Oorja Keyur Shah and Dairaneara Haith
Professor Zhi Li
MIS3640
October 21, 2020

Assignment 2: Text Analysis and Text Mining

Project Overview

Movie reviews are an essential way to measure the performance of a movie as people express emotions through reviews. Other than ratings and votes, textual reviews could provide a unique perspective to measure the performance of a movie. Sentiment Analysis refers to the process of processing, analyzing and understanding texts to help us classify and interpret qualitative and subjective data. Movie producers and script writers could use this information so as to understand their customer feedback thoroughly. We used IMDB to find movie reviews for the Fault in Our Stars. First, we installed the imdbpie package in order to be able to retrieve data from the IMDB website. In Python, we read the data into a dictionary and created different functions to analyze the data. Our objective was to analyze the implications of a review by its word count and by specific words that it contains, as we know that movie reviews are the best way to see how the audience reacted to the movie. The assumption was that the more words that a review has, the more opinionated and biased the author was, while the most frequent words give an impression of the movie overall.

Implementation

When inputting the IMDB website data into Python, we knew we needed a dictionary in order to read each review as a separate string key, while also allowing us to elaborate on the words within each review as values. Once we had that established, we created functions to assist us in analyzing the reviews.

Oorja Keyur Shah and Dairaneara Haith
Professor Zhi Li
MIS3640
October 21, 2020

These functions included, “countWords()”, “countFrequency()”, and “authorGraph()”. The first function consisted of a for loop that assigned the word count of each review to that author, giving us the word, freq relationship in the dictionary. The goal of the next function was to analyze the most frequent words in the reviews as a whole, and this consisted of exempting stop words and creating a word, freq relationship for each word used. The last function creates a graph that displays the word count per author for each review.

When writing the function, “countFrequency()”, we had to decide how to display the values in the dictionary. With a list, not all of the words were showing up as it maxed out at a certain amount. So leaving the output in its raw form allowed us to analyze the most frequent words. Another aspect of this function that we had to decide was the stop word. At first we wanted to create a list of our chosen stop words in order to directly evaluate what words we want to keep. After printing the set imported stop words, we decided to go with the imported set because it did not eliminate any of the most important words. An additional challenge that we overcame was ensuring that the graph produced was accounting for the amount of words in each review, rather than the amount of characters. We realized that we were accounting for the number of characters initially, and made sure to use the split function along with the len function to convert that to words.

Oorja Keyur Shah and Dairaneara Haith
Professor Zhi Li
MIS3640
October 21, 2020

Results

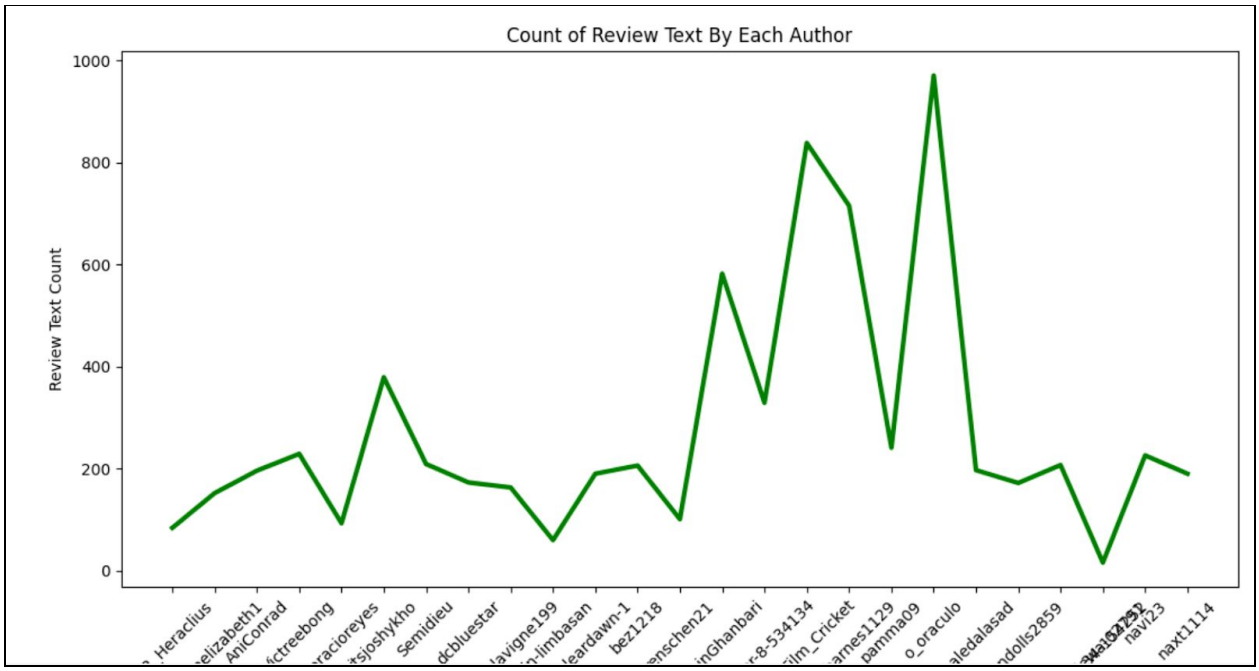

Fig 1. Graph to show the number of words in each review by author

Positive Reviews	**Negative Reviews**
Love (23)	Sad (7)
Like (20)	Monotony (1)
Best (10)	Pessimist (1)
Romantic (8)	Indifferent (1)
Amazing (6)	Hate (2)
Great (5)	Difficult (1)
Favorite (5)	Ridiculous (2)
Beautiful (2)	Horrible (2)

Oorja Keyur Shah and Dairaneara Haith
Professor Zhi Li
MIS3640
October 21, 2020

Fig 2. Table to show words with positive and negative connotation with the number of times each word appeared in a review

The table shows a list of the words that had positive and negative connotations. Seeing the frequencies of the words, words such as ‘love’, ‘like’ and ‘best’ appeared more than words such as ‘sad’, ‘hate’ and ‘horrible’. Through this we were able to see that the majority of the reviews were positive and that the movie was overall liked by its audience. In order to verify this, we looked at user ratings on the IMDB website which correlates well with our results in that the movie has 7.7 out of 10 ratings, suggesting that the majority of the people enjoyed the movie.

Referencing the graph output from Python, it conveys the reviews with the most words and gives an impression of the average word count. From this we got that the average word count of a movie review for The Fault in Our Stars was around 250 words. The top reviews with the most words were from the authors: ‘pamma09’, ‘jfischer-8-534134’, and ‘The_Film_Cricket’. By evaluating the top reviews, we realized that there is a correlation with the number of words to the likelihood of the review being a spoiler of the movie. This relationship can go outside of just this movie and relate to reviews on IMDB as a whole. These top three reviews all gave a summary of the movie while adding a small portion of their actual feelings about the movie. Although these reviews are summaries, they are also biased. Each review has a different summary about the same movie, such as in pamma09’s review they use the words: “real” and “presentation”, while in The_Film_Cricket’s review they use the words: “nothing”, “real”, and “meaning”. These are two opposing views that give a summary of the movie.

We realized during the analysis of this program that certain words could have a double meaning, thus opposing our assumption. For example, the word “Love” appeared twenty three

times and we assumed that this directly related to positive reviews. On the other hand, when accounting for the fact that the genre of this movie is Romance, this popular word could be referencing the genre instead of the review author's impression. Other than examples like this, the other top positive words are more likely to directly relate to the review author's feelings and impression about the movie. Words like "Love" are outliers to our analysis but are important to note.

Reflection

The word frequency determination went well as we were able to predict that the movie had more positive reviews than negative. We were also able to conclude that people who had more to write about, generally had more of an extreme opinion about the movie (positive or negative). Apart from the text analyzing processes that we used, I believe that we could have used more of the natural language toolkit (NLTK Python) for more of an in depth analysis of textual reviews.

We used pair programming as it is easier to work together and help each other as we go along. Having two workers working together, we alternated between the driver and the navigator so that we were both in charge of a different thing each time. This helps us learn from the other person's way of coding and at the same time helps each other improve and learn together. We decided that it was beneficial to pair-program in order to be on the same page and be on track with each other in case of any errors or problems. Some issues that arose while working together were issues with loading certain python packages on our computers. Had one person downloaded a package via Command Prompt and the other had not, there would be issues in running the code due to the absence of the package on one of our computers.

Oorja Keyur Shah and Dairaneara Haith
Professor Zhi Li
MIS3640
October 21, 2020

Another significant issue we encountered was extracting all of the reviews from the website. We utilised the given code to extract the reviews, but were only able to extract the ones on the first page. This meant that our sample was restricted to only twenty-five reviews. Performing the code on a training set and then testing it out on the testing set would serve better results.

On the bright side, we were able to code our program together quite efficiently. There were some things we were confused about, but looking back at our session exercises helped us figure our way out. As well as experimenting with different python modules that we researched from docs.python.org, we came out of this assignment with more modules understood than used in this program.