Assignment 1: First Vectorized Representation

Kathryn LiPetri

MSDS 453: DL Section 56, Fall 2022

October 9, 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 data wrangling and vectorization. This vocabulary will eventually be used to represent the content of each movie review, so that they can later be classified and clustered. 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.

There are two steps for this assignment. The first step is a qualitative approach where each member of the class uses their own ten movie reviews. The goal of the first assignment is to manually identify some terms in the ten movie reviews that are proposed to be good candidates for the corpus vocabulary. The choice of terms will be based on the term's importance (in at least one document) and prevalence (in at least two or three documents).

The second step is a quantitative approach where the results of code using the entire class-corpus of movie reviews are evaluated. The techniques used for this step include data wrangling, document vectorization using tf-idf, document vectorization using Doc2Vec, and token vectorization using Word2Vec embedding. Multiple different experiments are run using these techniques to gather insight and information for the corpus vocabulary.

The individual movie assigned to me was "Angel Has Fallen", which is a film in the Action genre. Some candidate terms chosen qualitatively for the corpus vocabulary are action, audience, character, plot, attack, assassination, trust, agent, escape, guard, president, kill, military, secret, and protect. Some of these terms such as attack, assassination, and agent could be more applicable towards the film I was assigned, but other terms such as action, character,

agent, plot, president, and secret could apply more generally to the action films and entire class-corpus.

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)
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

A total of twenty-one different experiments were run as part of this assignment. These experiments included three different data wrangling methods with computation of tf-idf scores for each data wrangling method, accounting for three of the experiments. Nine different experiments were run using Word2Vec with the three data wrangling methods and three different embedding dimensions for each data wrangling method. In addition, nine different experiments were run using Doc2Vec with the three data wrangling methods and three different embedding dimensions for each data wrangling method. Table 2 summarizes the twenty-one experiments that were performed.

Table 2. Assignment 1 Experiments Summary

Data Wrangling Method	Experiment Number and Description
Method 1 (tokenization + normalization)	1. tf-idf 2. Word2Vec (embedding dimension = 100) 3. Word2Vec (embedding dimension = 200) 4. Word2Vec (embedding dimension = 300) 5. Doc2Vec (embedding dimension = 100) 6. Doc2Vec (embedding dimension = 200) 7. Doc2Vec (embedding dimension = 300)
Method 2 (tokenization + normalization + lemmatization + stopwords)	8. tf-idf 9. Word2Vec (embedding dimension = 100) 10. Word2Vec (embedding dimension = 200) 11. Word2Vec (embedding dimension = 300) 12. Doc2Vec (embedding dimension = 100) 13. Doc2Vec (embedding dimension = 200) 14. Doc2Vec (embedding dimension = 300)
Method 3 (tokenization + normalization + lemmatization + stopwords + stemming)	15. tf-idf 16. Word2Vec (embedding dimension = 100) 17. Word2Vec (embedding dimension = 200) 18. Word2Vec (embedding dimension = 300) 19. Doc2Vec (embedding dimension = 100) 20. Doc2Vec (embedding dimension = 200) 21. Doc2Vec (embedding dimension = 300)

Results

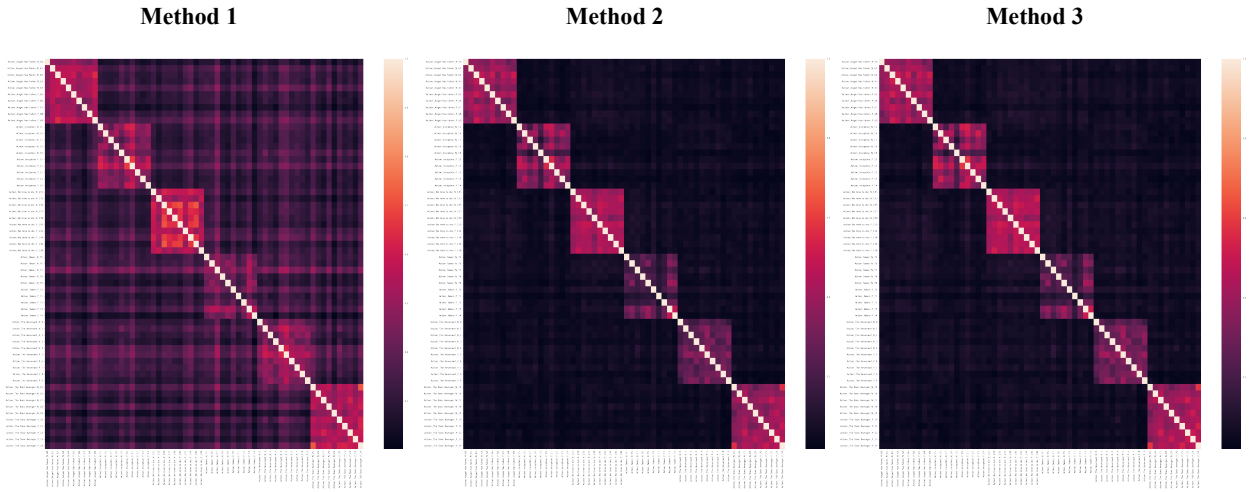
Focusing on the movie reviews in the action genre, three different tf-idf experiments were run using three different methods of data wrangling, as summarized in Table 3. In Table 3 the top ten token words determined from each data wrangling method are also summarized along with their tf-idf scores.

Table 3. Tf-idf Summary for Data Wrangling Methods 1, 2, and 3

Method 1 tokenization + normalization		**Method 2 tokenization + normalization + lemmatization + stopwords**		**Method 3 tokenization + normalization + lemmatization + stopwords + stemming**	
Vocabulary Size = 15,999		**Vocabulary Size = 14,193**		**Vocabulary Size = 10,491**	
Token	**Tf-idf Score**	**Token**	**Tf-idf Score**	**Token**	**Tf-idf Score**
the	47.40	film	5.66	film	5.72
and	24.22	movie	4.48	movi	4.48
of	23.38	ha	4.10	ha	4.10
to	21.47	one	3.72	one	3.72
in	15.14	batman	3.39	batman	3.39
is	12.91	like	3.24	like	3.34
that	11.14	wa	3.03	wa	3.03
it	9.95	time	2.58	time	2.65
as	8.23	character	2.57	charact	2.57
with	7.65	bond	2.53	bond	2.48

Heatmaps were created to plot the movie review documents, for the action movies, along the x- and y-axes to visualize cosine similarity in document vocabulary. One heatmap was created for each data wrangling method. Areas lighter in color show the intersection of two documents with very similar vocabulary. Areas in darker color show the intersection of two documents with very different vocabulary.

Figure 1. Heatmaps, Action Movie Reviews



In addition to tf-idf experiments, nine different Word2Vec experiments were run using a one-hundred tokens, including a random subset of terms and five chosen prevalent terms, and t-SNE and heatmap visualizations were created for each experiment. One experiment was performed for each data wrangling method (Method 1, 2, and 3) and three different embedding dimensions (100, 200, and 300). Figures 2 and 3 show examples of a Word2Vec t-SNE plot and a heatmap, respectively.

Figure 2. Word2Vec t-SNE, Method 2, Embedding Dimension 200

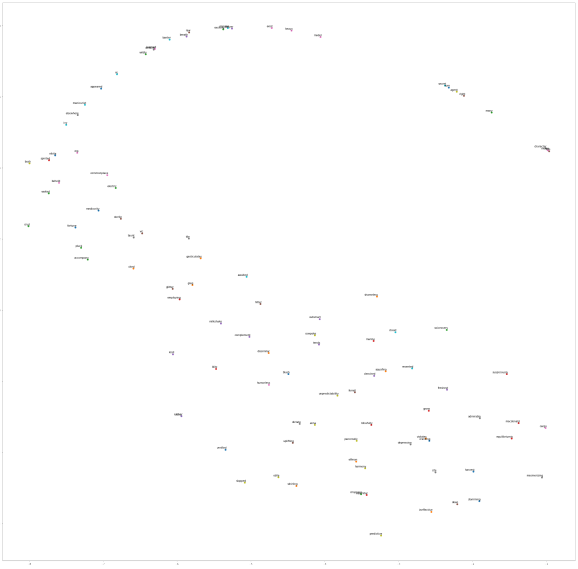
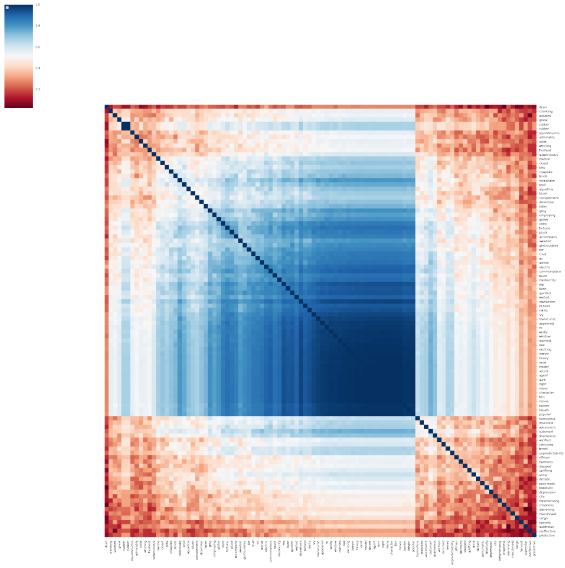


Figure 3. Word2Vec Heatmap, Method 2, Embedding Dimension 200



Lastly, nine different Doc2Vec experiments were run using all the documents in the class-corpus, and t-SNE and heatmap visualizations were created for each experiment. One experiment was performed for each data wrangling method (Method 1, 2, and 3) and three different embedding dimensions (100, 200, and 300). Figures 4 and 5 show examples of a Doc2Vec t-SNE plot and a heatmap, respectively.

Figure 4. Doc2Vec t-SNE, Method 2, Embedding Dimension 200

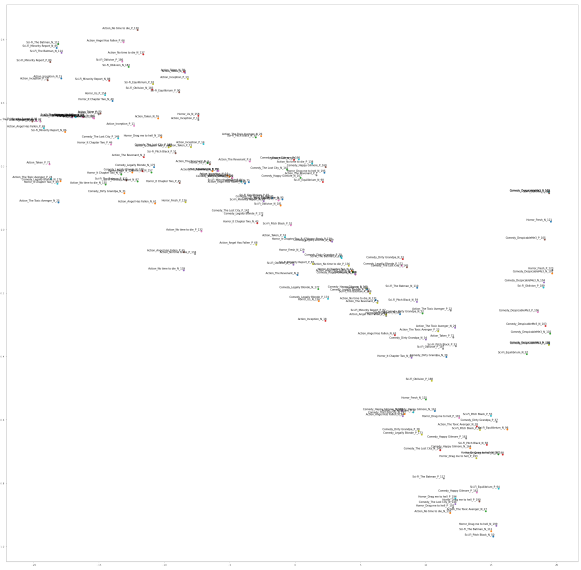
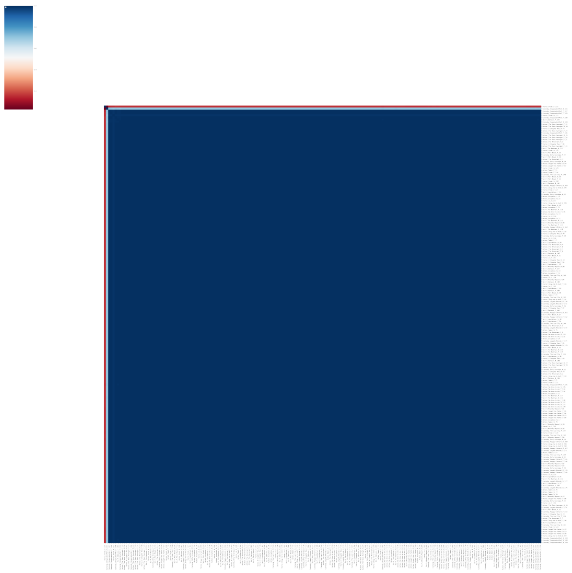


Figure 5. Doc2Vec Heatmap, Method 2, Embedding Dimension 200



Any visualizations not presented in the main body of the report can be found in the Appendices section of this report.

Analysis and Interpretation

As were summarized in Table 3, the top ten words and their tf-idf scores were determined for three different data wrangling methods. Using only tokenization and normalization, the tokens determined from Method 1 are all common stopwords that will likely not be helpful as part of a corpus vocabulary. Using Method 2 of data wrangling, with the addition of lemmatization and stopword removal, the tokens are much more descriptive. “Film” and “movie” are at the top of the list with the highest tf-idf scores, and some words that belong to specific movies are on the list, such as “batman” and “bond”. The word “character”, that was from my list of prevalent terms, made it onto the list. Some words such as “ha” and “wa” were also on the top ten token list for Method 2, but these words seem less descriptive than others. Method 3 added stemming to the data wrangling methods, and greatly reduced the vocabulary size to 10,491 words. All of the tokens from Method 3 were the same as for Method 2, but some of the

words were cut short, such as “movi” instead of “movie”. The tf-idf scores for the words from Method 2 and Method 3 were also very similar. Based on the results, Method 2 seems to have produced the best token words because stemming produced many words that are not actual words. Looking at the top thirty tokens using data wrangling Method 2, of the chosen prevalent terms, “character” was ninth on the list with a tf-idf score of 2.57, and “action” was seventeenth on the list with a tf-idf score of 2.00. Any other chosen prevalent terms were not in the top thirty tokens.

Based on the tf-idf results of the top thirty tokens using data wrangling Method 2, my suggested list of words for the class-corpus vocabulary is summarized in Table 4.

Table 4. Suggested Class-Corpus Vocabulary

Token	Tf-idf Score
one	3.72
batman	3.39
like	3.24
time	2.58
character	2.57
bond	2.53
toxic	2.44
even	2.20
make	2.05
action	2.00

Based on the tf-idf heatmap visualizations shown in Figure 1, data wrangling Method 1 is not desirable because there is too much similarity between the movie review documents. The comparison between data wrangling Method 1 and Methods 2 and 3 show the importance of removing stopwords from the data. Data wrangling method 2 and 3 are comparable and show that movie review documents from the same movie have similar vocabulary to each other, however, there is not much vocabulary overlap between movies of different titles within the

same genre. There is not much similarity in vocabulary between different movies, and even some of the reviews of the same movie have more sparse overlap, such as for the movies “Taken” and “The Revenant”. Movie reviews for “No Time to Die” seem to have the most similarity or overlap.

Figure 6. Word2Vec t-SNE Plots

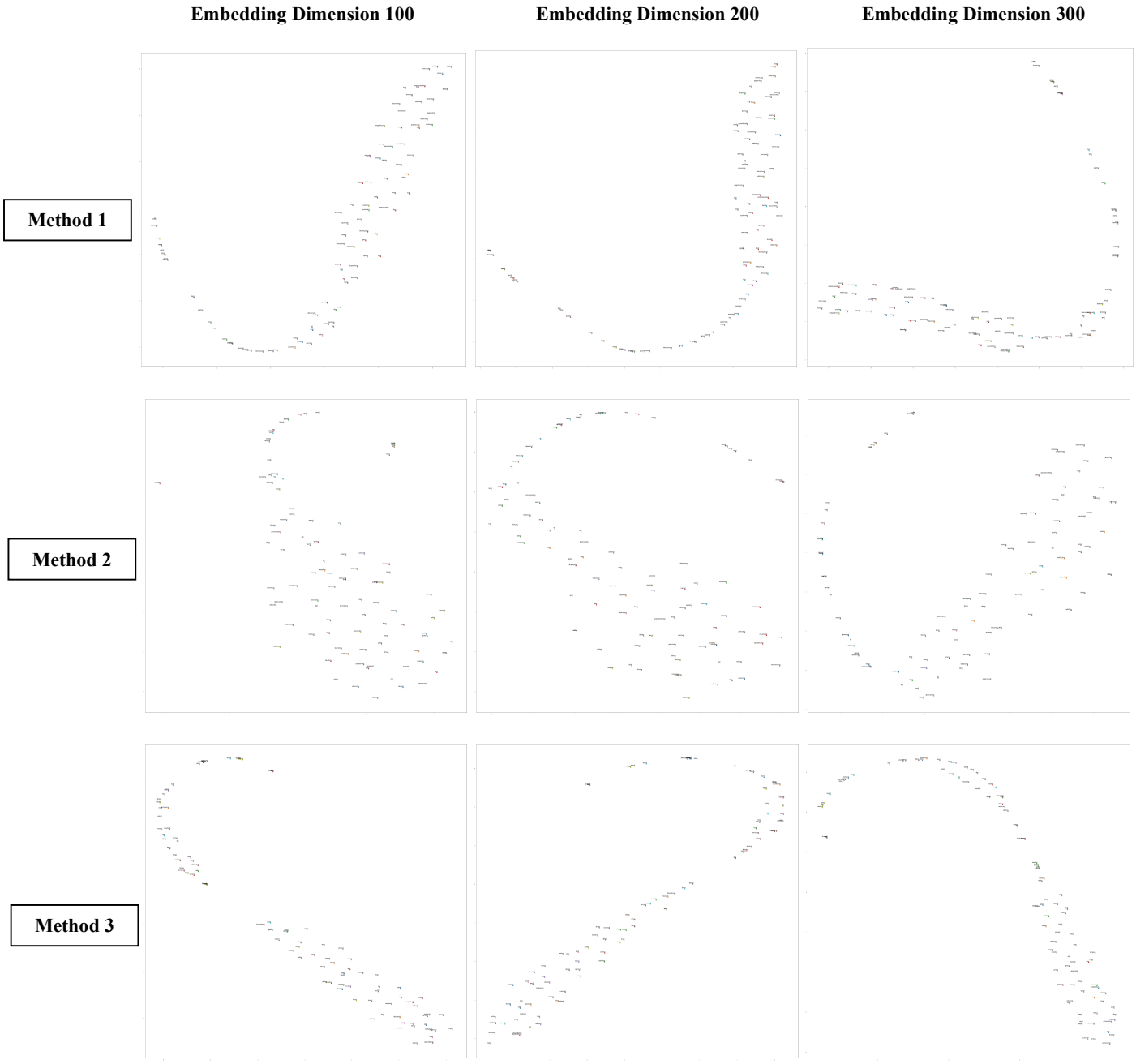


Figure 6 shows a summary of the t-SNE plots from the Word2Vec experiments. Within each data wrangling method, the shape of the t-SNE plots remains relatively the same. The change happens when different embedding dimensions are used. The shape rotates or becomes a mirror image of itself going from left to right. The shapes for Method 1 and Method 3 look like each other – a larger cluster of words with a long, thin tail of words. The shapes for Method 2 have a more spread-out cluster of words with a tail, as well. Since there are not a lot of data in the class-corpus movie reviews, it is difficult to see a clear pattern or clustering in the t-SNE charts.

Figure 7. Doc2Vec t-SNE Plots

Figure 7 shows a summary of the t-SNE plots from the Doc2Vec experiments. With data wrangling Method 1, there is no real shape to the plot because there are fewer words filtered out from the documents. For data wrangling Method 2, there appear to be two distinct clusters of words present at each embedding dimension. Each plot has one larger cluster of words next to one smaller cluster of words. Each of the smaller clusters seems to contain mostly documents related to “Despicable Me 3”, while the larger clusters contain most other movies. This implies that lemmatization and removal of stop words helped to separate documents into at least two groups. With the addition of stemming for Method 3, clusters are not as apparent as they are for the Method 2 charts. It appears that Doc2Vec visualizations show, as the tf-idf experiments showed, that Method 2 works best for our class-corpus movie reviews.

Heatmaps for Word2Vec and Doc2Vec experiments are included in the Appendices. All the Doc2Vec heatmaps look very similar, and it is difficult to draw any conclusions on which documents might have similar vectors. For the Word2Vec heatmaps, Method 2 heatmaps showed the most variation between words, and some words such as “mediocrity” and “commonplace” were found with similar word vectors for the embedding dimension 200.

Conclusions

The terms proposed that would be best for the corpus vocabulary, based on the results from the different tf-idf experiments, are: one, batman, like, time, character, bond, toxic, even,

make, and action. Based on the results of the experiments, data wrangling Method 2 seems to be the most effective for our class-corpus. The list of token words for tf-idf was improved from Method 1 and did not change much for Method 3 (only that some of the words ended up being truncated into non-English words). This was also observed in the td-idf heatmaps. Cosine similarity improved for the Method 2 visualization and stayed the same for the Method 3 visualization.

For the Word2Vec heatmaps, Method 2 data wrangling with embedding dimension of 200 gave the results that made the most sense. The Word2Vec t-SNE plots were not conclusive because the shapes did not change much with different embedding dimensions, and clusters or groups in the words were not apparent.

For the Doc2Vec heatmaps, nothing conclusive was determined. The heatmaps looked the same for the different data wrangling methods and embedding dimensions. The Doc2Vec t-SNE plots did give some insights, especially for Method 2 data wrangling where visible clusters were apparent. Method 2 and embedding dimension 300 seemed to give the best results for Doc2Vec t-SNE plots.

Method 2 seems to be the best data wrangling method overall, compared to Method 1 and 3. Embedding dimensions of 200 or 300 work well, depending on the type of experiment and visualization.

Appendices

Figure A1. Word2Vec Heatmaps

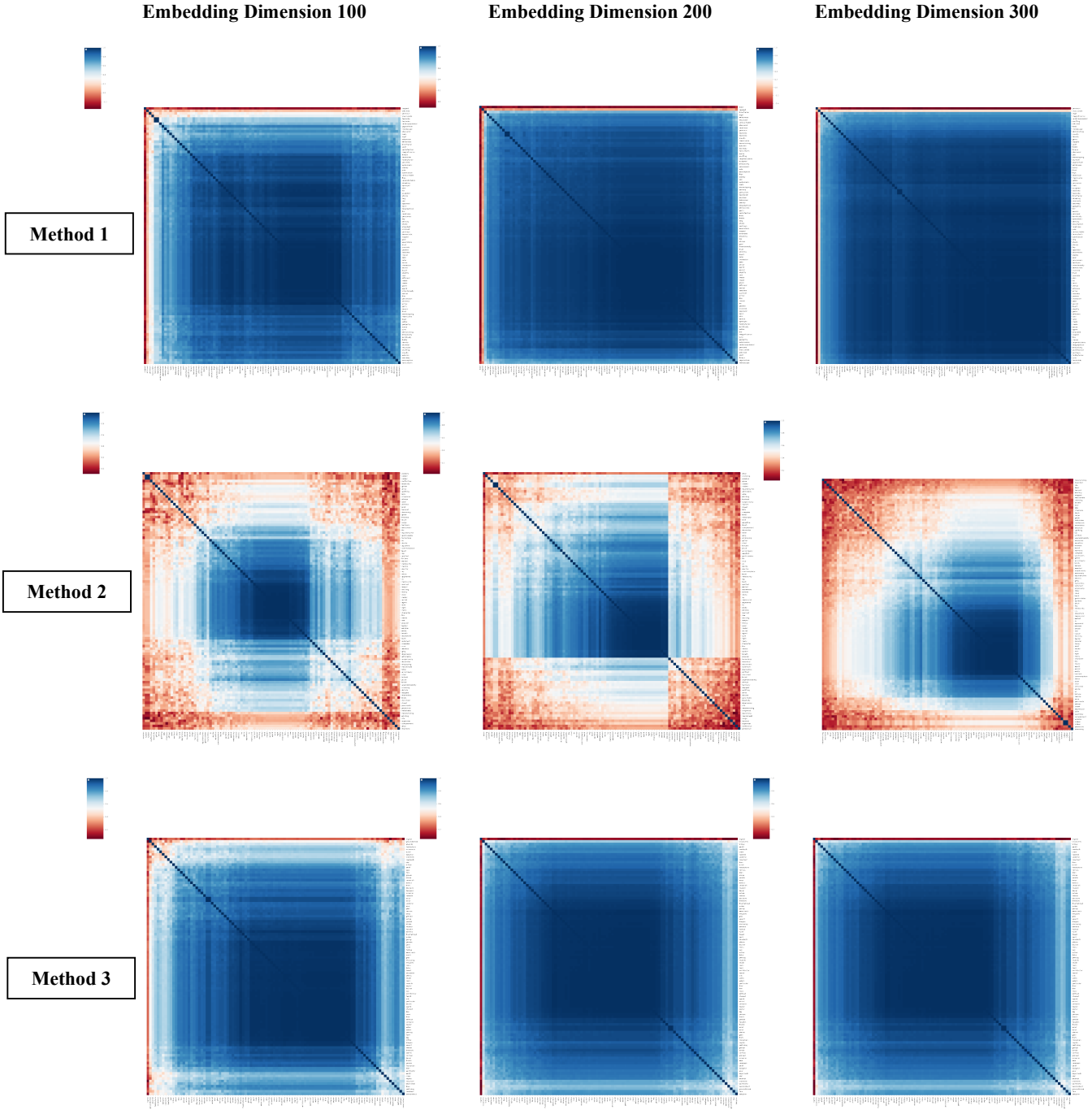


Figure A2. Doc2Vec Heatmaps