Software Design Web Scraping Project Rebecca Jordan

Project Overview

In my project, I aimed to analyze the objectivity of a fictional character known for his dedication to logic, Spock from Star Trek. I scraped the HTML of IMDb's quotes pages for both Spock and Kirk, and used linguistic post-processing to analyze both the sentiment polarity and subjectivity of each quote. I then graphed these results.

Implementation

The primary challenge in this project was converting the link-filled HTML of the IMDb page to a list of strings that the pattern.en module could analyze. I scraped the website for the HTML and saved it on my computer so I would not need to repeatedly send requests.

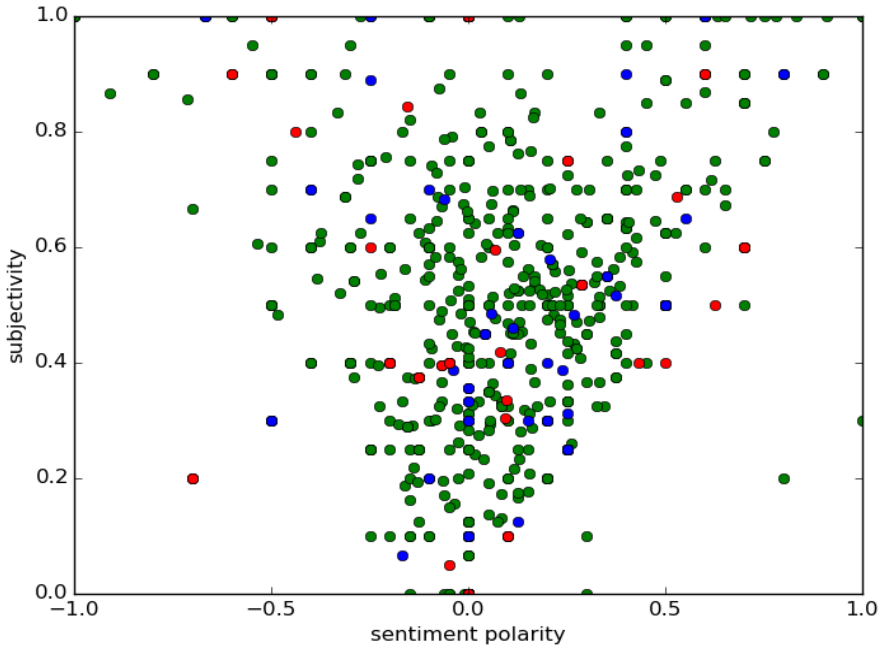
At first I tried using Beautiful Soup's HTML parsing, but could not obtain the quotes I needed. In the course of trying to figure this problem out, I examined the source HTML more closely. The text of the page was formatted in such a way that every name was a hyperlink to the actor, and quotes near Spock's in the script were also included. Rather than using a premade parser to sort through all the code and then distinguish between names in the script and names within quotes, I used this format to my advantage and wrote a simple algorithm that searched the text of the code for Spock's name and the accompanying HTML tags that indicated it was a link to Leonard Nimoy's IMDb page. Having found that, I added the immediately following text to a list of quotes, and used the "<" of the next HTML tag as an end marker. Since Spock was variously referred to as Spock, Mr. Spock, Spock Prime, etc. in different scripts, I searched for each of these names. I aggregated quotes from Spock in the reboot movies and quotes from Kirk that happened to appear on the same page in separate lists for comparison. This method resulted in three lists consisting of strings containing a single quote each.

On each quote in each list, I used pattern.en's 'sentiment' function, which gives a pair of numbers corresponding to the quote's sentiment polarity (positive or negative) and its subjectivity. I then plotted these pairs of numbers as individual points, resulting in a scatter plot for each character showing the spread of positive, negative, objective, and subjective statements.

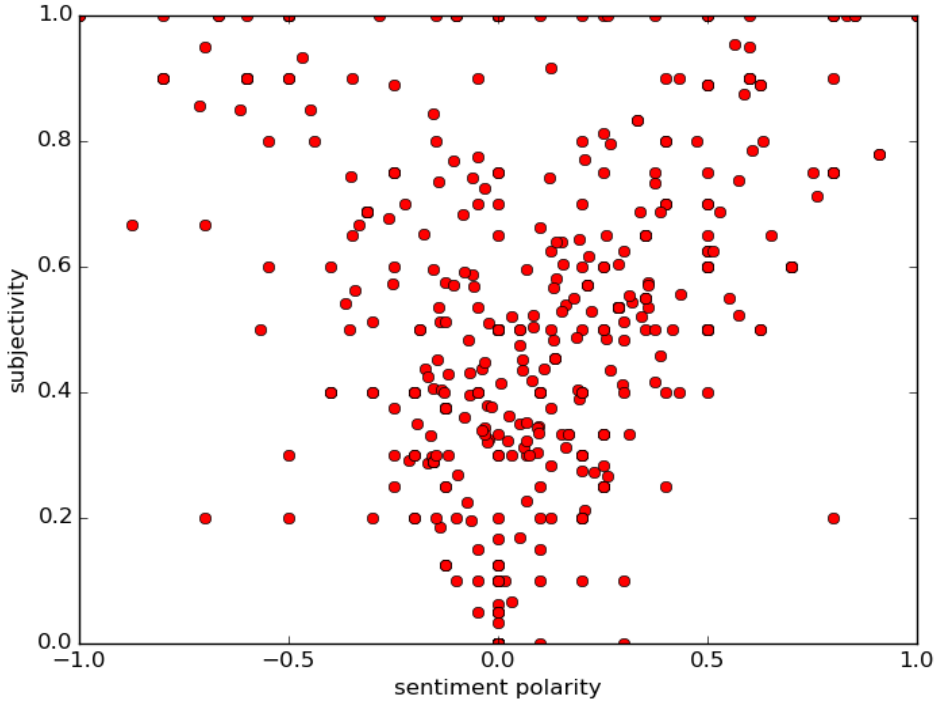
Results

My results are shown in the graphs on the following page. Spock's quotes run the full range from objective to subjective and negative to positive sentiment. I was surprised to see that the results did not skew toward the objective or negative, which I had expected. Nor did the incarnation of Spock in the new movies show any clear difference from the original, though the low quantity of quotes available made this hard to judge. The most intriguing aspect of the result was the inverted-triangle shape of the cloud. The more objective the statement, the more neutral its emotional content. This is, as Spock would say, logical. Statements with strong emotional content are unlikely to be objective.

Wondering whether this pattern was an artifact of the sentiment analysis or a characteristic unique to Spock's statements, I implemented a very similar program to run the same analysis from Captain Kirk's IMDb quotes page, yielding the second graph. As it turns out, his evenly-scattered points in the first graph also resolve into an inverted triangle when more are plotted and there is no readily visible difference between the two spreads.



A graph of sentiment polarity versus subjectivity for Spock (green circles), New Spock (blue circles), and Kirk(red circles). Negative sentiment polarity corresponds to emotions like anger or sadness, while positive sentiment polarity corresponds to positive emotions. The subjectivity scale goes from 0 to 1, with 0 being a statement of fact and 1 being completely subjective.



A similar graph with quotes only from Captain Kirk or comparison.

Reflection

I find the results of this project fascinating. I learned to scrape data from a website, perform automated sentiment analysis, and create graphs with Python. I also found that from a sentiment analysis perspective, Spock's quotes don't really stand out.

However, there are a lot of places in which I could improve. I should go back and learn to use Beautiful Soup's HTML parsing so that in future applications when writing my own search function isn't feasible, I will have another method available. While writing that search function, I was unable to properly translate unicode characters such as apostrophes or remove the \n end line marker from some of these quotes. The number of quotes without these complications is sufficient for me to be satisfied with my results, but I plan to revisit this code soon and solve those issues. Once I solve that issue, I should implement unit tests to ensure that multiple-sentence quotes, quotes with unicode characters, and quotes with stage directions are being properly analyzed.

Despite these specific problems with my code, I feel I gained skills both in coding and in finding existing solutions and methods for solving challenges in code. The structure with suggested modules was very helpful, but I wish resources for HTML and unicode parsing had been available on the website from the beginning.