Final Report

Amber Lee Curran (akc6be@virginia.edu)

DS 5001

17 December 2021

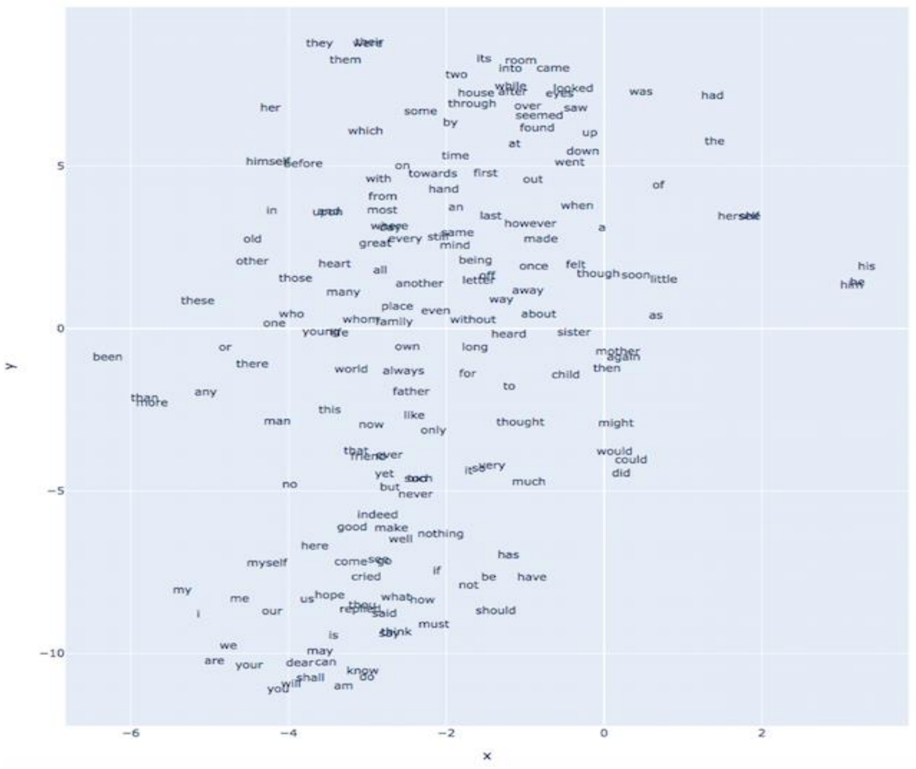
In this project, it was my goal to explore the most popular and well-known literature downloaded off Project Gutenberg to analyze any resemblance or difference in text structure, terms, and sentiment. Specifically, I looked at the top 5 pieces of literature downloaded in the last 30 days as of today. These novels included in order: *Frankenstein*, *A Christmas Carol*, *Pride and Prejudice*, *The Scarlet Letter*, and *Alice's Adventures in Wonderland*. These novels are known as classics in English literature, and therefore I wanted to explore these texts to understand perhaps why they are highly downloaded and considered classics in relation to one another.

After downloading the top 5 novels in plain text as the F0 source format, a Jupyter notebook in Python was used to extract only the part of the Project Gutenberg text file to include the words contained in the novel itself with no additional markups. The F1 was created including each line of each book. Next, F2 tables were created to extract the initial LIB, DOC, TOKEN, and VOCAB to be used for further analysis. The LIB table includes each book id, title, author, and file name. The DOC table includes the book id, chapter number, and separates each paragraph in a cell. The TOKEN table parses the text down to each word and uses NLTK to determine the part of speech of each token. The VOCAB table includes each term in the corpus and the number of times the term appears in the corpus.

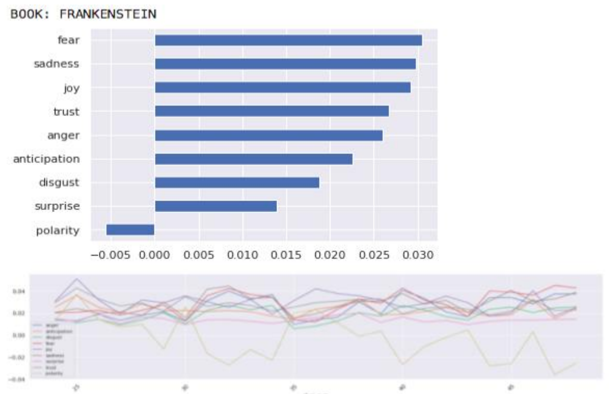
The F3 was created by annotating the VOCAB table including stopwords, stems, and the maximum parts of speech expected for the word from NLTK. The F4 was created by using the TOKEN table to create a bag of words table. This then allowed for the creation of a document term count matrix and the creation of a Term Frequency Inverse Document Frequency (TFIDF) matrix. The F5 was created to explore the principal components, topic models, word embeddings, sentiment analysis, and similarity and distance measures. The PCA model allowed for picking the top 10 K components, creating loadings, and creating the document component matrix. The LDA models created theta, phi, and topic tables to be analyzed and visualized by clustering topics. The word embeddings used a word2vec model to explore distances in vector space. Sentiment analysis looked at how the eight main emotions are portrayed across each model. And lastly, the similarity and distance measures were computed between novels by sampling and plotting how they compare to one another with a pair plot and dendrograms.

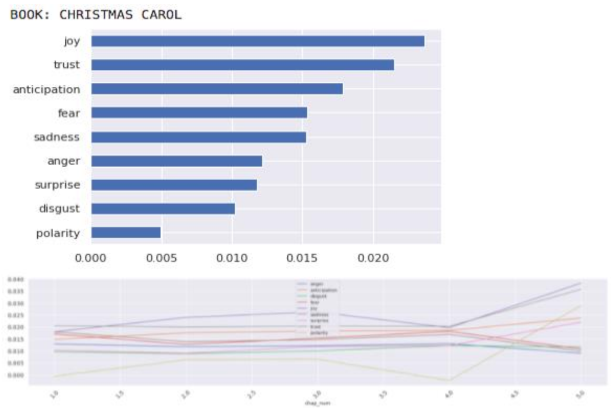
The F6 included many visualizations of the analysis done in the F5. The TFIDF heatmap shows the top 20 terms sorted by tfidf sum. Many of these terms were proper nouns, therefore I also created a heatmap of the top 20 terms sorted by tfidf sum without proper nouns which indicates the most popular words were mostly nouns.

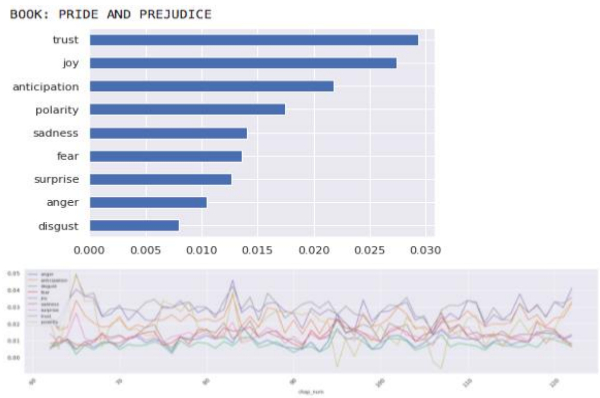
The t-SNE plot uses the word2vec model to create vector space representations of each word for which can be plotted in an x-y plane. In the plot, we see words that are highly associated with each other are closer in vector space than words that are less associated with each other.

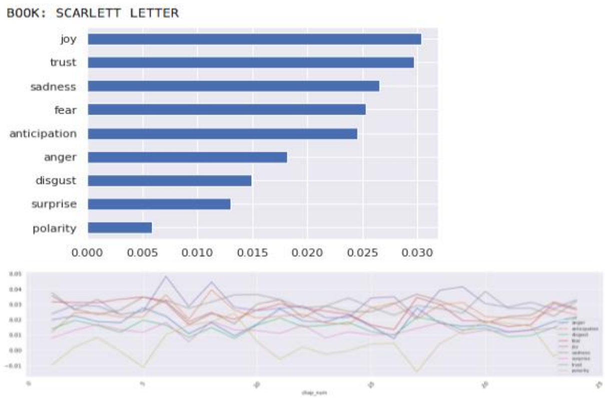


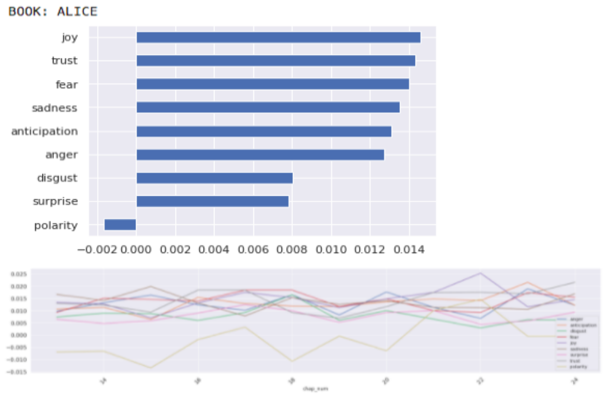
The sentiment analysis plots describe the eight emotions with the addition of polarity for each novel as a whole in the bar plot, and based on chapters throughout the novel in the line plot. As you can see, different novels indicate different emotions, and show different fluctuating or stagnant changes in emotion throughout the novels.


BOOK: FRANKENSTEIN

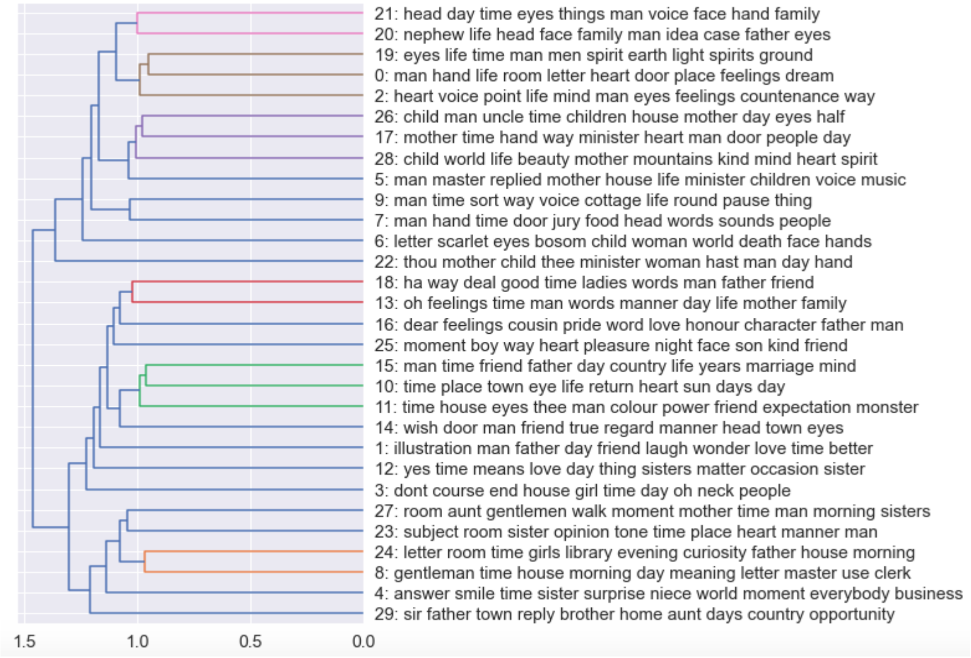

BOOK: CHRISTMAS CAROL



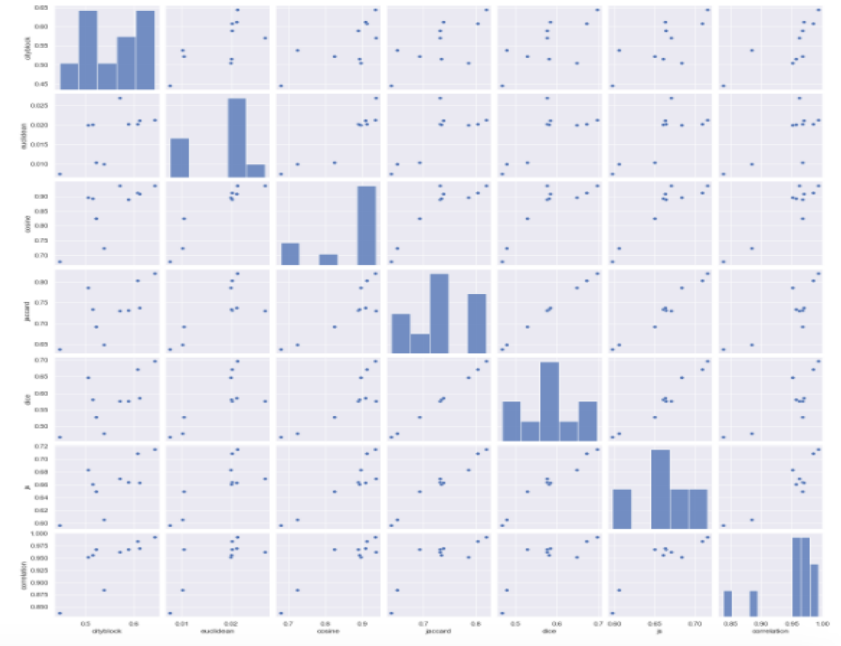




Topic models using LDA were also visualized in plot trees to show alike cluster topics with connected branches, close branches, and similar colored branches.

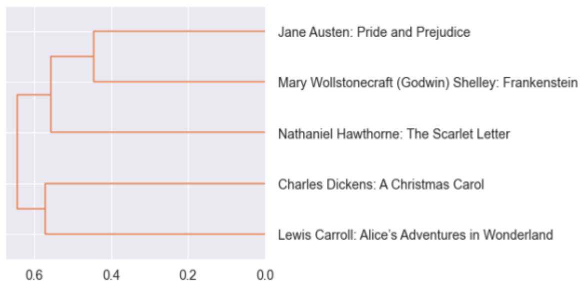


Similarity and distance measures between documents were visualized through scatterplots between the distance measures. The bar plots and scatter plots indicate some strong positive correlations while others do not indicate much correlation at all.

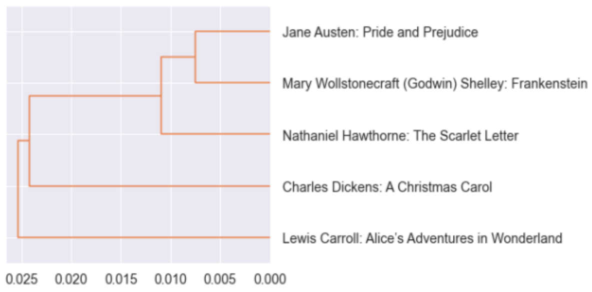


Lastly, dendrograms were also used to explore each distance measure and how the novels are seen to be related or less related to each other. In all dendrograms, it seems the distance measures are separating *A Christmas Carol* and *Alice's Adventures in Wonderland* together, and the other three novels together.

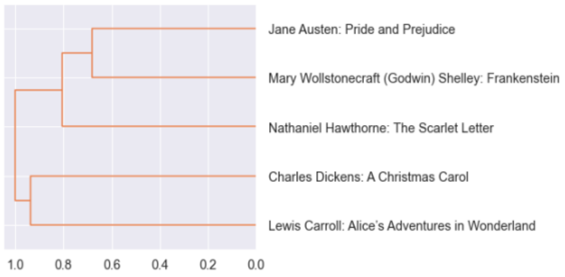
```
Distance: Cityblock
<Figure size 432x288 with 0 Axes>
```



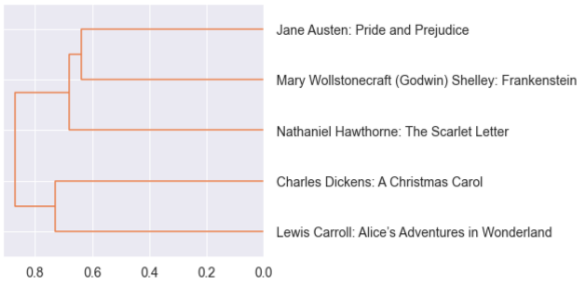
```
Distance: Euclidean
<Figure size 432x288 with 0 Axes>
```



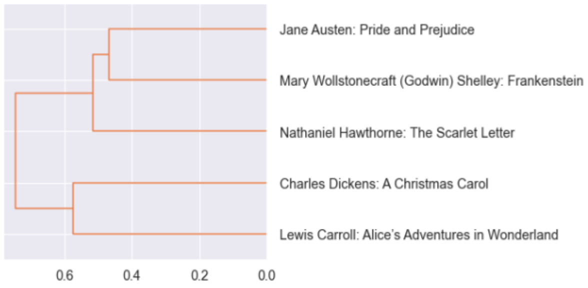
```
Distance: Cosine
<Figure size 432x288 with 0 Axes>
```



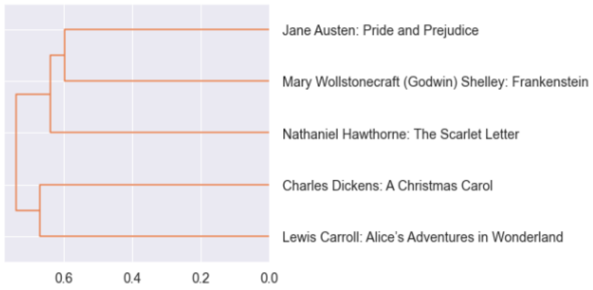
```
Distance: Jaccard
<Figure size 432x288 with 0 Axes>
```



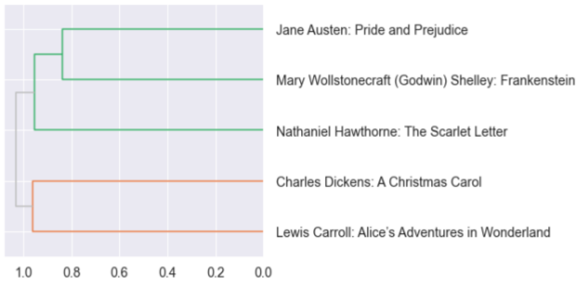
```
Distance: Dice
<Figure size 432x288 with 0 Axes>
```



```
Distance: JS
<Figure size 432x288 with 0 Axes>
```



```
Distance: Correlation
<Figure size 432x288 with 0 Axes>
```



Overall, the novels show some similarity and differences, although further analysis could be done to fully investigate why these novels are so highly downloaded. Future analysis would be interesting to see how different the novels are as you move down the top 100 list, or how a very different piece of literature (for example a scientific journal) would compare if included in the analysis. I would also be interested in how this list may change over time, for example A Christmas Carol is listed as second at the moment, however, at a different time of year it may not be as popular. A limitation with the code would be the length of time it took to run.

Originally, I set out to analyze the top 10 texts, however, with the same code used here, it took over an hour to run fully. To alleviate this problem, I should have split the code into separate notebooks that performed different tasks, or used caching on some of the outputs.