Project Overview

I used Project Gutenberg to download text files from the Communist Manifesto, the U.S. Constitution, Dictatorship vs. Democracy by Leon Trotsky, and Utilitarianism by John Stuart Mill. In order to analyze these texts, I used sentiment analysis in order to compare the positive, negative, and neutral sentiments of each text in a bar graph. Additionally, I found the most common words of each text that were longer than four letters, and created a list for each of the texts that had words that were not in the other texts.

Implementation

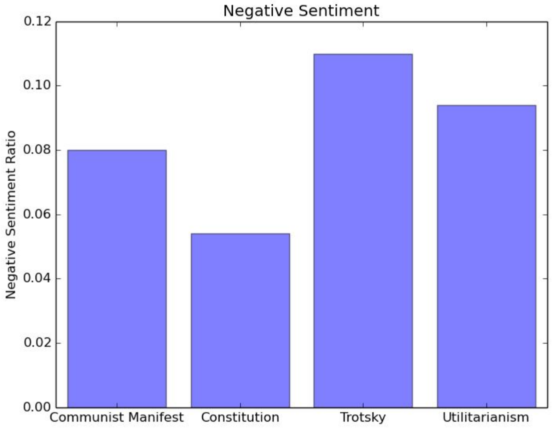
In order to determine the negative and positive sentiment in the texts I had to use a sentiment analyzer. Then, once I determined the ratio of positive, neutral, and negative sentiment in each of the texts, I was able to compare the texts according to the sentiments in three bar charts. Originally I had planned on making one scatter plot of the data. However, the plot was visually confusing, so in order to make the data more understandable, I chose to split the data up into three bar graphs.

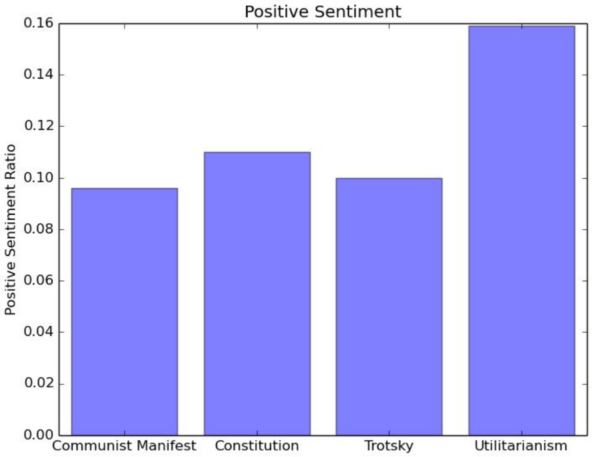
To further analyze the texts, I used the texts to create a dictionary of the words in the text and the amount of times they showed up within the text. In order to make the results more interesting and to eliminate more trivial words such as ‘a’, ‘or’, ‘and’, etc., I decided to create a list of tuples with the words being longer than four characters, and find the most common words for each text from this list. Originally I did not eliminate words that were less than four characters, because some words that are four characters or less long contribute to the analysis of the texts, but for the most part, I determined that they were unnecessary.

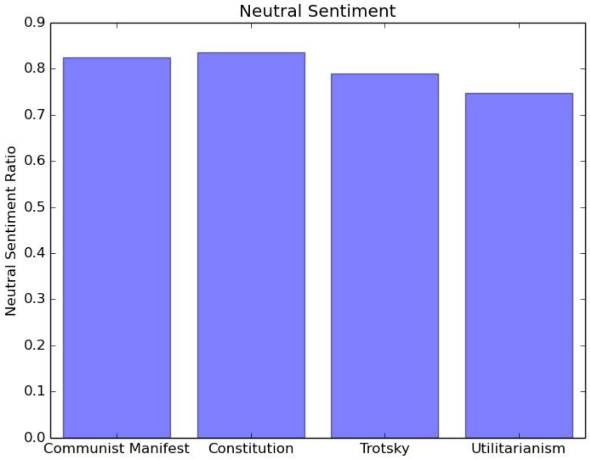
I tried to move forward in my analysis by finding the cosine similarity between the texts, but there were errors in my code that I am still trying to debug. The main idea for this analysis would be to compare two texts by determining the amount of shared words, then finding the frequency of those words in each text and the amount of texts that these words appeared in a list of reference texts. These two values would determine the TF-IDF of each word, which would be used to find the cosine similarity of the two texts (by comparing the lists as two vectors and finding the dot product of the two vectors divided by the magnitude of each vector). Although I think my logic was sound, my execution was flawed and this text is still not running properly.

Results

As the figures below demonstrate, there are some surprising differences between the texts, and some that were to be expected. For Trotsky’s *Dictatorship vs. Democracy*, I was not surprised that this text had a higher negative sentiment than all of the other texts. However, I was surprised to see that *The Communist Manifesto* was more negative by a significant margin than the U.S. Constitution. Another unpredictable result is that John Stuart Mill’s *Utilitarianism* is significantly more positive than the remaining texts. Reflecting on this now, I had some misconceptions about each of these texts, which is why the results seem so shocking.


Negative Sentiment
Negative Sentiment Ratio
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Communist Manifest
Constitution
Trotsky
Utilitarianism


Positive Sentiment
Positive Sentiment Ratio
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Communist Manifest
Constitution
Trotsky
Utilitarianism


Neutral Sentiment
Neutral Sentiment Ratio
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Communist Manifest
Constitution
Trotsky
Utilitarianism

For the second part of my project, the most common unique words for each of the texts were not as surprising. These words represent the words that were most common in each text and did not appear in the other's text most common lists, which does not mean that they did not appear in the other texts at all. The Constitution's unique words consisted of ‘United’, ‘States’, ‘constitution’, ‘representatives’, ‘congress’, ‘senate’, and ‘president’, all of which would be expected for the Constitution. The Communist Manifesto contained equally predictable words, such as ‘abolition’, ‘bourgeoisie’, ‘socialism’, ‘electronic’, and ‘communists’. *Dictatorship vs. Democracy* contains ‘dictatorship’, ‘revolution’, ‘economic’, ‘power’, ‘labor’, and ‘workers’.*Utilitarianism*’s list was comprised of ‘utilitarian’, ‘justice’, ‘happiness’, ‘moral’, ‘good’, ‘desire’, and ‘justice’, which is where the positive sentiment must have partially stemmed from.

All in all these words make sense for each text, and it is quite easy to guess which words belong to which text. The key insights from this analysis is that Trotsky was more of a cynic than I originally thought, and that if there was anyone here that we should grab a drink with, it should be John Stuart Mill.

Constitution:
(['representatives', 'senate', 'several', 'state,', 'about', 'constitution', 'make', 'shall,', 'within', 'during', 'laws', 'office', 'states;', ‘number', 'public', ‘time', 'congress', 'house', 'president', 'each', 'ebook', 'section', 'states', 'states,', 'united', 'shall'],

Communist Manifesto
['abolition', 'class,', 'development', 'communists', 'existence', 'socialism', 'terms', 'therefore,', 'production', 'bourgeoisie,', 'feudal', 'works', 'society,', 'electronic', 'society', 'property', 'social', 'modern', 'conditions', 'bourgeoisie'],

Dictatorship vs. Democracy
['whole', ‘dictatorship', 'socialist', 'workers', 'russian', 'very', brevolution', 'first', 'power', 'economic', 'labor', 'were', 'revolutionary', 'soviet', ‘kautsky'],

Utilitarianism
['utility', 'though', 'desire', 'right', 'same', 'principle', 'good', 'feeling', 'utilitarian', 'general', 'should', 'happiness', 'justice', 'being', 'some', 'moral', ‘human', 'those'])

Reflection

At the beginning of the project, I struggle a little bit because I tackled something that was more complicated than what I was easily capable of doing, which was the cosine similarity. In addition, I poorly executed my code, by simple writing all the code without testing as I went. When I had to debug the code, it was much more difficult to determine the issue. I would definitely improve my debugging processes and implement doc tests when I am beginning to code as opposed to when I am finishing. That being said, I wish I had known to start with something I understood and then tackle something more difficult and to test my code as I wrote it. However, I learned a lot from this project and had fun while doing so!