Annie Kroo

Software Design

Data Mining: Estimating Literary Eras

Project Overview:

I used project gutenberg to collect a variety of book texts that represent the different time periods that they were written in. With these texts, I separated the long single string into a list of strings that each include a single word. Using a series of matrix transformations, nltk library and dictionaries to compare the relative frequency of words in different books. From this I looked at a book with an “unknown” publication date to see what book it was most similar to. Finally I then used the most similar book’s publication date as an estimate for the "unknown" book.

Implementation:

To compare books I began by separating my data collection from my main script through the use of pickling. This allowed me to run my main mining code without overloading the project gutenberg site. Because I had 12 books that I needed to pickle, I created a file opening function to allow me to easily change how many books I was using and to streamline my process. I then went about sorting the text for which I used the tokenize method of the nltk library. This allowed me to separate the words in each book into words. I chose to use the nltk library because it was already designed to separate words from strings not only by spaces or non-alphabetical characters. This meant that my word counts would be more accurate. To determine the frequency of any one word in a string I used the collections library’s Counter method to create dictionaries for each unique word. This proved useful later in my program as it made it easy to search for a specific word through the use of keys.

With these operators set up I began actually manipulating the imported text. To set up the dictionaries I actuated my file opening function and took down important characteristics of the texts, such as the dictionaries containing their words, individual word counts, and the text’s overall word count. I then created a function for a general dictionary or set of dictionaries that, given a word, would give the relative frequency of the word in the directory or set of directories. This was all brought together in a function that swept all of the words in the un-dated book. In this function I took the average of all of the differences in word frequencies for each book. Finally I looked for the index of the value closest to zero and used that to index a list of each known book’s publication date. This gave an estimated value for the un dated book.

Results:

By running my code with a variety of books as un-dated and comparing the output of my program to the actual publication date, I was able to gauge how effective my code was. Below is a table describing the results of this comparison:

Actual Publication Date	Estimated Publication Date	# of years off.
1845	1890	45
1890	1890	0
1859	1860	0
1847	1820	30
1780	1890	100

The code was able to accurately determine the correlation between the date of publication of the un-dated book and the publication of a series of books whose publication dates were known. This correlation exists because of the fluid nature of our language. Because some words come in and out of fashion and because there tend to be similar motifs and trends in literature from the same era, looking at the similarities in word usage allows the program to fairly accurately estimate when the text was written.

Reflection:

While this was relatively good at ascertaining the era in which a book was written, it certainly could be improved drastically. To improve this program, one could provide more example or known texts. Another issue with this was translation. Some of the texts that I used were originally written in a language other than english and translated decades, centuries, or millennia later. Unfortunately this makes it such that the original words used are corrupted and are less telling of the time period in which they were written. Additionally, more text analysis could be done looking into sentiment, grammar, and themes. This project was certainly a struggle and took longer than I anticipated. This was mainly because I had some trouble grasping the concepts important to completing this code. In the end however, I feel that I learned a lot and am very pleased with my results.