments.

Results

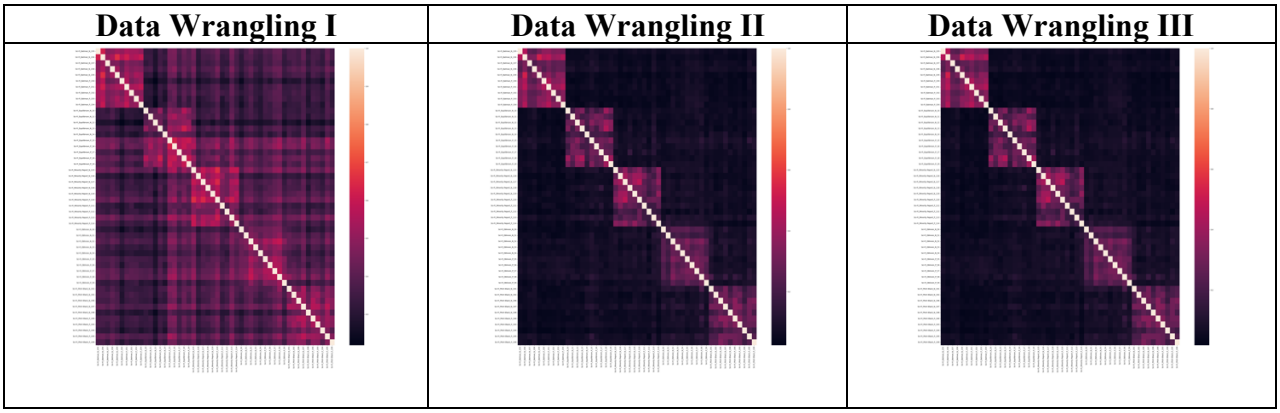
Preliminary Qualitative Results and Identified Prevalent Terms

Based on manual qualitative term extraction, four terms were identified for addition to the class corpus. The first term that should be considered is *"John Preston"*. This term appeared in all documents except one and appeared at least several times in almost all documents. This term also had a median TF-IDF score of 0.0287, the highest out of all terms among documents for the movie. The second term "*Grammaton cleric*" was seen several times in some form across all documents. It had a median TF-IDF score of 0.0209. The third recommended term is the name of the movie itself, "*Equilibrium*", present among all documents but one and mentioned several times in those documents. It had a median TF-IDF score of 0.02. While the term "movie" was used across 8/10 documents, it is most likely not a useful term to distinguish "*Equilibrium"* documents from other movie review documents. Thus, this term is not recommended for the class corpus, despite its median TF-IDF of 0.0185. The fourth term that should be added to the class corpus is actor "*Christian Bale".* This term appears in all but one of the documents and is mentioned several times in most of the documents. It scored a median TF-IDF of 0.0182. The term "*emotion*" appeared in every single document, and often appeared more than once, up to thirteen times in one document. It had a medium TF-IDF of 0.0122.

TF-IDF Vectors

After examining the results, it is clear that Data Wrangling methods II and III yielded superior results. As can be seen in *Figure 1,* the cosine similarity among documents discussing the same movie is far brighter (closer) than among documents discussing different movies within the genre. While this can be seen only faintly in Data Wrangling I, the contrast is far greater in Data Wrangling II and III indicating that the latter two data wrangling methods perform significantly better at distinguishing document reviews discussing one movie in the "sci-fi" genre from other movies.

Figure 1. TF-IDF Cosine Similarity Heatmap Between Documents



Word2Vec

The same data wrangling pattern can be seen for the Word2Vec embeddings. *Figure 2* displays the colors in the Word2Vec similarity matrices (which start out light, representing similarity) and become darker as data wrangling sophistication increased. The exception is the bottom right corner, which contains the custom prevalent words identified. As the rest of the plot becomes darker, the bottom corner retains its light shade. This demonstrates that data wrangling methods II and III were effective at learning the similarity of the prevalent document terms while learning to discern other terms from it. *Figure 3* demonstrates this as well – the size of the blue area (representing tokens that are hierarchically closer together) shrinks and any red (representing tokens that are further apart from one another) grows as the experiments moved from data wrangling method I through III. This also demonstrated that the sophisticated data wrangling methods perform favorably at clustering tokens similar tokens together and separating from dissimilar tokens. This is desired – the goal of clustering is to create distance between items of interest and other items to make the clusters more distinguishable. In the outputs generated, the tokens that appeared closest together included many of the terms identified in the qualitative term review and terms with high TF-IDF values, validating their importance.

Figure 2. Word2Vec Similarity Matrices (with Embedding Size 300)

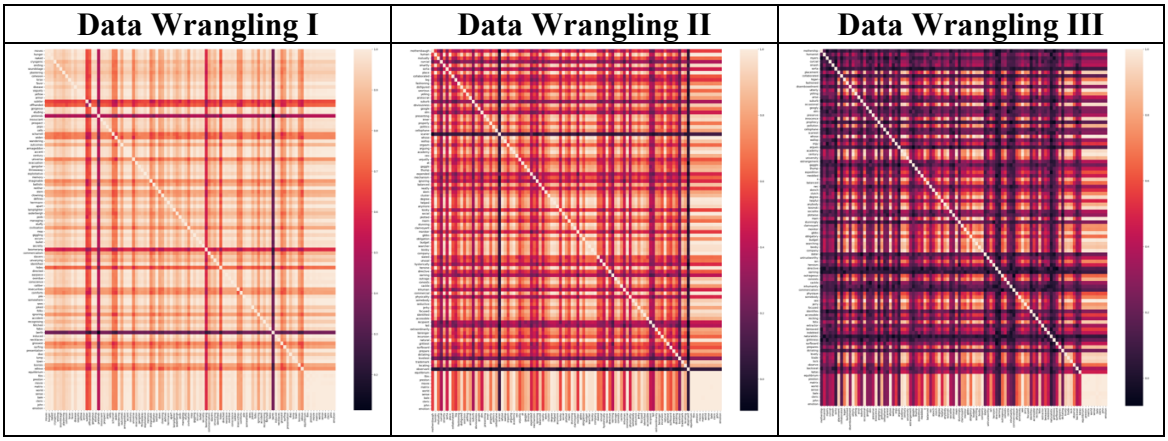
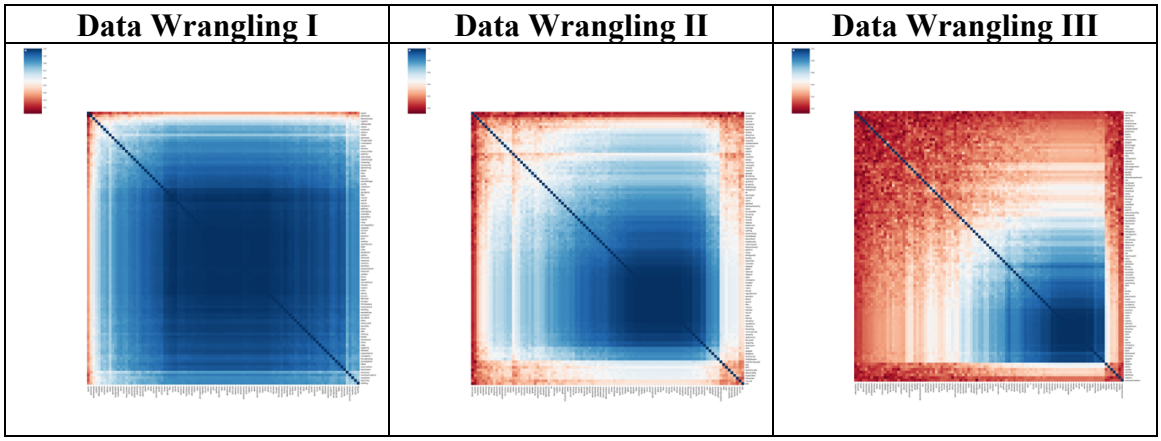


Figure 3. Word2Vec Similarity Cluster Maps (with Embedding Size 300)



The role of embedding sizes seemed to have a rather limited impact on performance compared to the data wrangling methods. The improvement of increasing embedding sizes from 100-300 seemed very negligible. *Figure 4* presents the most visible improvement that an increased embedding size provides for Word2Vec models – the increasingly darker colors for the higher embedding sizes suggest that an increased embedding size may have provided slight benefits in capturing a few more distinguishable features. The increasingly deeper red in *Figure 5* as the embedding size increases indicates that increased embedding sizes help in providing more information for finding differences between tokens.

Figure 4. Word2Vec Similarity Matrices (with Data Wrangling Method III)

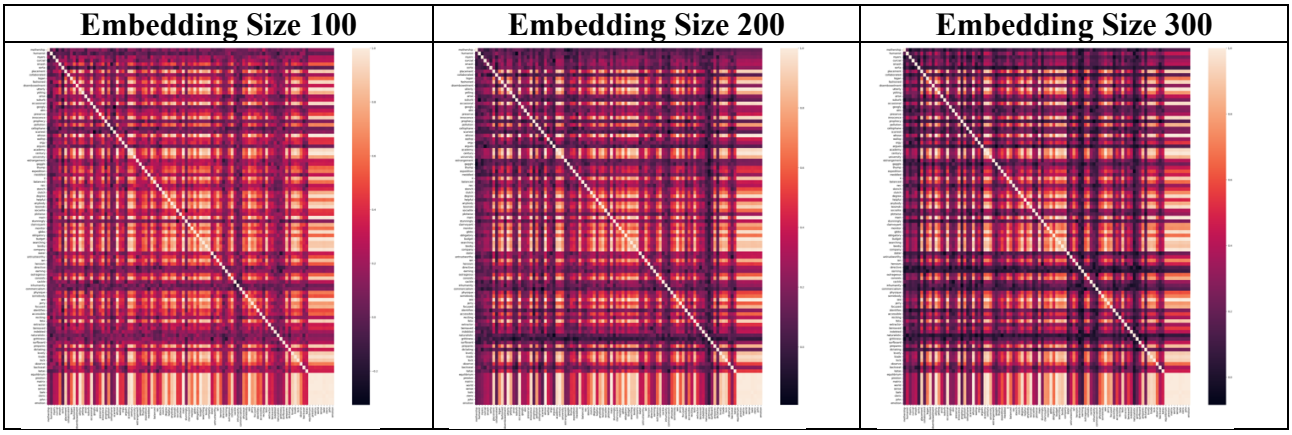
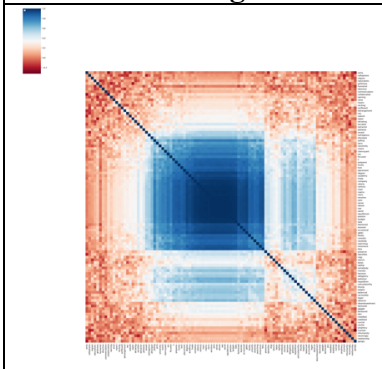
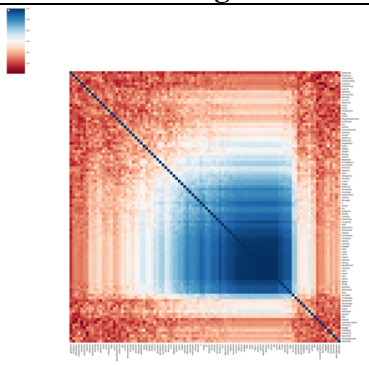
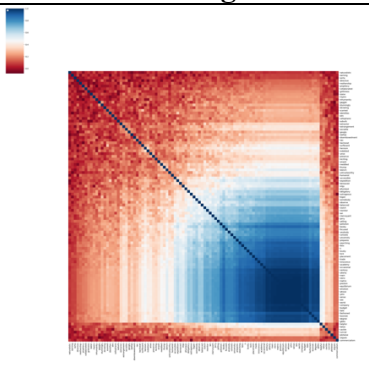


Figure 5. Word2Vec Similarity Cluster Maps (with Embedding Size 300)

Embedding Size 100	Embedding Size 200	Embedding Size 300
		

T-SNE Plots

The t-SNE plots for the Word2Vec models are difficult to interpret, since they are the reduction of highly dimensional outputs. The shapes are difficult to interpret and varied from experiment to experiment in ways that are not easily explainable. However, a long tail (as seen in *Figure 6*) with the prevalent "*Equilibrium*" terms grouped together was widespread among all experiments, suggesting that all methods were able to group these similar words together in some way.

Figure 6. t-SNE Plot of Word2Vec Tokens (Data Wrangling III, Embedding Size 300)

Doc2Vec

The Doc2Vec models performed very poorly at clustering together documents of the same movie as well as documents of the same genre. By looking in the Appendix at the supporting visualizations in *A5*, *A6*, one can see that there aren't clearly defined clusters in the same way as with Word2Vec. Ironically, the documents look less distinguishable with more sophisticated data wrangling steps and larger embedding sizes. The t-SNE plots for the Doc2Vec experiments in *A7* do not seem to yield many useful clusters as the Word2Vec models did.

Analysis

The TF-IDF vectors were able to successfully cluster together movie reviews belonging to the same movie within the "sci-fi" genre. The performance benefitted most directly from the application of lemmatization and removal of stop words, enabling the sparse vectors to remove noise and condense their dimensionality.

The Word2Vec model seemed to perform well when clustering together similar words. In particular, it seemed to successfully cluster the prevalent terms identified in the qualitative review. The Word2Vec model's performance was significantly enhanced by more sophisticated data wrangling steps. The biggest jump in performance appeared to occur in data wrangling method II when lemmatization was applied and stop words were removed, likely reducing the noise in the data. The removal of non-alphabetic tokens and custom stop words common among all movie reviews further reduced noise in data wrangling method III, allowing for more distinct clusters to be formed. The increased embedding size was helpful in providing more features to use during clustering, but only marginally improved performance compared to the more robust data wrangling techniques.

The Word2Vec model also seemed to align well with the prevalent terms defined in the qualitative term extraction of "*Equilibrium"*. The visualizations indeed showed qualitatively identified terms such as "equilibrium", "christian", "bale", "john", "preston", "cleric', "emotion", and "sense" as close to each other, affirming their importance for clustering and identifying document reviews of "*Equilibrium"*.

The Doc2Vec model performed poorly compared to the Word2Vec model. Further experimentation with larger datasets may yield better clusters for Doc2Vec, as Doc2Vec may require larger amounts of data. Regardless, it is clear that Doc2Vec struggled to cluster similar documents together. More experiments should be conducted to improve the Doc2Vec embeddings for clustering tasks.

Conclusion

In summary, this report emphasizes the critical role of data preprocessing – particularly lemmatization and stop word removal – in improving the performance of document clustering techniques. While TF-IDF vectors and Word2Vec embeddings managed to create meaningful clusters, experiments suggested Doc2Vec may be less optimal for this specific task. The findings detailed in this report provide valuable insights into the importance of data exploration, data wrangling, and conducting many experiments when working with natural language processing tasks, particularly in the domain of document clustering. Further exploration and experimentation with larger datasets may yield even more refined results in the future.

References

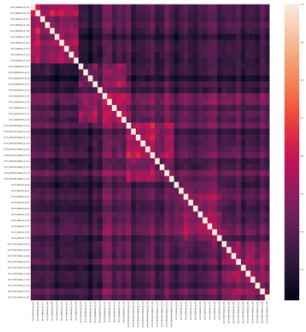
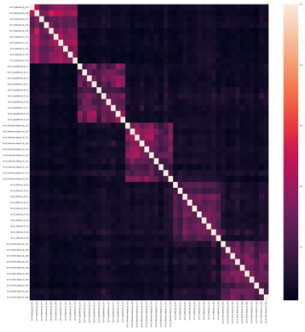
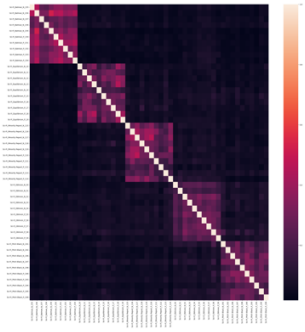
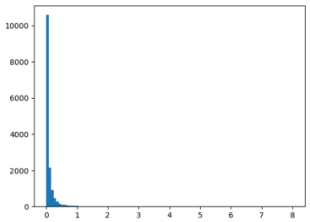
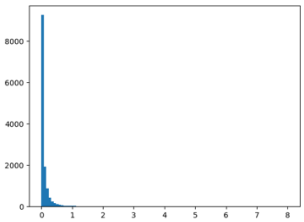
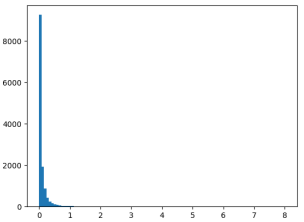
Řehůřek, R. (2022, December 21). *Gensim: Topic modelling for humans*. Doc2Vec Model - gensim. https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html

Řehůřek, R. (2022, December 21). *Gensim: Topic modelling for humans*. models.word2vec – Word2vec embeddings - gensim. https://radimrehurek.com/gensim/models/word2vec.html

Scikit-Learn Developers. (n.d.). *Sklearn.feature_extraction.text.TfidfVectorizer*. scikit. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Appendix

A1. Data Wrangling Comparative Analysis Plots

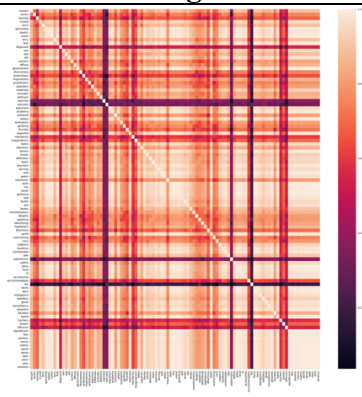
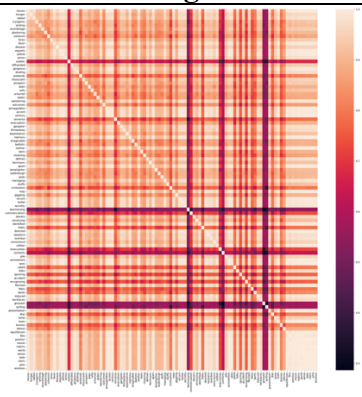
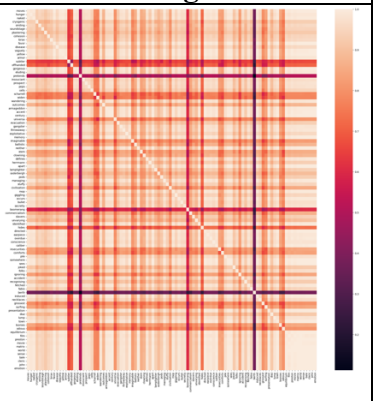
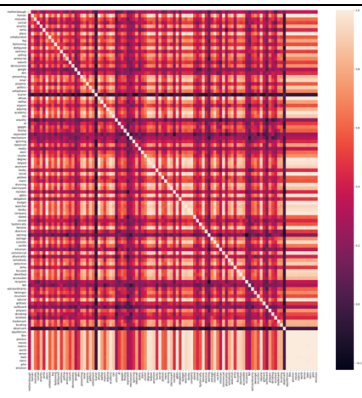
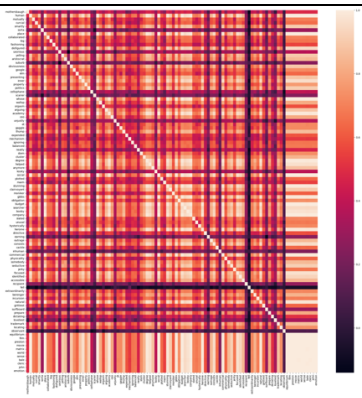
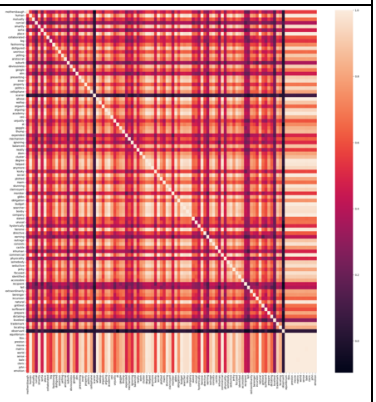
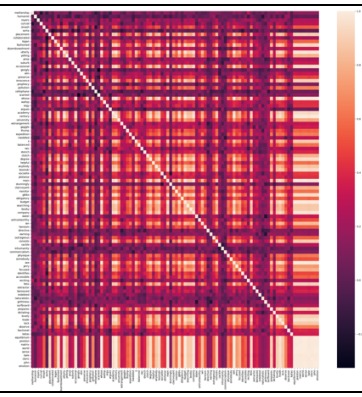
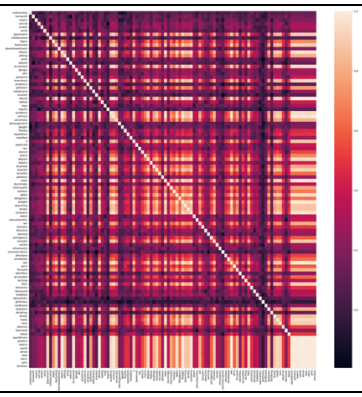
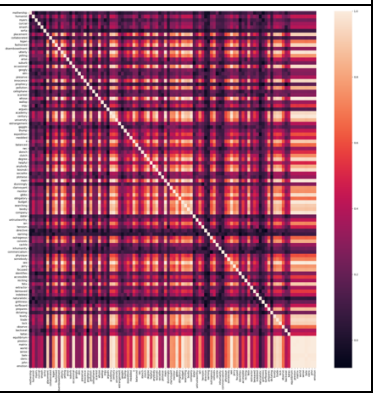
	Data Wrangling I	Data Wrangling II	Data Wrangling III
TF-IDF Embedding Heatmaps			
TF-IDF Histogram			
Top Terms By TF-IDF			

	Mean TF-IDF
the	45.72
and	22.96
of	22.11
to	21.32
in	14.13
is	13.35
that	10.95
it	10.93
as	7.50
with	7.37

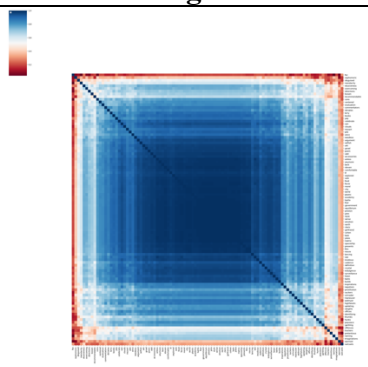
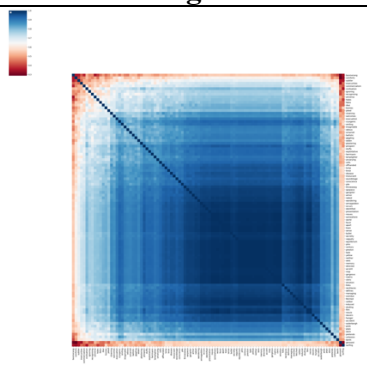
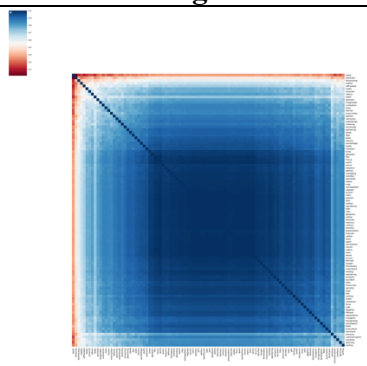
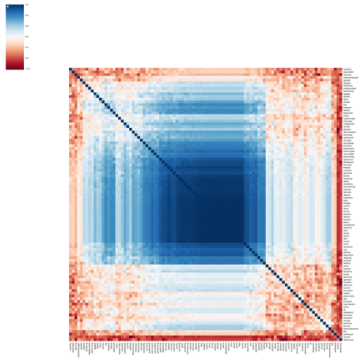
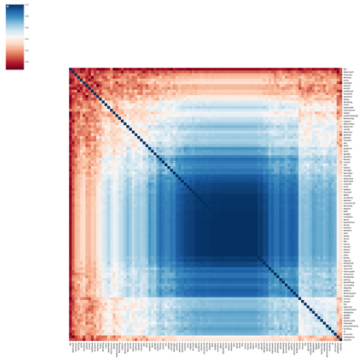
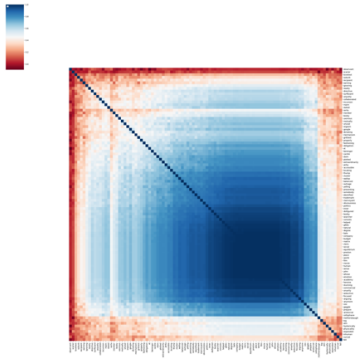
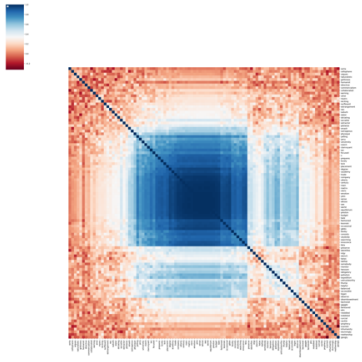
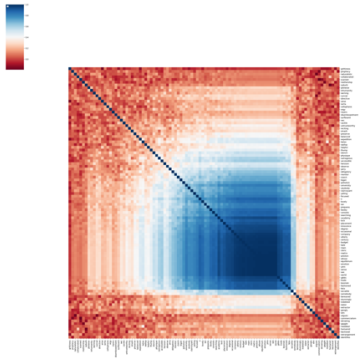
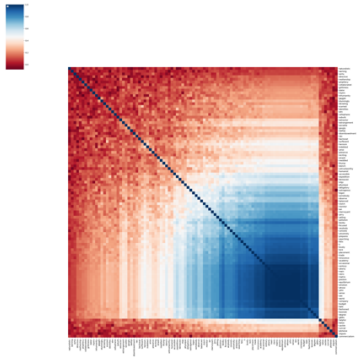
	Mean TF-IDF
film	5.72
movie	4.76
ha	3.59
wa	3.41
one	3.36
bond	3.14
like	2.89
time	2.62
get	2.31
character	2.26

	Mean TF-IDF
time	2.62
make	2.25
action	2.10
scene	2.06
even	2.02
elle	2.01
would	1.99
horror	1.87
batman	1.84
way	1.82

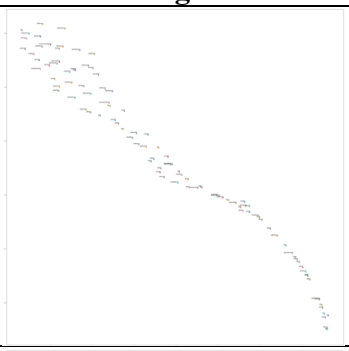
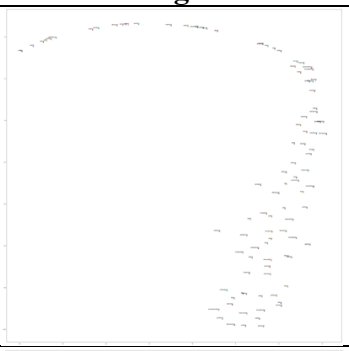
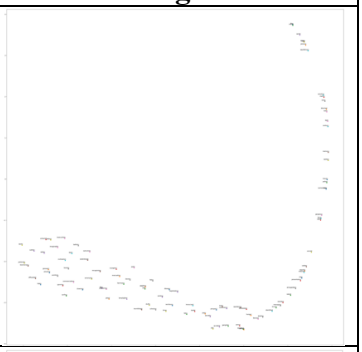
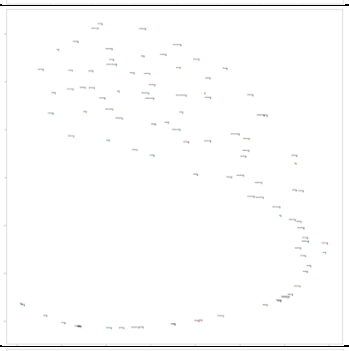
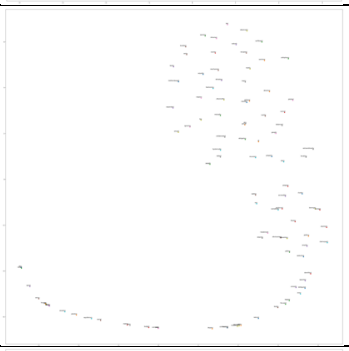
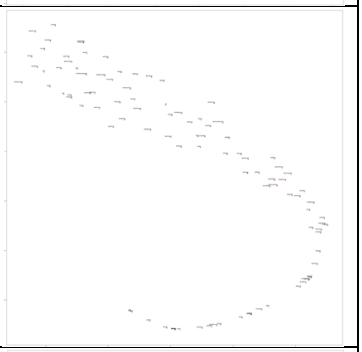
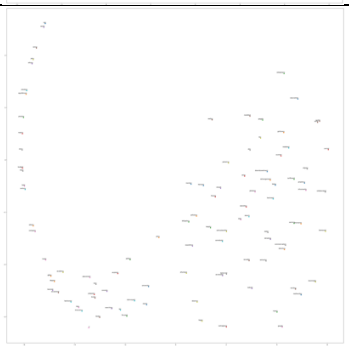
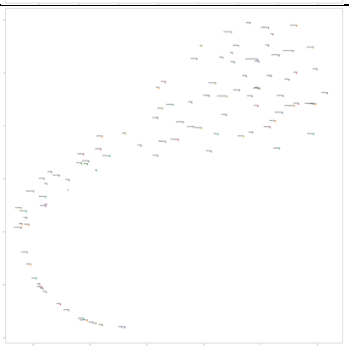
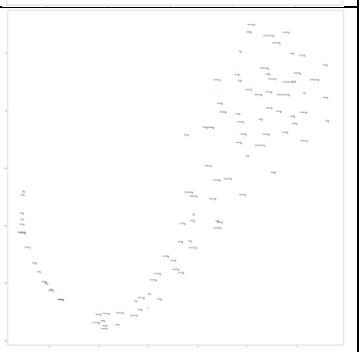
A2. Word2Vec Similarity Matrices

	Embedding Size 100	Embedding Size 200	Embedding Size 300
Data Wrangling I			
Data Wrangling II			
Data Wrangling III			

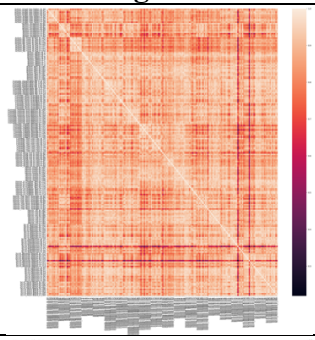
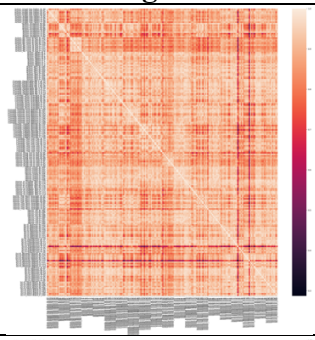
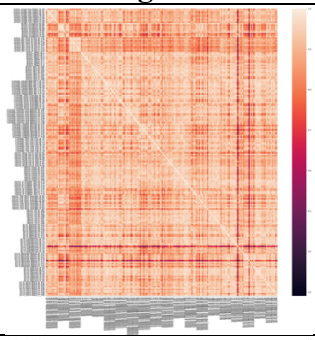
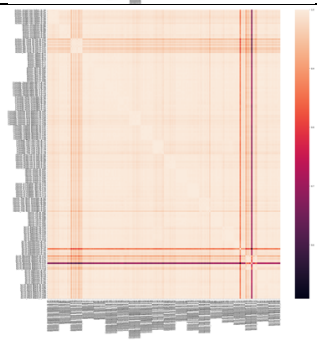
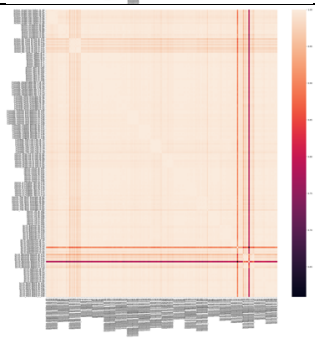
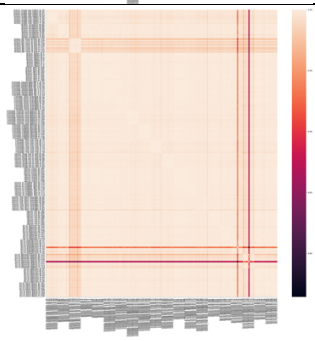
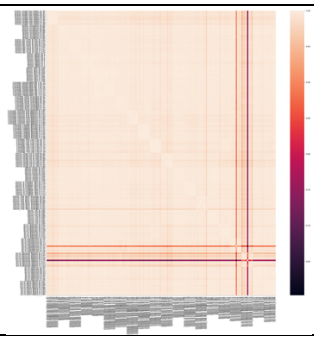
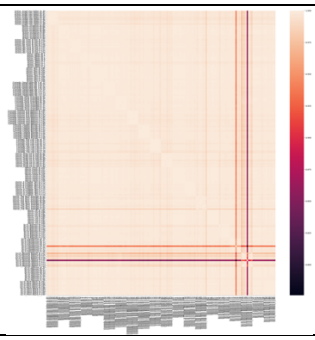
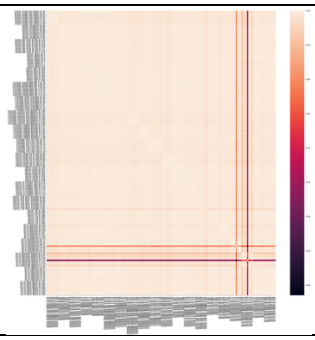
A3. Word2Vec Similarity Cluster Maps

	Embedding Size 100	Embedding Size 200	Embedding Size 300
Data Wrangling I			
Data Wrangling II			
Data Wrangling III			

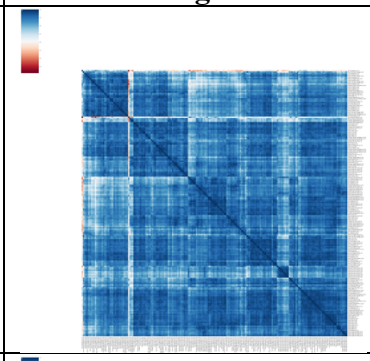
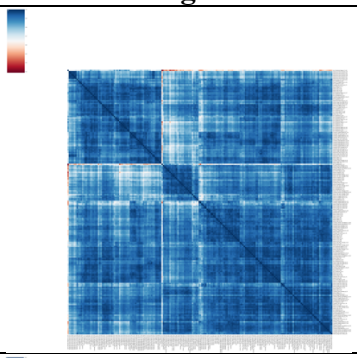
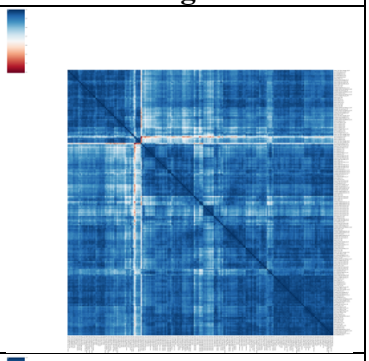
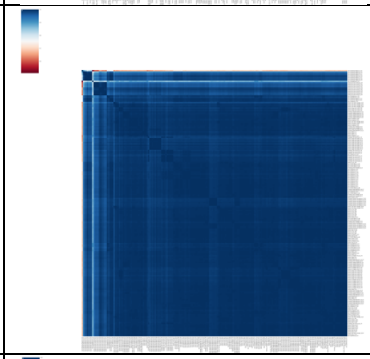
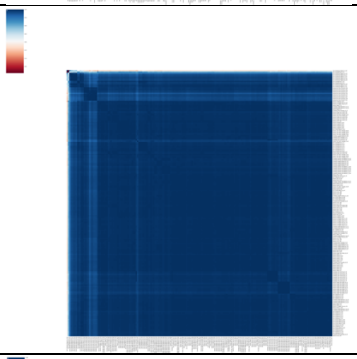
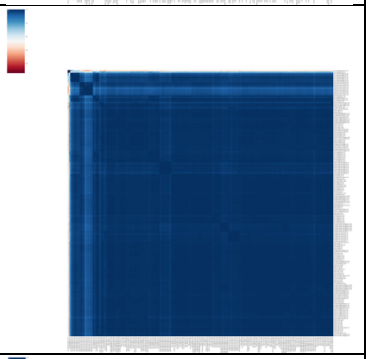
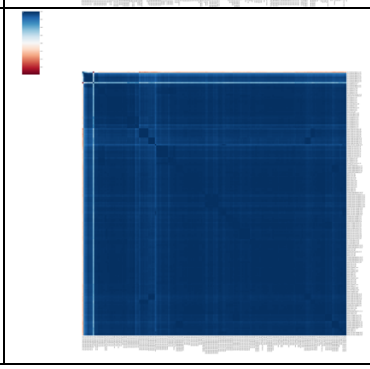
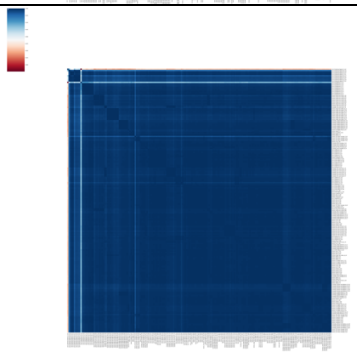
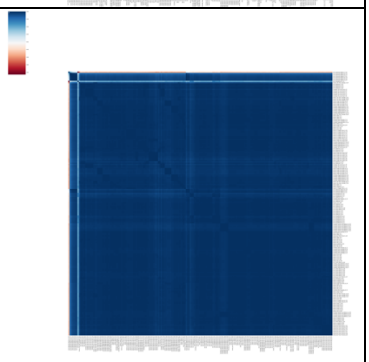
A4. Word2Vec t-SNE Plots

	Embedding Size 100	Embedding Size 200	Embedding Size 300
Data Wrangling I			
Data Wrangling II			
Data Wrangling III			

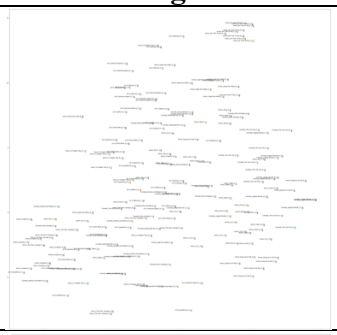
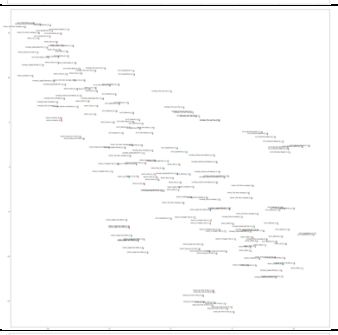
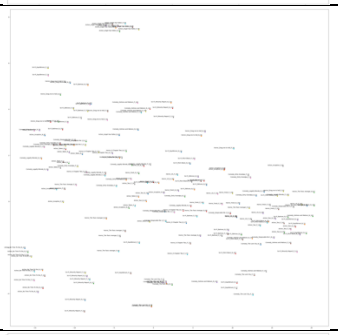
A5. Doc2Vec Similarity Matrices

	Embedding Size 100	Embedding Size 200	Embedding Size 300
Data Wrangling I			
Data Wrangling II			
Data Wrangling III			

A6. Doc2Vec Similarity Cluster Maps

	Embedding Size 100	Embedding Size 200	Embedding Size 300
Data Wrangling I			
Data Wrangling II			
Data Wrangling III			

A7. Doc2Vec t-SNE Plots

	Embedding Size 100	Embedding Size 200	Embedding Size 300
Data Wrangling I			
Data Wrangling II			
Data Wrangling III