Movie Script Corpus Analysis

Michael Alcorn, Ronit Joshi, Prateek Machiraju, Neil Vakharia, Connor Vines

April 29, 2023

All source code is provided here: `https://github.com/prateekma/comp590-NLP-project`

1 Motivation and Problem Definition

Since the day we are born, our life is dominated by the entertainment industry. From the educational movies we absorb as pre-schoolers to the films we spend our Friday nights watching, cinema has an unseen impact on our lives. For audiences, an exploration of movies can help them be more aware of how multimedia influences individuals and groups. For screenwriters, an exploration of movies can give valuable insight into the core components and natural flows that successful movies tend to have. Therefore, we ask two essential questions:

1. How can we use the script of a movie to understand and predict the overall sentiment and vibes that the movie presents?
2. Given an understanding of how a movie unveils its dialogue, tones, and plot points, can we generate new scripts that maintain popular conventions?

If we are able to answer these questions, not only will we gain a better understanding of the art of movie-making, but we will also be able to create new and engaging content that can captivate audiences.

2 Related Work

Although machine learning is instrumental to entertainment systems such as the AI recommendations of Netflix, natural language processing has been relatively unexplored in cinema. Most scientific analysis surrounding film scripts has been done through universities. A Stanford study uses deep neural networks to predict IMDB review scores and Worldwide Gross Revenue. As shown in the study, the predictions became significantly more accurate when using word embeddings and increasing the granularity of the data, a trend that is seen in our experiments as well (Gross et al.). Nevertheless, no substantial analysis has been done on using components of the script to characterize the atmosphere of the movie itself.

3 Data

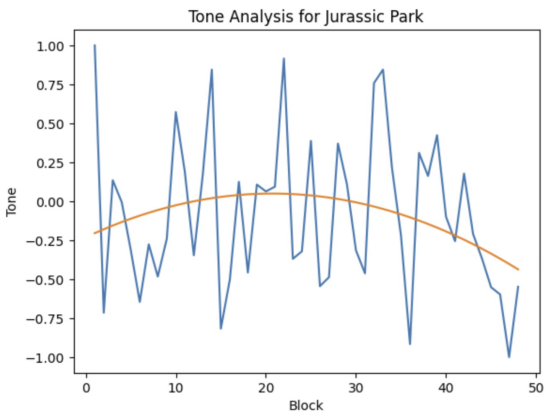
Our dataset, Film Corpus 2.0, comes from the Natural Language and Dialogue Systems lab at UC Santa Cruz. The corpus is a compilation of hundreds of movie scripts in the form of text files. These scripts are split up by genre, including Mystery, Action, Comedy, Horror, etc. Some genres are more populous than others - Music has 5 scripts, but other genres have over 70 (Baskin Engineering - UC Santa Cruz, 2015).

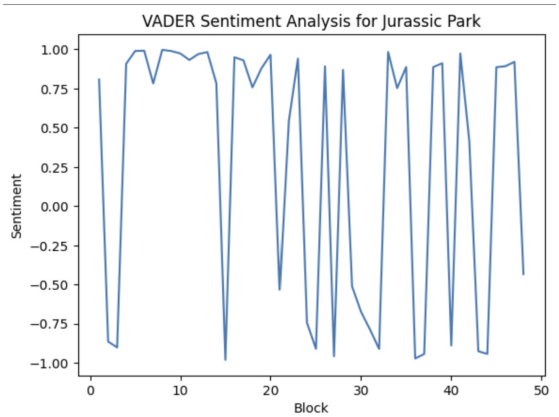
4 Method Description, Experiment, and Results

4.1 Sentiment Analysis - Capturing Tone

For our first goal, we felt it was imperative to create an algorithm that captured tone. Whereas sentiment is simply a metric for positivity or negativity, tone is far more layered. As a baseline, we analyzed Jurassic Park (Spielberg, 1993) as the movie has varying shifts of humor, seriousness, and intensity. We began with some essential pre-processing. In the scripts, the character who says each line was given before the line itself, ie: "HAMMOND \n We have real dinosaurs out there now." These line signatures were removed, and text was lowercased afterwards. From there, we looked into segmenting the script into blocks where each block captured a unique tone the movie presented. This required some tuning; a block length of 1 sentence was too granular because many sentences were simply neutral filler; a block length of 200 sentences had too much content to distinguish precise shifts in tone. A block length of 30 was chosen because an average scene is around 30 sentences.

Then, each block was pre-processed further. Punctuation, numbers, and stop words were removed as they would add unnecessary influence on our tone calculations. A list of tonal words were established. The positive tone words were: joy, excitement, happiness, delight, enthusiasm, love, contentment, serenity, and fun. The negative tone words were despair, fear, sadness, disappointment, apathy, hate, misery, chaos, and boring. These words are popular tones that offer a more nuanced picture than the simple classifications of positive and negative. Using the GloVe representation of word embedding vectors, we began to calculate tone. For each block, we computed the total cosine similarity between each word and each positive tone subtracted by the total cosine similarity between each negative word, all divided by the number of words. The score was normalized to a [-1, 1] range. The results were plotted below and compared to a baseline naive sentiment analysis on each block.





Although there is no concrete accuracy, we used context to evaluate the strengths and weaknesses of our models. As seen above, the VADER sentiment analysis is successful in capturing climatic events. Each dip in the graph represents a very negative event such as the dinosaur escape or when a character gets mauled. However, the issue with the standard type of sentiment analysis on Jurassic Park and other

movies well is that the scores are very polar. They jump between -1 and 1, but tend to miss the values in between. As we know, the script of Jurassic Park has more emotional depth and has varying intensities of optimism and pessimism. We believe the word-embedding based algorithm captures the tone much better, and as with the movie, increases in cheeriness and wonder as the protagonists become more and more fascinated with the park and its residents. However, after the first act, we see a steady decline as Isla Nublar falls into mania. Upon inspection, each block's actual text generally has the tone that we calculated. When looking at other movies, a similar trend is seen as well, where sentiment analysis gives a good idea of polar intensity, but tone analysis provides a much more holistic picture of how the energy of the movie flows all throughout.

4.2 Sentiment Analysis - Predicting Genre

An important question to ask when dealing with calculating the sentiment of a movie is: would we be able to predict the genre of the movie based on a set of features that encodes the movie's sentiment? Training the data required defining some hyperparameters: the number of plot-parts the movie is evenly split into, N_p where each plot part signifies a section of a plot diagram or narrative arc (rising action, climax, conclusion, etc). A chunk is the smallest unit of text (a dialogue or conversation) that has the VADER sentiment analysis run on it. The chunk size, c, is defined as the number of lines in a chunk. The values $N_p = 5, c = 30$ seemed to be the optimal hyperparameters for this problem.

Running the sentiment analysis on each chunk gave us scores for positive, negative, neutral and compound sentiments. These were averaged across all chunks in each plot part. Therefore, for each plot part i, the training resulted in features for average mean ($\bar{\mu}$), and average standard deviation ($\bar{\sigma}$) for each sentiment type: positive, negative, neutral, and compound (each denoted by $\bigoplus, -, \sim, \bigotimes$ respectively). So for a certain movie, the features were $\bar{\mu}_{k,i}$ for $i \in \{1, ..., N - p\}$, and $k \in \{\bigoplus, -, \sim, \bigotimes\}$, which was a total of $8N_p$ features. The feature pre-processing was done for each movie and encoded into separate files.

A multiclass classification problem of predicting one of G genres given the $8N_p$ features proved to have extremely low accuracy compared to the baseline. Instead, we trained a binary classification logistic regression model for each genre to predict whether or not the input movie belongs to the specified genre. The train-test split was $80\% - 20\%$. For each model (each genre classifier), the number of positive instances (instances that did belong to the genre) was equal to the number of negative instances, for both training and testing. In other words, half of the movies each genre's classifier was trained on did belong to that genre, and the other half didn't. The same was true with the test set, **which meant that the majority-vote prediction baseline was** 50%. This was done so that the models won't be too used to predicting negatives.

Training the models on different subsets of genres yielded very interesting results: Training similar genres, such as {Action, Crime, Adventure, Sci-Fi} resulted in low accuracies (likely because the models were being trained on similar features for all genres). Training polar genres, however, such as {Romance, Horror} gave us very high accuracies, as it was easy to distinguish between them. In the following data, each table shows a different subset of the genres trained on the models, and the accuracy, precision, and recall for each model. The ones on the left are "different" genres, whereas the ones on the right are "similar".

Model	A	P	R
Romance	0.835	0.814	0.875
Horror	0.869	0.848	0.903

Model	A	P	R
Sci-Fi	0.524	0.531	0.531
Action	0.504	0.504	0.871

Model	A	P	R
Adventure	0.623	0.608	0.718
Horror	0.852	0.843	0.871
Comedy	0.609	0.608	0.634
Romance	0.785	0.784	0.801

Model	A	P	R
Action	0.547	0.563	0.458
Crime	0.578	0.589	0.548
Adventure	0.532	0.545	0.462
Sci-Fi	0.412	0.419	0.406

Additionally, we also trained the models on each pair of genres and ordered it by accuracy. The top 5 highest accuracy pairs were {Action, Romance}, {Sci-Fi, Romance}, {Comedy, Action}, {Thriller, Comedy} and {Romance, Adventure}. This makes sense, as these are all very polar genres. The top 5 lowest accuracy pairs were {Drama, Comedy}, {Comedy, Mystery}, {Sci-Fi, Mystery}, {Drama, Adventure} and {Drama, Crime}. This is because they are very similar genres, which caused the models to have trouble differentiating between them. Overall, this data seems promising, and leads to the conclusion that a statistical analysis on the sentiment of each plot part is a decent metric to identify the genre.

4.3 Script Generation

For our second goal, we wanted to generate new scripts that maintain popular conventions while also being unique and engaging. We began with an n-gram model trained on specific movies, and increased n from two to five. For example, when using a bi-gram model on the Harry Potter series, we received sentences such as "professor mcgonagall looks regards hermione ron chuckles uneasily. dark forest home dobby brightens", which lacks any larger structure; however, pairs of words are relatively reasonable: "Professor Mcgonagall", "Hermione Ron", etc. When using a 5-gram model, the sentences become much more coherent: "he shakes [his] head vaguely annoyed [and] starts [to] move. He stops stunned. fleur: do [they] want us back [in the] hall. Harry turns [and] finds Krum, Diggory, [and] Fleur majestically roaring fire." Once we add previously removed stop words as shown in brackets, the five-gram model makes much more sense than our bi-gram model and creates a believable generation of Harry Potter, with stage directions as well as dialogue.

The next step in our iteration to find the best model to generate movie scripts was a character-level feedforward neural network. We adjusted several parameters of this model, including character window size (i.e. how many characters of context should be used to generate the model) and the number of epochs to produce the lowest cross entropy loss in the model. After tuning these parameters, we ended up with a reasonable cross-entropy loss of 0.0004745. The results of the feedforward neural network resulted in significantly better script generation and some samples of these scripts are shown below:

```
LEIA
Anoat system.  There's not much There.
HAN
No. Well, wait. This is interesting. Lando.
LEIA
Lando system?
HAN
Lando's not a system, he's a man. Lando
Calrissian. I'm the administrator of this
facility.  And who might you be?
```

```
LEIA
Leia.
LANDO
Welcome, Leia.
HAN
All right, don't lose your temper. I'll come
right back and give you a hand.
```

```
RIEEKAN
Solo?
HAN
RIEEKAN
Commander Skywalker, do you copy? This is Rogue
Two.  this is Rogue Two.  Captain Solo, so you
copy? Commander Skywalker hasn't come in.
```

Below is another example of a script generated by the neural network. In this example, the model doesn't quite generate English at times.

```
PIO
It sounds like Han.
LANDO
(in the gilll. seru!
LANDO
Wh, wha beailes. Thed Pre't sher.
REETSON
Come in!  Lake a chat ore of this pithe
pederays woll pearing andreq?
HAN
Youh, into That boply nowllly. Ches I mand
that'm nit hemp,rie.
```

```
HAN
I'm no too gon.
LUKE
(into comlink) We're going to the Dagobah system.
LUKE
(into comlink) Yes, Artoo?
LUKE
(into comlink, chuckling) That's all right.  I'd
like to keep it on manual control for a while.
HAN
(harried)I saw them! \
```

These errors can be fixed with finer tuning of the model, and perhaps moving to a feedforward neural network with a wider context or one that uses larger tokens (i.e. words or parts of words). This research is still on-going, but the results with character-level generation are very promising thus far. Unfortunately, we were unable to get a Transformer model for generation completed before the deadline, as we had many issues in its creation. However, given more time or future research into this project, a transformer model would be the next step we took in script generation.

5 Conclusion

In this study, we explored the impact of movie scripts on the sentiment of a film and the potential for generating new scripts. By utilizing sentiment analysis techniques, we were able to capture the tone of movies like Jurassic Park. We also were able to predict the genre of a film based on its sentiment. We found that increasing the granularity of our script generation models resulted in more coherent and contextually accurate sentence generation. Although we did not complete a Transformer model for generation, our findings indicate that our approach has the potential to contribute to a deeper understanding of the art of movie-making. Each member of our group made significant contributions to all aspects of this project. Michael was responsible for research and also assisted with coding in every aspect. Neil led the tonal analysis. Ron focused on the genre prediction aspect of this project. Connor was responsible for the N-Gram generation models, and Prateek led the development of the Feedforward Neural Network. We worked collaboratively in trying to get the transformer to work.

6 Bibliography

Baskin Engineering - UC Santa Cruz. (2015). Film Corpus [Dataset]. In NLDS Corpora - Natural Language and Dialogue Systems (2.0). UC Santa Cruz. https://nlds.soe.ucsc.edu/fc2

Gross, Joshua A., et al. "CS 230: Film Success Prediction Using NLP Techniques." 2021, http://cs230.stanford.edu/projects_winter_2021/reports/70992925.pdf