Movie Plot Analysis with Bag of Words Clustering

Kristen Bystrom, Joy Cundy, and David Dobre

November 21, 2018

1 Introduction

1.1 Problem

In this project, we want to learn about the importance of plot descriptions, their relationships with a movie's success, and the similarities between plot descriptions of various movies. Specifically, we want to:

- Learn what the most common words in movie plots are,
- Compute sentiment for each movie plot,
- Cluster similar movies together based solely on plot
- Compare and contrast the cluster results based on sentiment and other features
- Compare the ratings of movies based on genre
- See if we are able to predict movies based on the genre using machine learning algorithms
- Examine the production of movies over time

1.2 Data Acquisition

We used data from OMDB, WikiData, and Rotten Tomatoes. We acquired these data sets based on the instructions provided in the project description. They came as zipped json files, so we imported them as pandas data frames.

2 Methodology

2.1 Data Cleaning

After converting all our necessary files to pandas data frame, we focused on the movie plot feature in the OMDB data base. We removed stop words from the NLTK stop words dictionary, converted to lowercase, and removed punctuation. We started with 9676 movie plots, but filtered out 432 which contained non-ASCII characters (a semi-crude way of detecting non English plots) which left us with 9244 movie plots.

To analyze the movie genre data, we merged the rotten tomatoes and OMDB DataFrames by merging the shared omdb_id column. This allowed us to examine both the movie plots, genres, and the various ratings scores, as well as awards given to each movie. We then cleaned up the data

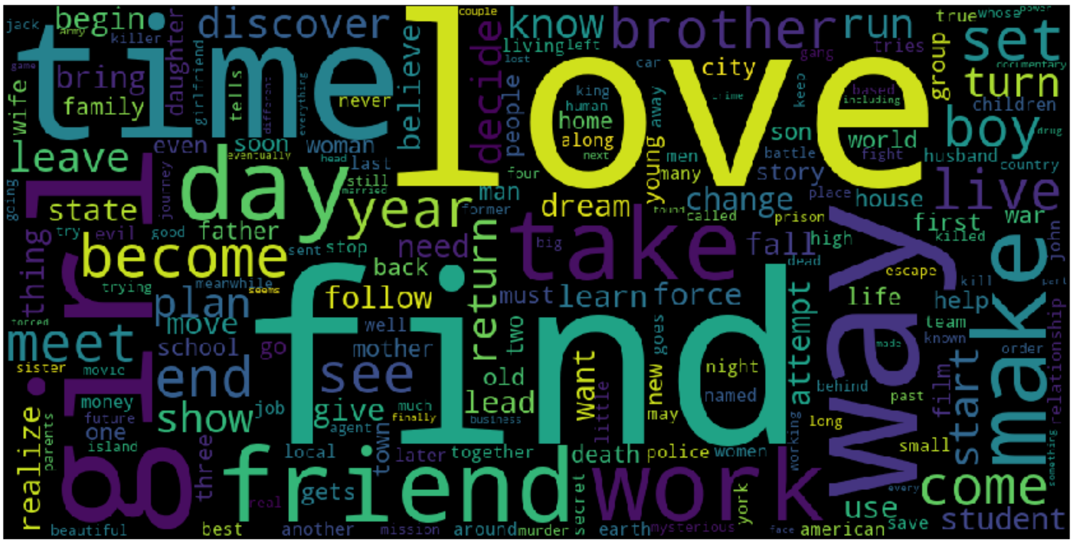

Figure 1: Top 200 most common words in movie plots.

by removing unused columns. For each movie, there were several genres listed, so we split those into separate columns and then melted the genres into a single column.

2.2 Bag of Words

Using the Natural Language Processing Toolkit in Python, we counted up all the unique words that occurred in the movie plots (after cleaning and filtering the data as described above) and obtained the n most common words, beginning with $n = 500$. Figure 1 shows a word cloud of these top 500 words. We then used the collections library to calculate the existence of each word in each plot to form a "binary" Bag of Words (where the values in the DataFrame are either 0 (exists) or 1 (does not exist)). We then used the sklearn.feature_extraction library to calculate the tf-idf (term frequency–inverse document frequency) value for each of words in their respective plots. Both of these methods create a sparse matrix usable for clustering or other machine learning models.

2.3 Sentiment

From the Textblob library, we used the sentiment.polarity and sentiment.subjectivity functions which computes a sentiment score for each movie plot that is scaled based on the data. This is a naive sentiment score based on providing a positive or negative weight to each word in the text string, taking an average, and then standardizing the final result by subtracting the mean of all sentiment scores for all movie plots. A movie with an average sentiment will therefore have a polarity of 0.

As an example, *Monsters Inc.* has a polarity score > 0.8 and a subjectivity score of 0.3 with the following words:

> "mike wazowski james p. sullivan inseparable pair isn't always case moment two mismatched monsters met couldn't stand monsters university unlocks door mike sulley overcame differences became best friends"

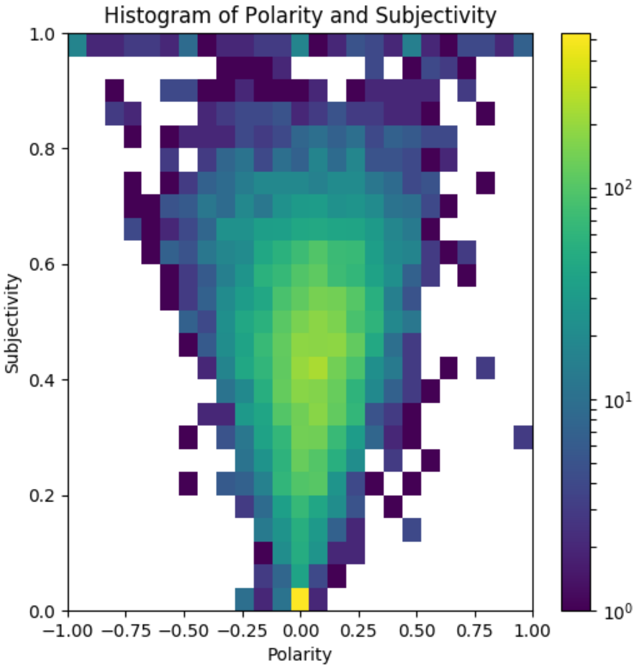

Figure 2: 2D histogram of sentiment polarity and subjectivity.

As another example, *Bloodfist 3* has a polarity score < -0.8 and a subjectivity score of 1.0 with the following words:

> jimmy boland man unjustly accused brutal crime within prison must fight survival freedom justice"

We can see that in the former case, words like "best", "friends", "overcame", and "inseparable" are likely contributing to the high polarity score while in the latter case, words like "unjustly", "brutal", "crime", "prison", and "fight" are likely contributing to the low sentiment score.

Figure 2 shows a histogram of the sentiment polarity and subjectivity (another metric for classifying sentences) for our movie plot data. Slices along specified subjectivities are approximately normally distributed with a mean of 0, but as the subjectivity increases the standard deviation of the polarity increases significantly. There is a large count of movie plots with 0 mean and 0 subjectivity, and with no obvious clusters outside of one other in the center of the histogram (around 0 polarity and 0.4 subjectivity), this does not seem like a good metric to classify the movie plots.

2.4 K-Means Clustering

Finally, with all the pre-processing and variable creation complete, we implemented k-means clustering on the bags of words with $k = 10$. We chose $k = 10$ because we guessed that the clusters might reflect the possible movie genres of which there are approximately 10 (depending on how strict you want to be about what constitutes a genre). The clustering for the binary bag of words (checking only the existence of a word, not its tf-idf value) quick. The 1-Cluster was the smallest with a cluster size of 29. The 6-Cluster was the largest with a cluster size of 2807. Figure 3 shows the cluster sizes in descending order.

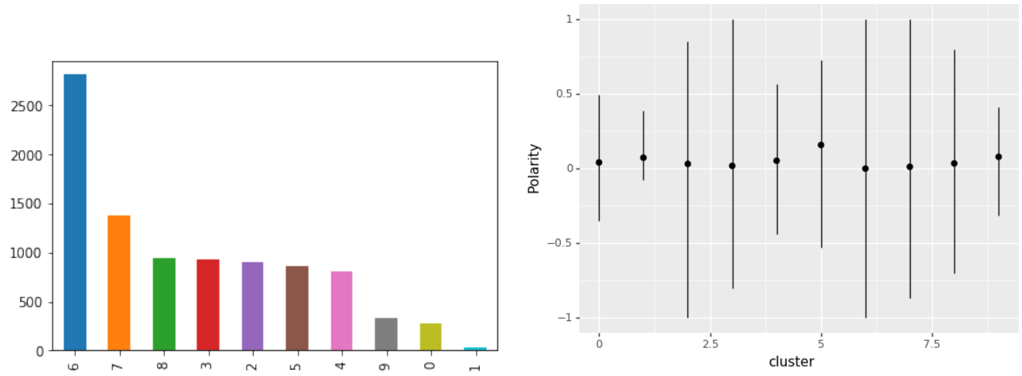

Figure 3: (Left) Barplot of cluster sizes (Right) Sentiment polarity by cluster.

Following the binary bag of words, we tested the TF-IDF analysis as well as much larger sizes of bags (> 1200 maximum features). We found insignificant differences with our results, with similar maximum counts in the largest cluster, and a similar distribution of counts in each cluster.

3 Results

3.1 Cluster trends

We wanted to find out if there are any trends among the clusters. We explored the sentiment, genre, and rating among each cluster in the binary and TF-IDF cases. For the sentiment, the means are not significantly different (they have overlapping confidence intervals), but that the variances between group are different based on a Levene's test with p-value of 2e-24 (Fig. 3.

When analyzing the clusters for their genres, we conducted a χ^2 with a null hypothesis that the clusters are not related to the genre. This was rejected with a p-value $<< 0.5$, indicating that there was some relationship between a cluster and its associated genres. However, inspecting the distribution of genres per cluster did not yield a distinct relationship between the clusters and specific genres. Finally, we briefly inspected any relationship between the clusters and a naive estimation of the rating (average between the critic and audience rating, chosen because of the correlation coefficient derived in Figure 4), but as in the case with the sentiment analysis above, the means were not significantly different. From this analysis, there seem to be no easy and obvious trends that can be extracted by unsupervised learning algorithms, and more sophisticated supervised learning are necessary to create more complete models.

3.2 Finding Genre Popularity

To find out more about the most popular movies, we looked at the most popular genres based on the average critic and audience ratings for each genre. Interesting but not surprisingly, critics tend to rate movies lower than audiences across the board (other than for Adult films, which were not reviewed by critics in our dataset; this was the genre rated lowest by audiences as well). Film-Noir films, News films, and short films tended to be rated highest by critics and audiences. There was not much difference between the two ratings. The only areas where critics evaluated a movie with a higher rating than the audiences did were the Film-Noir and Short film categories.

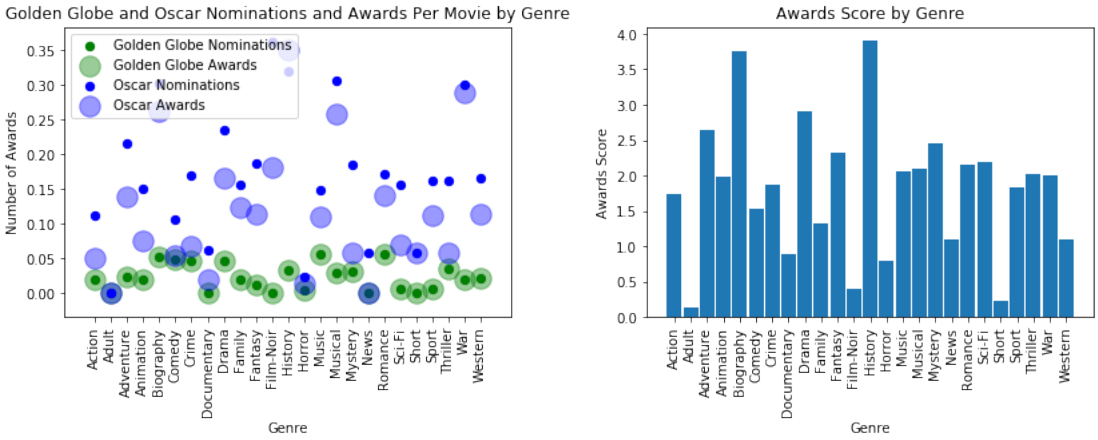

Figure 5: (Left) Golden Globe and Oscar nominations and awards per movie by genre (Right) Awards score by genre.

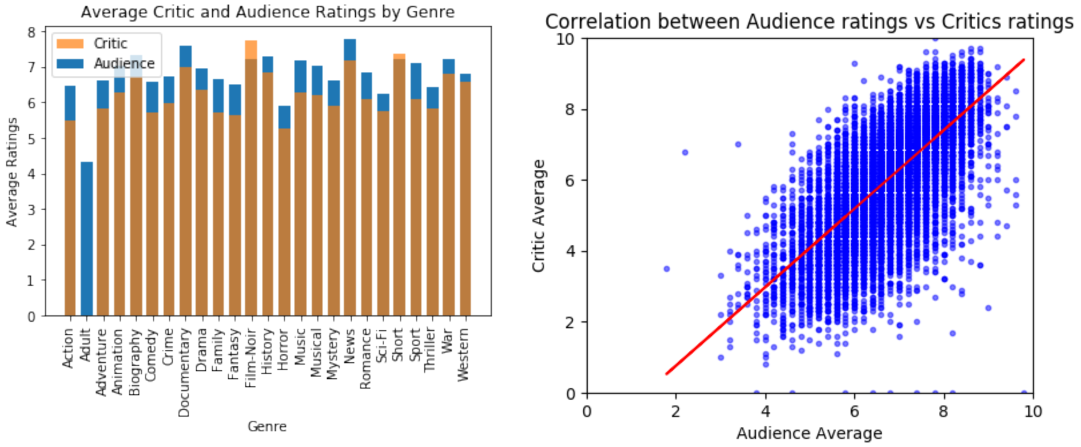

Figure 4: (Left) Average critic and audience ratings by genre. (Right) Correlation between critic and audience ratings, with a correlation coefficient of 0.70 and p-value << 0.5

We also looked at the number of awards given per genre to see if the most popular genres were reflected in the number of awards and award nominations given. The number of awards in the dataset were extracted from strings and divided into award nomination and award wins, and included Golden Globe Awards, Oscars, and Other. We excluded any award names that appeared rarely. The number of awards and nominations in the other category was much higher than in the other two categories, so we excluded them from the graph below. The total number of awards was divided by the number of total movies made for that genre, so that the number is scaled to avoid skewing the data for genres with a higher movie count. History, Biography and War films received highest number of Oscar awards, with Musicals coming in close, but received a higher number of nominations than awards. Golden Globes are awarded less often, and were awarded most often to Biography, Comedy, Romance, and Music films. One could interpret from this data that Oscars tend to be awarded more to films that feature serious topics such as War and History, while Golden Globes favour lighter topics and genres, likely to appear in Comedy and Romance genres.

We wanted something a little more conclusive that also included the awards in the Other category. We decided that the best way to do this would be to give a weighted score to each award/nomination category and then take the average of those scores for each genre. We gave Oscar and Emmy Awards the most weight, followed by Oscar and Emmy nominations, and then Other nominations. We weighted the Oscars because we decided the Oscars to be more prestigious than the Emmy's (this may or may not be true).

4 Discussion & Limitations

Our analysis showed us what the most common words in movie plots are, as well as some of the differences between movies with different vocabulary in their plot descriptions. We learned that different types of plot descriptions have the same polarity in general.

A challenge that we faced in this project is the limits on the size of our bag of words. We used a python Jupyter notebook for a portion of our analysis and there seemed to be some sort of rate limiting that would disconnect us from the kernel if we used to many words for our bag of words algorithm. When $n = 500$ most common words were used, the program ran a bit slow but was able to run to completion. However, when we tried to run the algorithm with $n = 5000$ most common words, the Jupyter notebook we were using which was run from SFU's server (sfu.ca/syzygy) would immediately terminate the kernel. After porting the Jupyter notebook code to python, the performance boost made it easy to handle larger values of $n(> 1200)$, however no appreciable difference in the results was noted.

When comparing the average rating to the movie genres, we first wanted to use the genre names listed in genres.json.gz. This file only contains two columns, the wikidata_id column and the genre types column. However, when merging this with the DataFrame collected from wikidata-movies.jason.gz, only four matches existed due to some labeling error in one of the files. The file omdb-data.json.gz contained a listing of genres, so we were able to merge that DataFrame with the DataFrame from rotten-tomatoes.json.gz to get the necessary data for comparing the movies genres with the audience and critic ratings.

When first comparing movie genres to the number of awards, we kept the lists of genres together, but quickly realized this let to too many separate sets of data which didn't lead to any conclusions since there were as many as three genres for each movie. To solve this, we split each genre per movie into separate columns. We then made the assumption that the first movie listed may be the most prominent genre, and compared critic and audience ratings for the first genre listed. We then decided that this assumption may be incorrect, and since there was no way to verify, we decided to use each genre listed per movie equally. Moral of the story: we don't always know how some information is collected, and we have almost no way of finding out. This method of cleaning the data reduced the chance that certain genres would be left out, but also means that each audience rating and number of awards per movie is counted up to three times.

When comparing the number of awards/nominations with movie genres, we first collected the total number of awards and nominations equally, but this did not give a fair representation because the number of awards received could be said to be more important than the number of nominations. We also have no way of knowing what the "other wins" and "other nominations" are or how prestigious the award is. In many cases there were many more awards/nominations in the "other" category that it dwarfed the Golden Globe and Oscar award data in graphs, so we had to come up with solutions to include it in our analysis or leave it out completely.

5 Project Accomplishment Statements

Kristen's Accomplishment Statements

- Cleaned movie plot text to prepare for analysis
- Computed polarity for 9244 movie plots
- Implemented binary bag of words in Jupyter notebook using collections to find clusters of similar movies
- Visualized movie plots through use of word clouds, histograms, box plots, and bar plots
- Wrote 3 pages of project report

Joy's Accomplishment Statements

- Compared the movie genres with the mean of the total average audience rating and mean of the total average critic rating
- Visualization of this comparison through stacked bar graph
- Compared the genre of a movie with the number of awards and nominations, used graphs to visualize this information
- Compare number of movies made over time

David's Accomplishment Statements

- Expanded on the movie data cleaning to include genre analysis
- Expanded the sentiment analysis to include subjectivity
- Rewrote and optimized code for improved speed performance in regular Python, allowing us to study much larger bags of words
- Implemented TF-IDF analysis in python to compare with the binary bag of words originally studied.