INTERNATIONAL BURCH UNIVERSITY

FACULTY OF ENGINEERING,
NATURAL AND MEDICAL SCIENCES
DEPARTMENT OF INFORMATION TECHNOLOGY

MOVIE REVIEW SENTIMENT ANALYZER

PROJECT PAPER

WRITTEN BY: AMINA HUKIĆ

SARAJEVO
January 2024

Summary

This project is going to look at movie reviews on the RottenTomatoes website and use a sentiment analyzer to try and guess the overall rating of the movie through them. The goal is to gain insight into the numerical rating of the movie through the words and tone of the review. We will be using data collection, text analysis and prediction. We will take the review text, handle it appropriately, and use it to create a sentiment number which we will then compare to the movie rating.

Introduction

Movie websites have become a big part of the cinema industry, with several very well-known websites being used to gain knowledge about the quality of movies one might be interested in. One of the biggest websites is the RottenTomatoes site, where there are countless movies that have been reviewed by both movie critics and a general audience.

For each movie, there is a critic score and an audience score, as well as a review section. The review sections contain written reviews and an individual score associated with each review. The score can be positive or negative.

The reviews shown are snippets of opinions, and we will try to use them to accurately guess the score of the movie. The scores are showcased in percentages so we will try and compare the two.

For the reviews, we will be using a sentiment analyzer to get an overall score between 0.0 and 1.0. We will be using the NLTK library and beautifulSoup to do this project. The NLTK library further has a SentimentIntensityAnalyzer which we will be using to analyze the comments.

Due to the fact that a lot of movies do not have audience reviews, we will be using critics reviews and the critics rating of a movie.

Literature review

NLTK is a Python library for working with human language data. It provides easy-to-use interfaces to lexical resources, along with a range of text-processing libraries for classification, tokenization, stemming, tagging, parsing, etc.

It was developed by Steven Bird and Edward Loper at the University of Pennsylvania and initially released in 2001. It has since become a widely used tool in natural language processing (NLP) and machine learning communities.

As of recent, it is still widely used for teaching and development. While NLTK itself may not be actively developed at the same pace as some newer libraries, its extensive set of resources and tools makes it a valuable asset in educational settings and practical applications.

Beautiful Soup is another Python library for pulling data out of HTML and XML files. It provides Pythonic idioms for iterating, searching, and modifying the parse tree, making it easy to extract and manipulate data from web pages.

It was created by Leonard Richardson. The first version, Beautiful Soup 3, was released in 2008. Beautiful Soup 4, a complete rewrite with new features and improvements, was released in 2012. It has a GitHub repository on which we can see the most recent changes and updates.

Sentiment analyzers, or sentiment analysis tools, are applications or algorithms that use natural language processing and machine learning techniques to determine the sentiment or emotion expressed in a piece of text. The goal is to understand whether the expressed sentiment is positive, negative, or neutral.

Sentiment analysis has evolved over the years with advancements in NLP and machine learning. Early approaches focused on rule-based systems, but modern sentiment analyzers often employ machine learning models, including deep learning techniques.

The development of large labelled datasets and improvements in algorithms have contributed to the effectiveness of sentiment analysis in various applications, such as social media monitoring and customer feedback analysis.

The sentiment analyzer we are using is from the NLTK library, and part of the VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis tool.
The sentiment analyzer used by NLTK (Natural Language Toolkit) is part of the VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis tool. VADER is specifically designed for analyzing sentiments in text data, and it's a pre-built, lexicon-based sentiment analysis tool.

Here are some key points about VADER sentiment analysis tool:

- it uses a pre-built lexicon (dictionary) that contains words scored based on their sentiment polarity (positive, negative, or neutral) and intensity

- the analyzer assigns each word in the text a polarity score ranging from -1 (most negative) to 1 (most positive)
- it takes into account the intensity of sentiment words and handles negations to provide a more accurate analysis
- it is designed to recognize and interpret emoticons in the text, considering them in the overall analysis
- in addition to the individual positive, negative, and neutral scores, VADER provides a compound score, which represents the overall sentiment of the text. This score ranges from -1 to 1, with -1 indicating the most negative sentiment, 1 indicating the most positive sentiment, and 0 indicating a neutral sentiment

All of these make it a impressively well built tool, with a very easy output that in simple terms expresses the sentiment of a given text.

VADER is well-suited for analyzing short and informal text, which makes it neigh perfect for our analysis of short and simple review texts.

Hypothesis / Research questions

When starting, our basic question was:

How accurately can we guess a movie's rating through the reviews?

While sentiment analyzers are getting more and more accurate and complex, it is still very difficult for computers to guess the meaning of words and accurately read the tone.

Hypothesis:

There will be a 0.25 average difference between the analysis and the movie rating.

Since the reviews are relatively short and sometimes rather vague, my assumption is that while the sentiment analyzer will be mostly accurate, due to the small data given there will be significant swayings between the guess and actual rating.

The sentiment analyzer will probably struggle to pick up on certain nuances in certain phrasing, words used, etc. The numbers given by the sentiment analyzer will probably be largely neutral to slightly positive, while some movie ratings will go very sharply in either direction.

Methodology

We will be using a rather simple method similar to what we have done in NLP classes so far. We will gather data through reviews, make a sentiment analyzer, and then analyze the data.

We will then compare the result of the analyzer and the actual rating, and take the difference between the two (as a positive integer). We will also take note of the largest and smallest difference, purely for additional context.

We will repeat that with several different movies in various genres, with various ratings, and then look at the average of the differences and see the accuracy of the analyzer.

The first part of our code is the treatment of data: we take the review text of a movie, analyze each review and return an average of the result.

Then, we will gather data. We have three separate datasets: firstly a mix of good and bad ratings, then bad ratings (less than 50%) and the good ratings (more than 50%).

Then each movie is being treated: it gets analyzed and we add the sentiment analyzer result and movie rating difference to a dataset.

In the end we will have three main points: average difference, positive review difference, and negative review difference.

Data and findings

When gathering data, we found a few issues.

First one being a very large difference between the sentiment analyzer resutlts and the actual rating. The issue turned out to come from the fact that the sentiment analyzer rates a text from -1.0 to 1.0, being negative to positive. The actual movie ratings go from 0.0 to 1.0, so the difference was made larger by that disrepancy.

A code block was added that turns a sentiment into a number in the appropriate range by adding 1.0 and dividing by two. While that may not have been the absolute best option, it was a satisfying fix for the timebeing.

We have a total of 39 reviews in our dataset, and as previously stated we ran three separate trials: mixture of reviews, positive and negative.

Here are the three main points:

- Average difference: 0.20234292682926827
- Positive difference: 0.17262010869565217
- Negative difference: 0.2596040625

This falls within our hypothesis, with a 20% difference between the review analysis and the movie rating.

Conclusion

So, we can conclude that our initial hypothesis of a 0.25 difference was fairly accurate. The accuracy of the sentiment analyzer is impressive considering the difficulty of dealing with written text, and the complexities of opinionated texts.

If we look at the sentiments, we can see that they tend to be slightly lower than the actual review, but we can see an interesting difference between positively and negatively rated movies.

When it comes to positively rated movies, the average difference was 17%. It is slightly better than the average difference (mixed reviews).

The sentiment analyzer tends to give lower number to the reviews than the actual rating of the movie. The highest number was around 0.82, while the highest rating was around 0.95. We could draw a conclusion from there that reviews are not capturing the full enjoyment of the writer, and that the words are not “strong” enough to express their apparent opinion.

That could also be explained by the fact that often times even with a large enjoyment of a movie one might not go further than complimenting it with words such as “good”, “enjoyable”, or at most “great” and “excellent”.

But the more interesting data comes from the negatively rated reviews. The difference is around 26%, and that comes from the sentiment analyzer usually rating the review higher than the movie rating actually is.

The lowest number given was a 0.40, while the lowest rating was a 0.07. This seems to imply that the reviews struggle even more to show the pure dislike towards the movie, apparently failing to use strong enough language to convey their true opinion.

We can see that while there are differences regarding the positive ratings, they are much much larger for the negative ratings. No review seems to have truly encompassed the true low of the critics opinion.

This is interesting considering that usually people find it so easy to critique things they dislike, and negative language can come easier than praises.

One of the reasons review seem to be so much more positive could be that they are professional critic reviews, not audience reviews. Therefore the words used will always be less harsh than they could be, and the dislikes a critic might have will be expressed much gentler than they would otherwise be, with more explanations as to why and less “mindless” hate words.

The main takeaway from these results is that the written opinions of people can differ in large part to a numeric rating system, since it is usually difficult to express a complex opinion in one small number.

References

Wikipedia (used to gather information about NLTK, BeautifulSoup and sentiment analyzers): https://www.wikipedia.org

Website link: https://www.rottentomatoes.com

NLTK sources: https://www.nltk.org

BeautifulSoup GitHub: https://github.com/wention/BeautifulSoup4