Was mJohn Bozzella
September 2015
Software Design

Mini-Project 1 Overview and Reflection

Overview:

This mini-project was my first experience with writing my own open-ended python script. I had written code for MATLAB in the past, but python is, (well was,) new to me. Since I had no experience, I decided to keep the scope of my project relatively small, as I was not really sure how much I could get done in the first place. My main goal ended up being taking some song lyrics from genius.com, a website for finding and annotating song lyrics which contains a pretty big database of songs, and finding the ratio of unique words to total words in each song. My basic approach was to take the html from the page, turn it into plain text, cut that html code down into just the lyrics in each song, and then to find the word count and unique words.

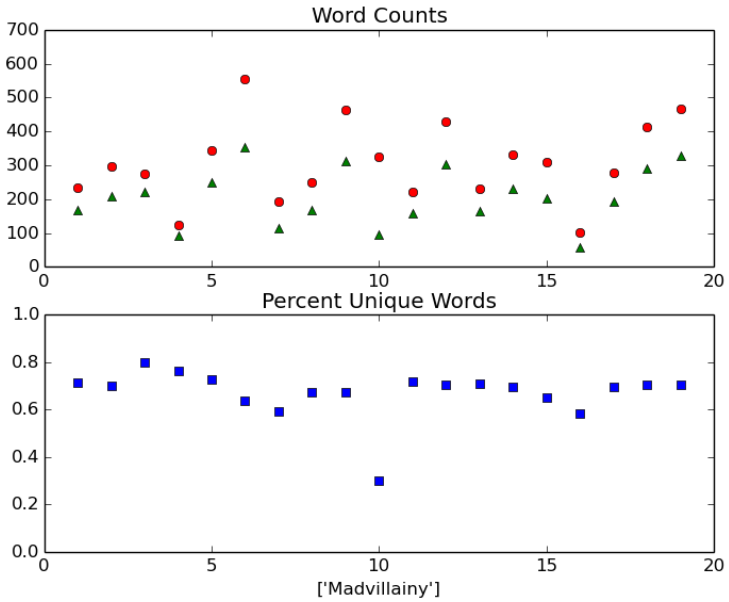
Implementation:

Primarily my code has two functions. The first works by using the pattern library for python to take the html from a specific URL, corresponding to a song, and then turn it in to a list of the plain text by word, as well as a list containing each line of the html as individual elements. These lists are each put into a bigger list, with each element being the list of lines or words respectively. The function then searches the list of words (which is in and of itself an element in a the main list) for keywords in the html that surround the lyrics and outputs a list whose elements are lists of the lyrics from each URL. The second function then takes in that list of lyrics and for each element counts the number of words and unique words and plots both of them using the matplotlib library. The function also computes the ratio of unique words to total words in each song, using some array math which is contained in the build-in numpy library.

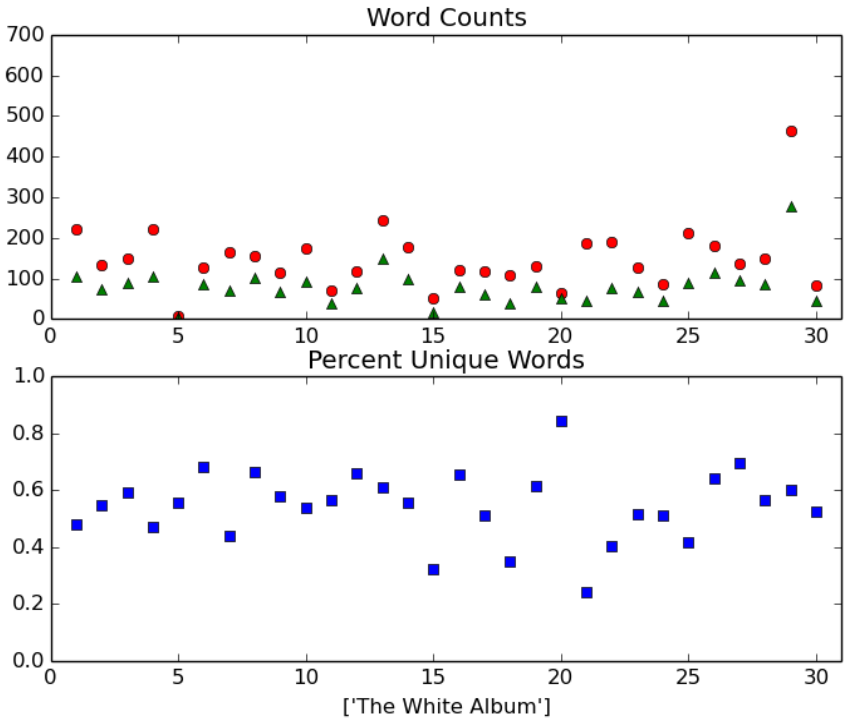
Since the project in its current iteration runs in a script, when you build the entire script in some IDE, say sublime text 2, the only outputs are the graphs of the words per song, unique words per song and the ratio of them together. The given script lyricsimportplot.py already contains four lists of songs that create plots, though the functions can run on any given URL from genius.com. I picked four albums, Madvillainy by MF DOOM, (technically the artist is Madvillain but you can look it up if you're interested,) and then two of the most popular albums of all time, The White Album, by the Beatles, and Thriller by Michael Jackson. I also added the best selling album from 2015, 1989 by Taylor Swift.

Example Outputs:

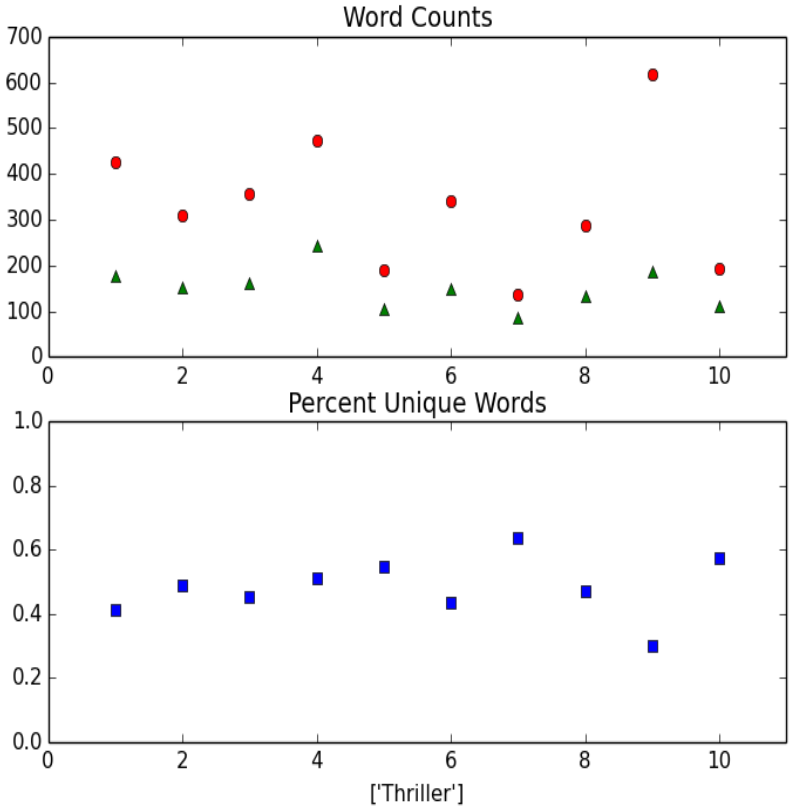
Madvillainy:



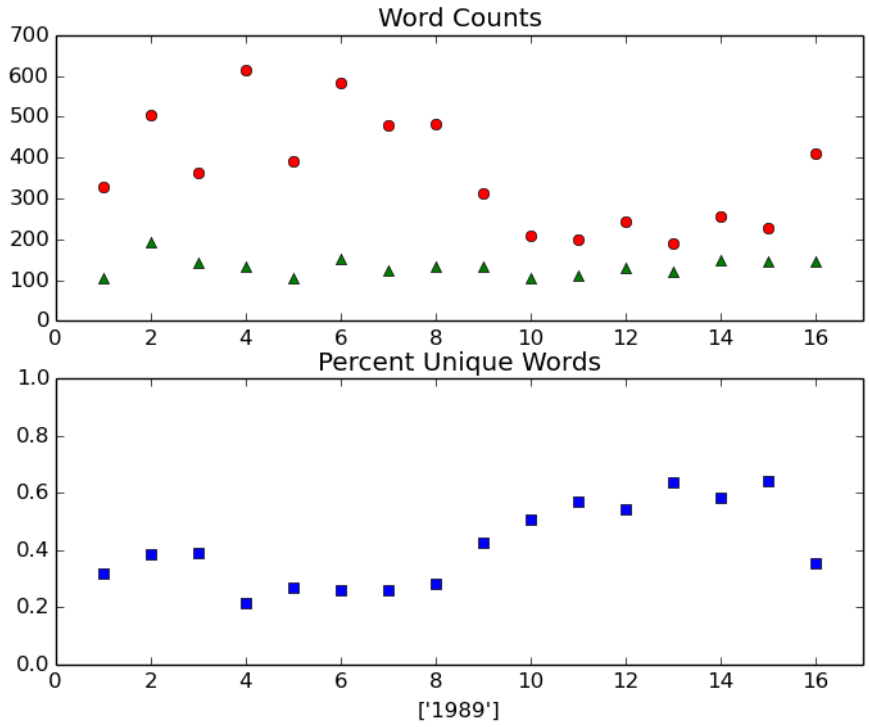
The White Album:



Thriller:



1989:



Reflection:

This project honestly went really well. When I ran into errors or didn't know how to do something syntax-wise, the internet was a fantastically helpful tool. Matplotlib was pretty well documented, and numpy is so common that it is not hard to find help for that, and since I was not really doing anything groundbreaking, I was pretty much always able to find some solution or workaround for all of my actual code issues. I could definitely make the code better, clean up some of the comments and remove a line or two that I think doesn't actutally do anything, but I also want to try to make this work better, maybe even to the point where it could take in a single URL corresponding to an album and then graph the outputs for each song anyway. Also while doing that I would make it easy to run from the command window instead of having to build the script each time.

I definitely didn't try to do too much this project, since I was well aware of the fact that I didn't really know what I was doing at first, so I managed to do everything that I wanted to do. Obviously as the project progressed I would always think of improvements, but it's nearly impossible to totally finish a program like this on the first go around. I made sure that my code would work for any song URL from genius, so I did a lot of testing, for example, four different albums worth of songs to make sure that the outputs were what they were supposed to be, so it was very easy to test to code regularly. I plan to make this program better, useful to the point that if I wanted to look at different artists and some lyrical statistics that it would be useful for that, but for my first program, I think it went really well.