Catherine Xu

Assignment #2

I. Project Overview

For this project, I used the imdbpie package in Python to retrieve IMDb data. The purpose of this project is to allow the user to enter a film of his or her choice, and provide some insight on that particular film. I delved into word frequencies of the review texts, from which I generated the top 10 most commonly used words of the reviews of a particular film. I then performed sentiment analysis on these texts, generating an average compound score to provide the user with an overall positive/negative outlook on the film by reviewers.

II. Implementation

In the process of finding the word frequencies of each word in the reviews, I wanted to remove all stopwords because of their redundancy to our analysis. I stored these stopwords in a set rather than a list because it is more efficient to check for the existence of a word in a set. Since the reviews of a film in the imdbpie library are given as a list of strings, I converted all words of these reviews to lowercase words and iterated through these words, excluding those that belong to the set of stopwords. I chose to store these words into a dictionary because a dictionary provides you with the ability to map a word to its frequency, which is contingent to my goal of generating the word frequencies of each word.

To retrieve the top n most common words in all the reviews of the film (and in my case, the top 10 words), I took the dictionary of words to their frequencies and generated a list of tuples of (frequency, word). In doing so, I'm able to sort these words in descending order by the most frequently used words to the least frequently used words to generate the top n most common words. To display these results in a more user-friendly way, I used the enumerate function to provide the user with an output of a list (normal, not python list) of n words and their respective counts.

While performing sentiment analysis of all reviews of a film, I noticed that the function SentimentIntensityAnalyzer() provides polarity scores of a single review in the form of a dictionary. I stored and merged these dictionaries of polarity scores of each review into a list to allow for easy aggregation of these results, making it easy to iterate through these dictionaries of scores. Therefore, I could easily add up the stats that I believed would be useful in analyzing the positive/negative outlook of the film and calculated the average polarity scores from these totals as well as simply the total number of positive reviews and total number of negative reviews. I could have performed sentiment analysis by joining all reviews (originally in a list of strings) into one giant string of all reviews and generating the overall polarity scores based on the large

string. However, I chose not to do so because I'm uncertain of the accuracy of scores that the SentimentIntensityAnalyzer function would produce based on only *one* entire group of text. It seems that a sentiment analysis of a larger sample size of reviews, and then calculating the average of the scores, would produce a more accurate result.

III. Results

When analyzing the results, I input two films: *The Dark Knight* and *A Serbian Film*. Consider the results below:

The Dark Knight

```
Hi! This is the film analyzer.

Enter a movie title:
the dark knight
{'title': 'The Dark Knight', 'year': '2008', 'imdb_id': 'tt0468569', 'type': 'feature'}
Analyzing The Dark Knight...
Sentiment analysis of the current movie's reviews:
The total positive reviews is: 23
The total negative reviews is: 2
The average compound score is: 0.8249199999999999
The average positive score is: 0.19687999999999994
The average negative score is: 0.060919999999999995
Top 10 words in the reviews for the current movie:
1. Word: batman, Count: 74
2. Word: dark, Count: 54
3. Word: joker, Count: 49
4. Word: knight, Count: 43
5. Word: one, Count: 38
6. Word: heath, Count: 35
7. Word: ledger, Count: 32
8. Word: best, Count: 32
9. Word: performance, Count: 24
10. Word: will, Count: 23
PS C:\Users\cxu7\Documents\GitHub\text-mining>
```

A Serbian Film

```
Hi! This is the film analyzer.

Enter a movie title:
a serbian film
{'title': 'A Serbian Film', 'year': '2010', 'imdb_id': 'tt1273235', 'type': '
feature'}
Analyzing A Serbian Film...
Sentiment analysis of the current movie's reviews:
The total positive reviews is: 5
The total negative reviews is: 20
The average compound score is: -0.5475559999999998
The average positive score is: 0.09947999999999999
The average negative score is: 0.16207999999999997
Top 10 words in the reviews for the current movie:
1. Word: just, Count: 41
2. Word: one, Count: 36
3. Word: watch, Count: 33
4. Word: can, Count: 32
5. Word: like, Count: 30
6. Word: dont, Count: 30
7. Word: scenes, Count: 29
8. Word: serbian, Count: 28
9. Word: will, Count: 27
10. Word: good, Count: 26
```

Although I have never watched either of these films, I heard from others that the former is a really good film, and the latter is a gruesome film. The results seem to align with what I've heard - *The Dark Knight* has an average compound score of close to +1, and has many more total positive reviews than total negative reviews, and *A Serbian Film* has quite a negative average compound score, with more total negative reviews than positive reviews.

However, as we dive into the top 10 words of each film, I noticed that each individual word really doesn't convey much. However, if we combine some of these words to form a phrase or more descriptive words, these words can provide more meaning. For instance, in the top 10 for *the Dark Knight*, we can combine "dark" and "knight" to form "dark knight" (they are truly meaningless on their own), and perhaps combine "best" and "performance" to form "best performance", which substantiates the highly positive average compound score. In the top 10 of *A Serbian Film*, we can combine "don't" and "watch" to form "don't watch", which solidifies the negative average compound score, but we could also derive other insights from a combination of words such as "good scenes", which may offer a different perspective to the film. Though the sentiment analysis provides a more black and white outlook on the film (whether it is positive or negative), a combination of the top 10 words of each film provides room for more qualitative, critical analysis that may not be categorized as just "good" or "bad".

IV. Reflection

Upon completing this project, I achieved a much better understanding of when it might be best to use a list, dictionary, tuple, and set, which we learned in the previous classes. I also gained a better understanding of how to convert between these different data types, which we covered in

session 13. I believe I was able to achieve some useful insights by choosing to analyze the IMDb reviews through sentiment analysis and word frequency; however, I know that if given more time, I could have looked into improving the project by extending my analysis in word frequency. I could look into the movies and shows provided by the get_popular_titles() function in imdbpie, perform the word frequency analysis of each of these titles, and figure out the similarity between a title of the user's choice and the titles in get_popular_titles() using cosine similarity to recommend (perhaps the top 5) similar titles to the user. This would help take the project from solely analysis to further recommendations. In the future, I hope to go the extra mile and provide more transformation of texts than what is already given.