Sentiment Analysis with Movie Reviews

Ranako Holder and Justin Lim

Abstract

The primary goal of sentiment analysis is to identify and extract the emotion a piece of text is trying to convey. A text's emotion, sentiment polarity, is categorized as either negative, neutral, or positive. This paper presents our approach to determining polarity for a given movie review by breaking down reviews into single sentences and assigning polarity scores to the individual units. The unit scores were collected and used to calculate the arithmetic mean representing the polarity of the review as a whole. Experimental results demonstrate that our approach is less accurate than a more focused

1 Introduction

Oftentimes when making decisions in today's society it is normal to get a sense of analysis paralysis due to the abundance of choice we are presented with. This can be especially true with choices regarding entertainment. With technology such as TikTok, iPhones, and the internet as a whole, it can be difficult to sit down and just choose a movie to watch. As a result of this, the ultimate goal for our experiment is to construct a movie review system based on sentiment analysis in NLTK.

The general goal of this project is to determine the correlation between how positive movie reviews are compared to how lucrative they said movies were. This will prove to be an interesting project because it will provide the average consumer with a space to determine whether or not a movie is worth their time and money prior to even getting to the theater. This could also provide movie directors with important insight into the way that reviews impact their sales.

2 Previous Research

Sentiment analysis has been extensively investigated regarding how to analyze and extract opinions from texts with the purpose of interpreting the viewpoints of other individuals. It has become a critical tool for producers to interpret language from consumers and improve their products. This technique has even found its way to the Hollywood scene and allowed statisticians to analyze the public's thoughts on motion pictures.

2.1 Early Approaches to Sentiment Analysis

Early research in sentiment analysis focused on developing rule and lexicon-based approaches. To categorize texts as either positive, negative, or neutral, researchers often used predetermined linguistic rules to help aid their testing. These methods largely depended on criteria such as the context of words, the emotions they convey, and grammatical patterns to identify the tone of a particular text. While these approaches offered a strong foundation, they frequently had trouble with the nuances of language such as expressing emotions, irony, and the contextual interpretation of emotions.

2.2 Supervised Learning Approaches

With information like labeled datasets becoming more widely available, supervised learning approaches became more frequently used in sentiment analysis. On labeled movie review datasets, researchers trained classifiers using machine learning approaches including Support Vector Machines (SVM), Naive Bayes, and Maximum Entropy models (Pang et al., 2002). These classifiers quickly learned how to recognize patterns in sentiment and predict the polarity of future reviews.

2.3 Deep Learning Approaches

In recent years, deep learning methods have revolutionized sentiment analysis by leveraging the power of neural networks. Recurrent neural networks, in particular long-short term memory (LSTM) networks have been very successful in modeling sequential data, specifically, movie reviews (Socher et al., 2013). The purpose of these models is to capture the contextual dependencies between words. Convolutional neural networks have also been used for sentiment analysis tasks helping prove their efficiently in identifying trends.

2.4 Cross-Domain and Multimodal Sentiment Analysis

Researchers have looked into cross-domain sentiment analysis, which entails training sentiment classifiers on one domain (such as product reviews) and using them on another domain (such as movie reviews), in addition to looking at movie reviews in isolation. Additionally, multimodal data integration has been investigated to improve sentiment analysis accuracy and offer a more thorough knowledge of movie feelings. For instance, text reviews can be combined with visual clues from movie posters or video snippets.

In general, earlier work on sentiment analysis of movie reviews has advanced from rule-based and lexicon-based methods to more complex supervised learning, deep learning, and transfer learning techniques. These developments have enhanced sentiment classification performance and opened the door to more research into multimodal, cross-domain, fine-grained sentiment analysis in the movie industry.

3 Data, Method

The data set used in this experiment was the Sentiment Polarity Dataset 2.0 collected by Bo Pang and Lilian Lee and redistributed by NLTK. It contains 1000 positive and 1000 negative annotated reviews. Below is a table representing basic information about our dataset.

Data Statistic Summary Table

Number of Units	64,721
Unit Label Distribution	32,938 Positive 31,783 Negative
Number of Tokens	46,313

Among the six column titles used to label the dataset, we primarily focused on three titles: “html_id”, “text”, and “tag”. The “html_id” column labeled units in the “text” column that belonged to the same review. The tag column consisted of the human annotation on the review polarity.

In terms of preprocessing, for this dataset, we first converted all text from upper to lower case to avoid having two of the same words be considered two separate tokens. Next, we broke the text into unique tokens and then removed the stop words from the text.

4 Baseline

For our initial approach, we used a Python library called TextBlob provided by NLTK. Textblob takes a text, in our case a sentence(unit), and returns its respective polarity score as a float within the range [-1, 1]. Every unit belonging to the same review was added to a value we called “review_score”. The “review_score” is then averaged by the number of sentences used, returning the arithmetic average representing the polarity of the review.

We chose this approach because the arithmetic average is a very well-known metric. It is easy to calculate and understand its significance. Furthermore, it provides a broad idea for the review’s sentiment, and readers not as familiar with sentiment analysis or NLP can follow along.

5 Evaluation Metrics

The baseline and final evaluation metrics were an accuracy score. It aimed to answer the questions: Did our experiment's results match the annotated control results? What percentage of our results matched correctly?

(# of correct matches / # total reviews) * 100

6 Results

6.1 Baseline Results

The baseline was our first attempt at using Textblob, and we were aiming to test the viability of approach. To achieve this, we chose three movie reviews to test. We were able to receive three positive review polarity scores, floats in the range (0, 1], using our approach and when compared to their respective expected polarity values, we achieved a100% accuracy.

Overall, we were happy with the results of our baseline experiment. We managed to integrate Textblob into our arithmetic mean focused algorithm. While this was a great first step, there were glaring issues to the current baseline approach. The first issue became apparent during the implementation. We had to manually find the ranges of units within the "text" data column to find our selected testing reviews and then implement our approach.

This was very tedious and impossible to do if we wanted to use the entire dataset. As a result of this inefficiency, we used an extremely small sample size resulting in a skewed accuracy score. Moving forward we knew that our approach worked, but we had to optimize how we separated unique movie reviews.

6.2 Final Results

To improve from our baseline approach and results, we utilized the "html_id" column tag. To reiterate, the "html_id" tag uniquely identifies units that belong to the same review. For example, say a review with "html_id" = 12345 is broken up into six sentences. It would hard to differentiate where a review begins and end just by looking at the units within the "text" column. But by looking at "html_id" column, this becomes easier as the six sentences would correspond to a sequence of six 12345 "html_id" tags.

With our new grouping strategy complete, we applied our experimental approach to the entire dataset. Storing the data for comparison was done using a dictionary mapping "html_id" keys to their respective polarity scores. The next step was to covert the numerical polarity scores and reflect our approach's prediction to the movie review's polarity.

If
–1 < score < 0 then polarity = "neg"
Else
0 < score < 1 then polarity = "pos"

We chose to make these hard cutoffs because the dataset did not include neutral reviews, and realistically a completely neutral review would be hard to generate and not as informative.

7 Conclusion

After the completion of our experiment, we applied our evaluation metric and received a 61.1% accuracy. This is not a great score as it is slightly better than flipping a coin to guess polarity. We chose an approach that reflected our prior experience with sentiment analysis and in that regard, we believe we succeeded in creating a good entry level approach and implementation. Some final observations noticed was that the majority of inaccurate predictions occurred with negative reviews. We believe this is due to how we obtained polarity scores and its inability to account for context; Sentences before and after the target sentence. Additionally, it was noted that arithmetic mean is not as solid as a metric as previously believed.

What the mean represents is what every sentence would have scored if they all scored the same. Implying that every sentence contributes equally to overall polarity score. But in reality, not all sentences are equal in terms of polarity. A statement such as,

"I went to the movies"

would not convey the same sentiment as,

"The movie was terrible".

The polarity score using our mean approach would decrease the overall score when realistically, the polarity score for,

"The movie is terrible"

should be more heavily weighted. After reflecting on the process used during this experiment and our final evaluation, there certainly are many ways we can make further improvements.

8 Future Work

One possible improvement based on our results and final observations would be to return the polarity of the entire review instead of breaking it into sentences. As stated previously in our conclusion, our approach does not take into consideration context and by extension sarcasm. Text with negative sentiment can often contain sarcastic portions. While human beings can detect it in vocal speech through tone, pitch, and intuition, it is more difficult in written text. A sentence such as,

"That was the greatest movie I have ever seen!"

can be interpreted in two different sentiments. Taking this sentence by itself would produce a false polarity and negatively affect our approach's predictions.

The direction for future research and experimentation would be the addition of a text filter for less "important" sentences. This can be done using existing filters. Textblob provides a multitude of tools, such as part-of-speech tagging or noun-phrase extraction. But custom filters can also be created and tested for viability.

References

Pang, B. (2002) *Thumbs up? Sentiment Classification using Machine Learning Techniques*, *cs.cornell.edu*. Available at: https://www.cs.cornell.edu/home/llee/papers/sentiment.pdf (Accessed: 15 May 2023).

Socher, R. *et al.* (2013) *Recursive deep models for semantic compositionality over a sentiment treebank*, *ACL Anthology*. Available at: https://aclanthology.org/D13-1170/ (Accessed: 15 May 2023).

Group member's contributions

Ranako Holder- Data research/collection and cleaning

Justin Lim- Approach baseline and implementation