Sentimental Analysis on Reviews of Christopher Nolan's Works

Zejia (Harry) Chen

Introduction

I have always considered myself a cinephile. Normally on the weekend, I will pick an intriguing film to enjoy with my friends as a weekly ritual. One of the ways that we determine if a film is watchable or not is by looking up its review from various sources. However, we often found ourselves spending too much time online browsing through reviews that we did not leave enough time for enjoying the actual film. As a result, it will be super helpful if there is a way to easily identify different sentiments within a series of film comments, making every film lovers' life much easier by not wasting their precious time.

Plus, I am also a big fan of Christopher Nolan and his directing style. Nolan is an eclectic director who has worked on films ranging from sci-fi to historical. I am super interested in his work not only because he is one of the best directors in our time but also due to the controversy surrounding his works. No matter is his most received film like *Inception* (2010) or some of his less-known works such as *Insomnia* (2002) and *Following* (1998,) there are always different voices in response to his cinematic language, making us even harder to make a clear judgment about his film's quality.

Hence, in this research paper, I would love to conduct a sentimental analysis based on reviews of Nolan's films throughout his entire career. With the scientific approach, I hope to correctly identify different sentiments in reviews of Nolan's work and to find out features in the film that are most welcomed/rejected by the audience. By finally evaluating the performance of different methodologies, I determine to offer a sentiment classifier that can accurately reflect audiences' views and offer pertinent guidance to people who are struggling to find the right movie to watch.

Related Research

I found in total three existing works that are apropos of the purpose of my research:

1. Analyzing Movie Reviews using Sentiment Analysis, Gayathri Devi Nagalapuram
2. Analyzing Star Wars Movie Scripts, XAVIER
3. Sentiment Analysis for Movie Reviews, Ankit Goyal & Amey Parulekar

The first article is a major reference to my completed project. Since the research topic is identical to mine, the paper treats as a useful reference about directions that I should consider to complete the project, guiding me to structure the analysis in the right way step by step. On the flip side, instead of scrapping the review online, the author utilizes a pre-existing dataset from Kaggle to conduct her analysis. Compared to her dataset which contains film reviews from the past several decades, my scope is much smaller.

The second project on Kaggle is also about conducting a text (sentimental) analysis pertaining to film-related text documents, in this case, a movie script. The author hopes to perform a statistical analysis text analysis on the Star Wars script from the Original Tribology Episodes to showcase the most frequent feature words. However, since the entire text analysis is based on R programming, there are not many things that I can reference directly from the author's code. What is more, instead of analyzing the overall sentiment of one large chunk of text, I will be dealing with a large amount of short and fragment reviews. Plus, the text (Star Wars script) that the author uses for analysis is also more structural which requires disparate approaches in terms of cleaning and modeling compared to mine.

Unlike the previous two projects which both are self-published online, the third related work is much more formal. It is a rigorous academic paper authored by two college graduates. One of the paper's goals is to explore the performance of various statistical models (Logistic Regression, k-Nearest Neighbor classifier) applied to the film reviews. Such comparison is something that I want to include in the latter part of my project. It will be extremely useful to envision my own approach, by observing some of the menologies they adopt. In addition, since it is an academic research paper, its general structure, as well as its style of writing, will liken my final project write-up. Despite the similarity, the paper's methodology is more rigorous where the author creates different versions of bags of words to test out the model's accuracy. For the purpose of my research, on the

other hand, I am going to use only two bags of words to extract meaningful features from the review text.

CORPUS

Movie-rating websites are often used by critics to post comments and rate movies which help viewers decide if the movie is worth watching. Instead of finding the data on conventional movie rating websites, like IMDB or Rotten Tomatoes, where most of the reviews are made by professional critics, I decide to come up with a more specific dataset that is more tailored to my research direction.

The website that I am going to get the film reviews is called ***Letterboxd*** – a social platform focused on sharing opinions about, and love of, films. The platform allows users, many of whom are just film lovers, to rate and record their opinions about films which makes it ideal to conduct my desired sentimental analysis. Plus, I want to keep the scope of my research topic more specific. Hence, I am planning to scrap reviews of Nolan's film on *Letterboxed*.

Films that I selected from Nolan's career are as followed:

Film Title	**Year**	**Number of Reviews**
Inception	2010	370
The Dark Knight	2008	374
Interstellar	2014	326
Dunkirk	2017	320
Tenet	2020	255
Memento	2000	389
The Prestige	2006	377
Insomnia	2002	364
Following	1998	418

Thanks to the clean formatted website URL, I was able to scrape all the reviews I need with `Request` and `BeautifulSoup`. To ensure the quality of reviews, I decided

to scrap the review in an order of "Review Activity" (amount of likes, comments received.) To store the data locally, I first put the review in a data frame and subsequently stored the data frame as a local CSV file. In addition, I preprocess the data while I was scrapping it from *Letterboxd*. After examining some pages of reviews, I have come to an understanding that there are non-English reviews on *Letterboxd*. As a result, I took this into account while I was collecting the review from the website. I designed a function `isEnglish()` that takes an additional step to check if a review is consisting of English characters, ensuring the corpus is free from reviews that consist of non-English characters or emojis.

In terms of the actual dataset, it consists of two columns of data – movie rating and review text. Both the data are an inextricable part of my project. The review text forms the basis of my text analysis whereas the rating is treated as a critical metric to measure characterize reviews' sentiment which I will cover in more detail when expiating my methodology.

PROCESS & METHODS

My text analysis consists of two parts:

1. Review Sentimental Analysis using VADER & Tokenization
2. Binary Classification with Logistic Regression

Via this two-part process, I first wish to have some sorts of preliminary results and insights using hands-on methods such as VADER and Tokenization. Then, with a clearer direction in mind, I am going to refine my existing analytics pipeline using logistic regression and hope to gauge the performance between the two models.

Above all, to evaluate the NLP's methods efficacy, I must have some sort of benchmark to validate their prediction. Hence, I manually classify the review sentiments into two general categories, positive and negative, based on the rating given by reviewers. A piece of review is considered as a positive review if its rating is greater than three stars and is seen as negative if the rating is less than three stars.

After having a clear guideline, the first step of the analysis is to use VADER to evaluate each review sentiment grouped by the film names. For a review that has a VADER compound score greater than 0.35, it is treated as a positive review while a VADER score below 0 will get a review classified as negative. Finally, I determined

the VADER's accuracy by calculating the share of positive/negative reviews that are correctly identified by VADER.

Subsequently, I subset the corpus into two sub-corpora that each contains only positive/negative reviews based on rating, hoping to discover top features that are most frequently mentioned in the positive/negative review in each Nolan's film using tokenization. Except for following the regular tokenization procedures, I took some more steps to refine the analysis further. First, I removed possible stemmed worlds when tokenizing the review text. Then, I created a bag of words for both positive and negative reviews of Nolan's film by calculating the total word counts for each word across all the reviews. Since the bag of words ignores the semantic context of the review and concentrates primarily on the frequency of each word, I also tried n-gram modeling where I tokenized the bi-gram features in each of the review documents to offer more contextual information to the review text.

The next part of the study is to perform a binary classification to categorize reviews as favorable or unfavorable. Here, I used a simple logistic regression to classify the text sentiments and trace back to determine prediction accuracy. Finally, I perform a Z-test to determine the statistical significance of individual features. I then again use logistic regression to construct a model to make predictions by giving each feature a weight that pulls toward the class of positive or negative reviews, offering us insights about the features that are most likely to tip our classification model.

RESULT & DISCUSSION

- ## Review Sentimental Analysis using VADER & exploratory feature analysis using tokenization

As discussed in the methodology section above, I compute the VADER model accuracy by finding the union between reviews that are deemed positive/negative by VADER and the actual positive/negative review according to reviewers' rating. And the result is as followed:

Overall, out of 3200 pieces of review text, there are 2708 of them are positive based on the rating. Although VADER classifies 1299 pieces of reviews as positive, there are only 1148 of them are valid positive reviews (recognized by the film rating). Thus, VADER's accuracy in predicting the reviews' positive sentiment is about 42.4%.

The prediction outcome gets worse when looking at the VADER's accuracy in distinguishing negative sentiments. Out of a total of 223 negative reviews, only 86 of them are predicted correctly by VADER. Hence, its accuracy in predicting the reviews' negative sentiment is about 38.6%.

Just looking at the prediction rate, VADER is able to recognize positive reviews more effectively than negative ones by rightly picking up strong positive words such as "masterpiece", "brilliant", "perfect,". Nevertheless, it does not do an overall great job in predicting reviewers' sentiment, especially when it comes to negative sentiments tone. While part of the reason is due to the reviewers' informal and nonstructural language, it also reveals one shortcoming in VADER's underlying algorithm – the prediction is clearly swayed by certain words that have positive/negative tendencies. Often time, we will see VADER misidentify ambiguous emotion. Take positive reviews that have a low VADER compound score, for example, most of them contain words that express a mix of feelings like "cry," "tears," "break my heart," and even expletives. What is more, VADER does not seem to correctly detect a sense of loss or condolence expressed by reviewers. Oftentimes, when expressing a sense of loss to the actor who passed away in a film, instead of criticizing it, the reviews are actually demonstrating their appreciation to the film, memorizing the actor's repertoire. VADER, on the other hand, cannot detect this layer of sentiment from the reviewer. As a result, we see reviews like "I miss poor Robin Williams" and "UGH I MISS HEATH SO MUCH."

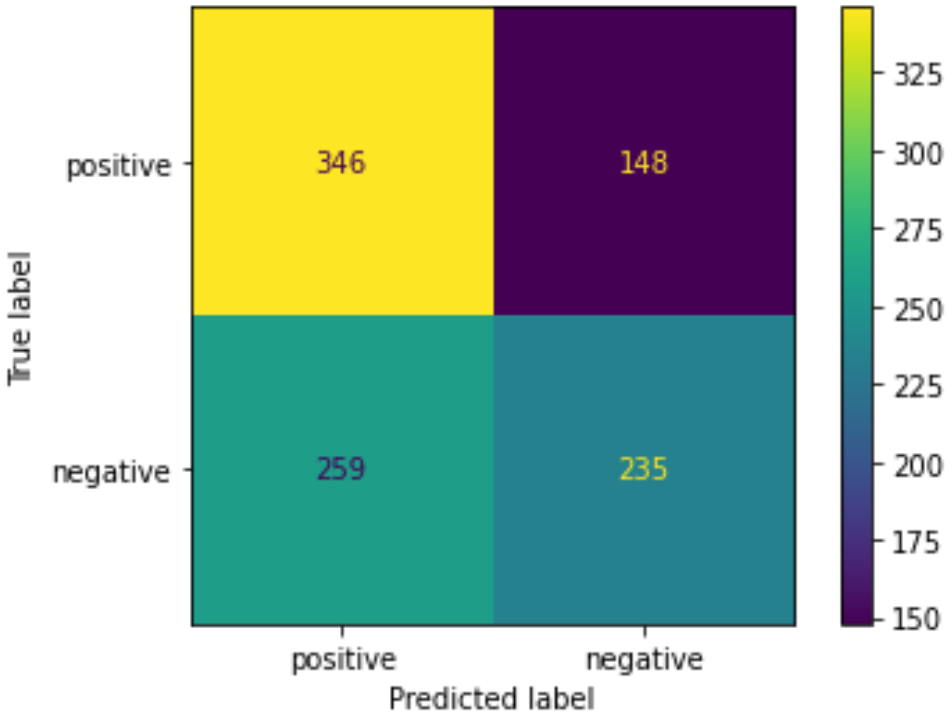
As for the next step, tokenization, I printed out the 10 most significant words ranked by their frequency for every Nolan's film inside the positive/negative review data frame. The outcome is quite satisfying. There are quite some words that can seemingly define the characteristics of a film, indicating spots in the film that are most welcomed/rejected by the viewers.

Aside from universal positive words like "masterpiece," "love," or "great" that can appear in any positive reviews, I am more interested in features that are more context-specific. For example, two most frequent bi-grams appearing in positive reviews of *Tenet* (2020) are "Elizabeth Debicki" and "Amalfi Coast" where the former is the main female protagonist and the latter is the actual filming locations. More interestingly, one of the top features in positive reviews of *The Prestig*e (2006) is Ziggy Stardust, a nickname for the brilliant artist – David Bowie. Therefore, for most of Nolan's films, places like the cast, the cinematography, and the filming location, are most likely to be appreciated and noticed by the audience, leaving a great impression throughout from the beginning to the end.

Depending on the film, things where the audience is not satisfied with also vary. Some people accuse that cinematography and narrative are confusing in Tenet, while others think The Prestige is eerie in terms of its cast. The most interesting one ought to be the occurrence of "paprika" in the negative review corpus of Inception. *Paprika* (2006) is a fantastic animation directed by Satoshi Kon. For years, critics have claimed that Nolan took heavy references from Paprika's when comping up with Inception's narrative. With such backlash, it is no wonder that "paprika" expressed an extreme sense of negative sentiment in reviews of Inception.

◆ Binary Classification

To explore other methodology to improve the analysis result, I tried binary classification models on various feature representations of the textual information in the reviews. The outcome, as expected, is much better than VADER's performance. Overall, the classification accuracy is around 59%. If we group the accuracy by sentiments, we will see the model has an accuracy of 70% in identifying positive reviews and about 50% in discerning negative reviews. Despite having better accuracy than VADER, both models fall short in singling out negative sentiments.



The final step of my analysis is to explore the significant features that are most likely to tip off the model that a given review text is positive or that another is negative. It turns out that the outcome is scintillating. There are two film titles that

appear in the list of words that distinguish positive reviews, namely The Dark Knight and Inception. It is possible that a substantiate amount of positive reviews' content is related to these two films, thus letting the model assign them with more weights. What is more, there are many actors/actresses' names appearing in the list, influencing the model's classification. For instance, Heath Ledger in *The Dark Knight*, Hugh Jackman in *The Prestige*, Cillian Murphy in *Inception*, all are distinct in positive reviews, suggesting that texts which mention these actors are more likely to be positive ones. On top of that, films like *Insomnia* (2002) and the actor Al Pacino have been recognized as salient features by the model in negative reviews, telling us Nolan's film or cast that are least favorable.

Conclusion & Next Step

From the results above, we can infer that for our problem statement, the Logistic Regression Model has a better performance in classifying sentiments compared to VADER. Apart from this, through tokenization and Z-test, we also get a glance at which Nolan's film, especially which part of the film, is most favored or disliked by the viewers. One peculiar thing to note is low accuracy when identifying negative reviews for both models. This might be because of the insufficient amount of training data (494) and the inconsistency between review texts and their rating of the film. Also, the low accuracy of the VADER score shows us that people have varied writing styles and VADER is not suited to data with high variance. One of the major improvements that can be incorporated as I move ahead in this project is to remove irrelevant reviews, reviews that are inconsistent with the rating, before training the classifiers. Another point of improvement can be to model this problem as a multi-class classification problem where we classify the sentiments of reviewers in more than binary fashion like "Happy", "Bored", "Impressed", etc. This problem can be further remodeled as a cluster problem where we can group the degree of affinity for the movie instead of just like/dislike.