Text Mining *Hamilton* Lyrics

Louise Nielsen

Project Overview:

I used RapGenius lyrics as the data source for the lyrics from *Hamilton* and the packages BeautifulSoup, python-requests, and pattern.en to scrape and analyze the data. I hoped to create a representation of the sentiment of different characters in *Hamilton* and how they vary over the course of the show. I hoped to learn how to scrape from the web and how to graph/present data in Python.

Implementation:

My implementation was intended to include three parts: scraping the lyrics from the web into a usable format, performing sentiment analysis on the scraped and formatted lyrics, and processing the results into a visual presentation. The web scraping used the python-requests library to download the page and the BeautifulSoup library to process the files more easily, as well as a lot of string manipulation. Due to a variety of reasons reflected on in the Reflection section, I did not get farther than this part. The sentiment analysis part would use pattern.en to perform sentiment analysis on lyrics, sorted by character, and store the polarity, subjectivity, mood, and modality of each character in each song. This would allow the third part to compare characters' overall data and to see how a single character changes or develops over the course of the show.

I struggled to find a scrapable source of lyrics for the show, so it was a difficult design decision to choose from which site I should scrape. I ended up trying three different sites (the only sites I could find that presented the full lyrics of *Hamilton*) in succession, none of which proved to be consistent enough to parse effectively. I initially chose http://www.themusicallyrics.com because it did not have the huge amount of links and annotations that http://genius.com and http://atlanticrecords.com had. This proved ineffective, because the downloaded html of pages for different songs were formatted inconsistently. Trying the other two, for similar reasons, also failed. Another difficult design decision was how to download the html files: I ended up choosing the python-requests method because it was better explained to me and simpler.

Results:

I am omitting this section because my project is too incomplete to have gotten results.

Reflection:

Not much of this project went well. Things to improve include time management, mapping out the project beforehand, and choosing a better or more consistent data source. Time management goes without explanation. Mapping out the project beforehand would have made it easier to understand the scope of the project and the amount of time I would really need to invest in it, which would help the time management bit as well. Choosing an easier data source would have been a very good idea for my first text mining project; much of what I was interested in learning was in the parts that I did not get to because I did not have a sufficient data set. I wish I had been more encouraged to really map out my project—what general parts, what components were in each of those parts, and what I need to know to create each of those—and to check to make sure my data set was reasonably parsable before proceeding.