Text Mining: Write-Up

Ava Lakmazaheri

Project Overview

I read in 12 philosophy books from Project Gutenberg, analyzed them for text similarity, plotted them using Metric Multidimensional Scaling, and then used a Markov model to generate a philosophical mantra from across all of the schools of thought.

I began this project with the hope to compare two opposing philosophies (categorical imperative and utilitarianism) to see if linguistic analysis could find the difference in their content, despite being from the same time period and likely using similar word choice. When neither linguistic nor semantic analysis found a distinction in their meanings, I expanded my text bank to range from Taoism to Nietzsche, and also extended to Markov Text Synthesis.

Implementation

The first task of the program, which only has to be run once, is pickling all of the text files from Project Gutenberg. Once this has been done, the program stores a cleaned version of the text of each book in a global list of strings. The choice to use a global variable here simplified the process of accessing the data and minimized the number of arguments that had to be passed through each function.

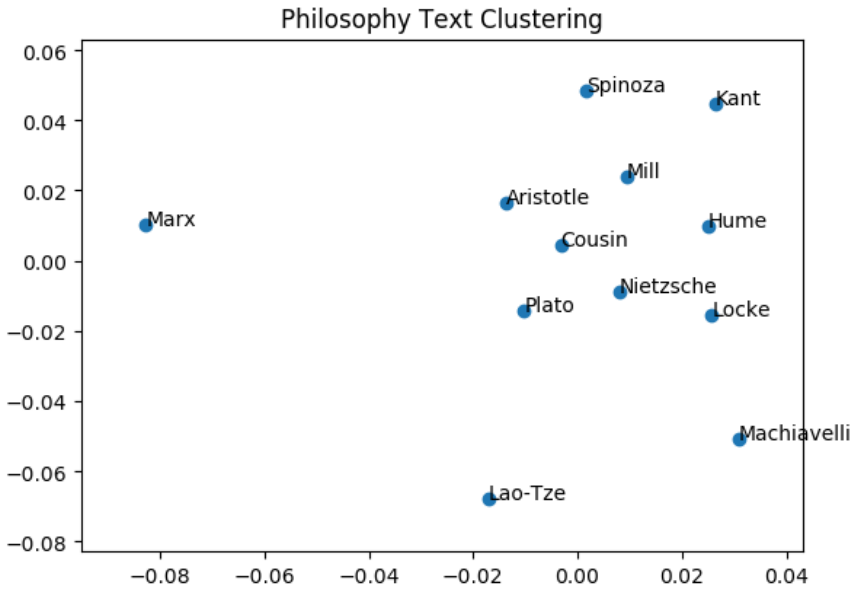
From here, the program broke into two distinct processes: analyzing text similarity and performing Markov text synthesis.

The key data structure used to gauge text similarity is a dictionary that maps a bank of all possible words (all unique words in all books) to its frequency of occurrences in the current text. The code then runs the cosine similarity metric between each combination of books to generate a 12x12 correlation matrix. Finally, the texts are visualized spatially using a scatter plot with MDS.

For Markov text synthesis, all of the texts in the global list are compiled into a giant string. Then tuples are used to document the relationships between words, storing associated prefixes and suffixes that occur in said string. Novel content can then be created by choosing a random start word and following allowed word pairings, until the code reaches maximum sentence length.

Results

Looking at spatial clustering among philosophical texts shows a loose general relationship for most; the outliers include more political texts like *The Communist Manifesto* and *The Prince*. Interestingly, the one sample of Eastern philosophy (*Tao Teh King*) is also pretty distant from the other works.

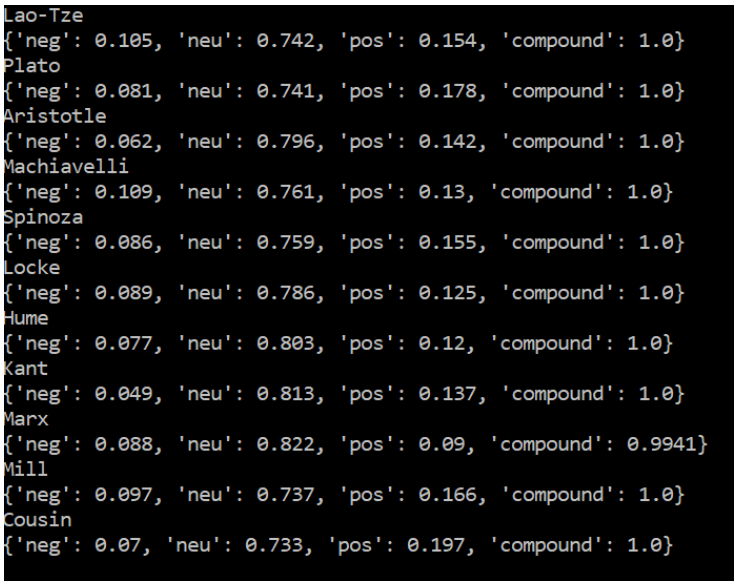


Below are some sample excerpts generated by the Markov text synthesizer:

> "Very good, I believe, I said, such an imperative, i.e., a practical philosophy, where it was always regarded at the sun himself, will not impress self-control on the shores of the sublime. Physical beauty. Intellectual beauty. Moral beauty.--Ideal beauty: it is not the true object and the tyrant. And this is a universal maxim, your action, being opposed to necessity, and not of horses? Of course."

> "As every political community must be supposed capable of the highest notions. Can we deny a providence and a state may be found in no situation; thus also it is also confirmed by some external cause, this endeavour to find a dispassionate survey of imperfect obligation; the latter possessing all the notes of the true, the beautiful, and the people, these are sister sciences--as the Pythagoreans say, and that no body can transfer the known and proved his principles universally and necessarily developed, is admirably fitted for all rational beings."

I also tested semantic analysis on this data set to see if it produced interesting results. Unfortunately but unsurprisingly, as a whole, philosophical texts used largely neutral language.



Reflection

Though I began with a plan of what I wanted my text mining to accomplish (comparing two specific philosophical perspectives), a lack of interesting results along each step of the way quickly dissolved my organization for the direction of the project. This made for a good personal lesson about the limitations of linguistic analysis to interpret meaning, however, it also meant I ended up trying more different types of analysis rather than diving into the intricacies of one in particular.

Implementation-wise, I found unit testing quite challenging for the data sets of this size and nature. I definitely did not plan well for integrating doc tests in my code, and instead checked that functions were working using incremental print statements throughout the program.