Project Overview

I decided to analyze the scripts for Harry Potter films in order to do sentiment analysis of the characters over the course of the movie. I used .pdf's of the scripts found on http://thepensieve.org/movieframemain.html, and then converted them to text files so I could parse them in python and obtain everything each character said in the films. I created plots of Harry, Ron, and Hermione's sentiment over the course of the movies.

Implementation

I split my code into two modules: harry_potter_analysis.py and file_handler.py. file_handler handles parsing through the code and creating usable data types, and harry_potter_analysis is responsible for analyzing sentiment and plotting Harry, Ron, and Hermione's sentiment.

file_handler's ultimate goal is to go from a script file to a list of tuples, where each tuple is a quote and the person who said it, (for example: [('RON', 'What is it?'), ('HARRY', 'Some kind of... cloak.')]). The procude is as follows. Using an example of a piece of a Harry Potter script:

```
Scene 1: Doorstep Delivery.
----------
LOCATION: Privet Drive - night
DUMBLEDORE: I should have known that you would be here, Professor McGonagall.
PROFESSOR MCGONAGALL: Good evening, Professor Dumbledore. Are the rumors
true, Albus?
DUMBLEDORE: I'm afraid so, Professor. The good, and the bad.
```

In order to find a quote, we search for the ':' character. Then, to ensure that the ':' indicates a quote, we make sure that the string between the ':' and the previous '\n' new line character is in all caps ('Scene 1' is not a character, 'DUMBLEDORE' is). We then look for the end of the quote. Quotes end either when someone else speaks, or there is a Scene break. Thus, we search for the next occurrence of an all-caps word followed by a ':' and for the next '----------', and whichever comes first indicates the end of the quote. We add the (char_name, quote) pair to a list, and repeat this process, starting at the end of the previous quote.

harry_potter_analysis first uses the list of (char_name, quote) pairs from file_handler and builds a dictionary that maps from char_name to a list of tuples in the format (line_number, quote). We then take this list of tuples and convert it to a tuple of two lists: a list of line_number's and a list corresponding quotes. This conversion is necessary to plot the quotes vs. time (where line_number is the time-axis) in matplotlib.pyplot. Finally, we use pattern.en.sentiment to get the sentiment of the quotes, and we plot a moving average of the character's sentiment over the course of the movie, ignoring quotes with a sentiment of exactly 0.0.

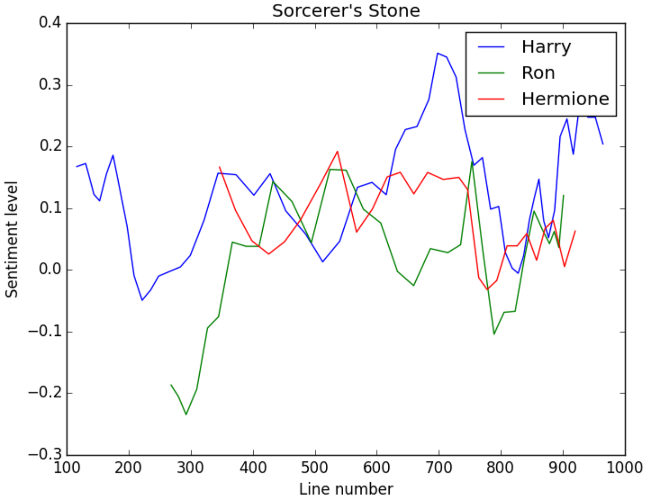
For the most part, the design decisions involved in this project involved choosing between efficiency and readability. I decided to focus mainly on readability, as a movie script is a relatively short text file; each film has roughly 1000 quotes, so the program runs nearly instantaneously. For example, rather than building up a list of (char_name, quote) tuples, then building a dictionary {char_name: [(line_number_1, quote_1), (line_number_2, quote_2)]}, and then building the lists ([line_number_1, line_number_2], [quote_1, quote_2]), it would have been much more memory efficient to just build a dictionary mapping from char_name to a tuple of lists while parsing through the code, rather than creating a long list of (char_name, quote) pairs and converting it to something we can plot later. However, the implementation would have been more confusing and difficult to read, and less intuitive to code.

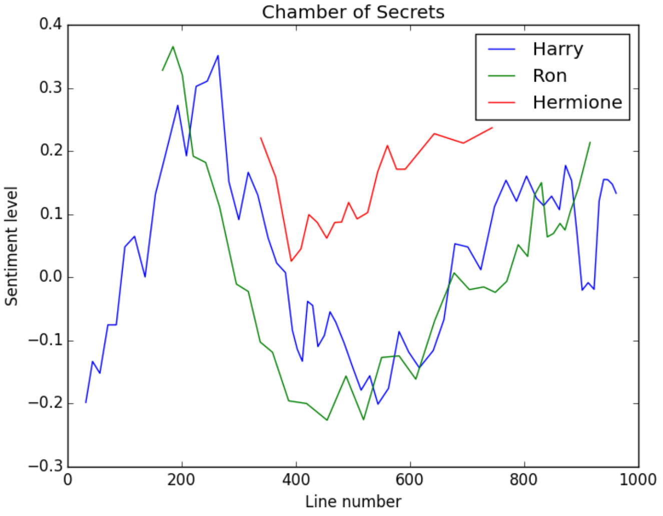
Results

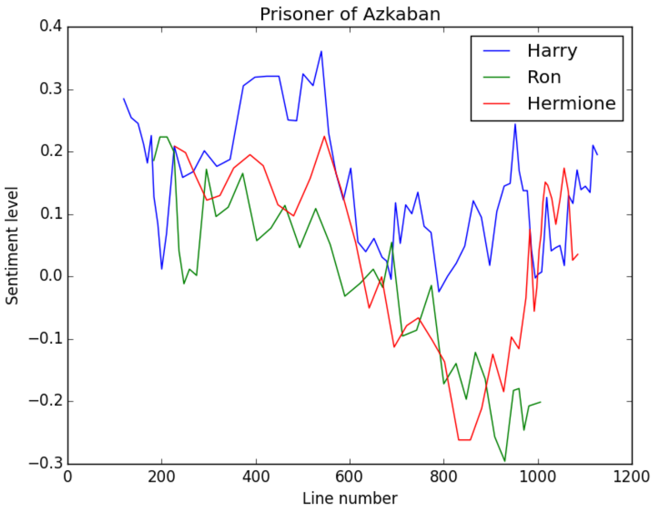
I was able to create figures of Harry, Ron, and Hermione's sentiment over the course of films 1, 2, 3, and 4. It was interesting to see that quite consistently, Harry had the most positive sentiment, and Ron had the most negative. Also, the trio's sentiment trended together.

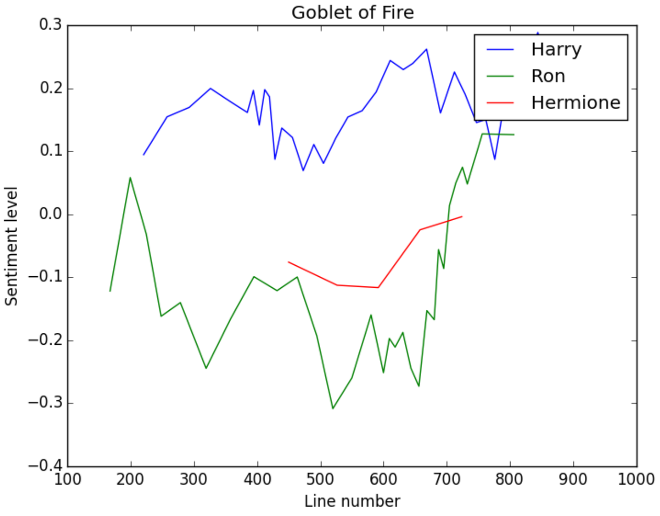
It was also interesting seeing the sentiment change according to events of the plot. In Prisoner of Azkaban Harry is happy when he believes he will get to live with Sirius and he will be free, and then becomes sad when Lupin becomes a werewolf and Pettigrew escapes. Ron spends much of Goblet of Fire with low sentiment because he is upset and jealous of Harry's fame and opportunity to compete in the triwizard tournament.

Below are the plots of the character's sentiments throughout the four films:









Conclusion

When I first starting coding the code to parse the script files, I tried to code quickly and had everything in one long function, with tons of terrible coding practice mistakes (some while True: loops) and confusing pieces of code. I was having strange issues with parsing the scripts and infinite loops kept occurring, and I had no idea why. I had to go back and split my long function into multiple header functions in order to test each piece of code in isolation and find the problem. I wrote unit tests for each function and this allowed me to find the problems. It would have been much easier if I had developed incrementally from the beginning and tested my code as I wrote it, rather than writing huge chunks of code that I had no idea if they worked. I think my project is appropriately scoped; something I'd like to do going forward is see the specific instances in the films where characters were either very happy or very sad.