University of New Haven
Tagliatelle college of Engineering

Masters In Data Science

Course: DSCI 6004 Natural Language Processing 03

Title: Movie Review Analysis Using Natural Language Processing

By

Rajesh Ponnaganti

Under the guidance of

Prof. Khaled Sayed, Ph.D.

Abstract:

The rapid growth of Internet-based applications, such as social media platforms and blogs, has resulted in comments and reviews concerning day-to-day activities.

Word2Vec is a deep learning inspired method that focuses on the meaning of words. Word2Vec attempts to understand meaning and semantic relationships among words. It works in a way that is similar to deep approaches, such as recurrent neural nets or deep neural nets, but is computationally more efficient. Sentiment analysis is the process of gathering and analysing people's opinions, thoughts, and impressions regarding various topics, products, subjects, and services. People's opinions can be beneficial to corporations, governments, and individuals for collecting information and making decisions based on opinion. However, the sentiment analysis and evaluation procedure face numerous challenges. These challenges create impediments to accurately interpreting sentiments and determining the appropriate sentiment polarity. Sentiment analysis identifies and extracts subjective information from the text using natural language processing and text mining. This article discusses a complete overview of the method for completing this task as well as the applications of sentiment analysis. Then, it evaluates, compares, and investigates the approaches used to gain a comprehensive understanding of their advantages and disadvantages.

Introduction

Sentiment analysis has gained widespread acceptance in recent years, not just among researchers but also among businesses, governments, and organizations. The growing popularity of the Internet has lifted the web to the rank of the principal source of universal information. Lots of users use various online resources to express their views and opinions. To constantly monitor public opinion and aid decision-making, we must employ user-generated data to analyse it automatically. As a result, sentiment analysis has increased its popularity across research communities in recent years. Sentiment analysis is also called as Opinion analysis or Opinion mining. We have seen a recent growth in the sentiment analysis task. To organize all the aspects of the reviews they selected a number of lexicons from several dictionaries. Sentiment analysis is previously being applied in various domains ranging from hotels to airlines and healthcare to the stock market Sentiment Analysis, contain the Data Collection, Feature Extraction, and Feature Selection Method, explaining all the steps from data extraction to various task of Sentiment Analysis, Sect. contain General Methodology for Sentiment Analysis and its Summary

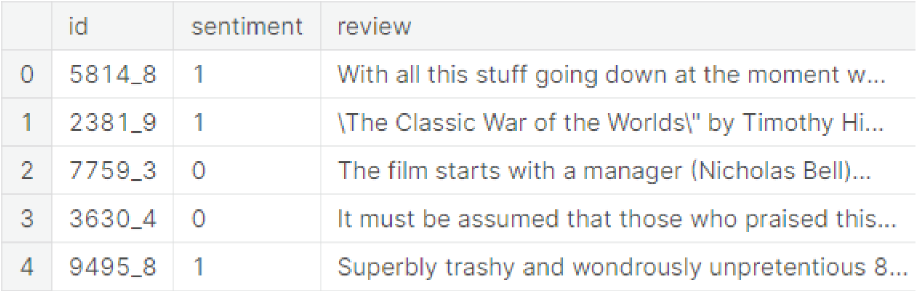
Analysis:

Data Exploration

In this project we will be working on a large dataset of movie reviews for the Internet Movie Database (IMDb) which has collected by Maas et al. The movie dataset consists of 50,000 movie reviews that are labelled as either positive or negative; positive in the sense that a movie was rated with more than five starts on IMDb, and negative means movies rated below five stars on IMDb, that neutral reviews are not included (outliers) in the dataset. These reviews are used to study the user opinions by implementing Logistic Regression and Stochastic Gradient Descent models for classification of data.

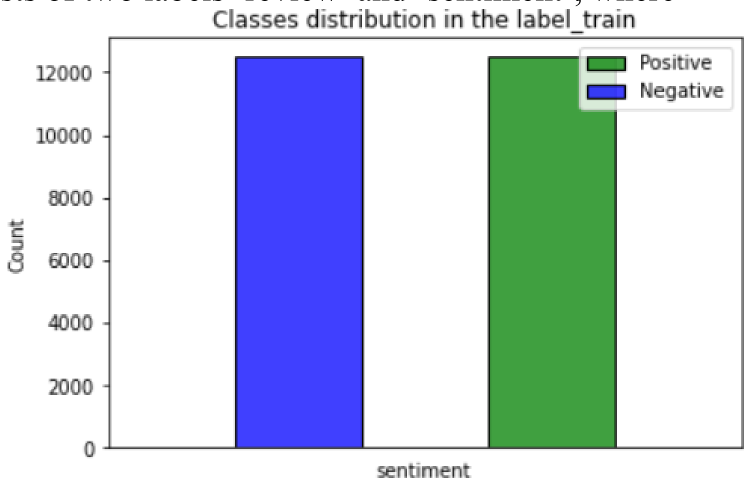
Exploratory Visualization

The table below shows the sample data of our dataset. The download data is in the form of text files and divided and directed to respective polarity subdirectories. The present form data is complex to explore, so there is a need to convert whole text files into a single CSV file using DataFrame. The converted data to be shuffled as the previous data is in sorted form. By using permutation methods the DataFrame data is shuffled and stored into a CSV file.



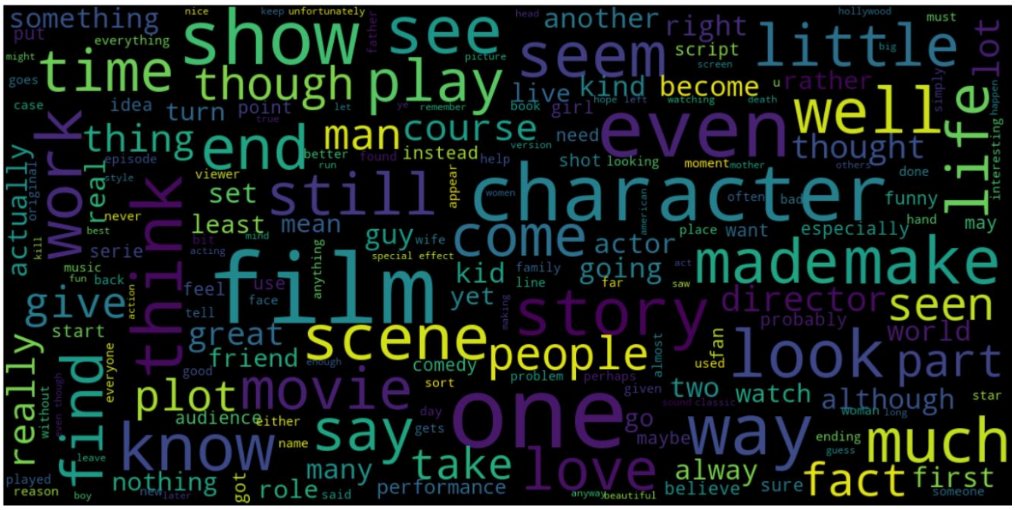
	id	sentiment	review
0	5814_8	1	With all this stuff going down at the moment w...
1	2381_9	1	\The Classic War of the Worlds\" by Timothy Hi...
2	7759_3	0	The film starts with a manager (Nicholas Bell)...
3	3630_4	0	It must be assumed that those who praised this...
4	9495_8	1	Superbly trashy and wondrously unpretentious 8...

The generated CSV file consists of two labels ‘review’ and ‘sentiment’, where the ‘review’ column consists of reviews of different users and ‘sentiment’ column consists of polarity of the user opinion.
The plot above shows the polarity distribution of reviews, that there are total 50,000 data points, in which 25,000 data points are positive reviews and the remaining 25,000 reviews are of negative reviews.



Word Cloud:

A word cloud is a popular visualization technique used in Natural Language Processing (NLP) to represent the most frequently occurring words in a text corpus. It provides a visual summary of the prominent words in a collection of texts, with the size of each word indicating its frequency.



Metrics:

The metrics used in the project are:

1. **Accuracy** is a common metric for binary classification test. The accuracy is the proportion of true results among the total number of cases examined.

   $$accuracy = (true\ postives + true\ negatives\ )/\ dataset\ size$$

   As my dataset is balanced, I used accuracy as a main evaluation metric to know how accurately my model will predict the output result.
2. **Recall** is the fraction of relevant instances that have been retrieved over total relevant instances.

   $$Recall = true\ positives/(\ true\ positives + false\ negatives)$$

   I used recall metric to know the percentage of true outputs of my regression model compared to the other two classification models used in the project.
3. **Precision** is the fraction of relevant instances among the retrieved instances.

   $$Precision = true\ positives\ /(true\ positives + false\ positives)$$

   I calculated this metric to know the exactness of my model in predicting polarity of reviews.
4. **F-Score** is a measure that combines precision and recall is the harmonic mean of precision and recall.

   $$F = 2\ .\ precision\ .\ recall\ /(precision + recall\ )$$

   The main reason behind using F-score, recall and precision metrics is for comparing the performance of Logistic Regression model with SVM and Naïve

Bayes models, that to convey why I used this model for my classification problem. As the basic rule to choose a best classification model is the recall and f-scores of the model should be higher and the precision value should be lower.

Algorithms and Techniques:

Bag of Words: I preferred Bag of Words technique to process my text into vectors as raw text cannot be fed into my model. I implemented tokenizing strings and giving an integer id and counting the occurrences of tokens in each document finally normalizing and weighting with diminishing important tokens that occur in most of the documents.
Tokenization: Break down the text into individual words or tokens. This process involves removing punctuation and splitting the text into words.
Vocabulary Building: Create a vocabulary of all unique words present in the entire corpus (collection of documents).
Vectorization: Represent each document as a numerical vector, where each element of the vector corresponds to the frequency of a word from the vocabulary in the document.

While Bag of Words is a simple and efficient way to represent text data, it has some limitations. It doesn't consider the semantic meaning of words or the context in which they appear. More advanced techniques like TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings (e.g., Word2Vec, GloVe) address some of these limitations by capturing more nuanced relationships between words.

Methodology

Data Preprocessing

Text cleaning is a crucial step in Natural Language Processing (NLP) that involves preparing raw text data for analysis. The goal is to remove noise, irrelevant information, and inconsistencies from the text, making it more suitable for further processing and analysis. Here are some common text cleaning techniques in NLP:

Lowercasing:

- Convert all text to lowercase. This ensures uniformity and helps in treating words with different cases as the same.

Tokenization:

- Break the text into individual words or tokens. This step is fundamental for many NLP tasks.

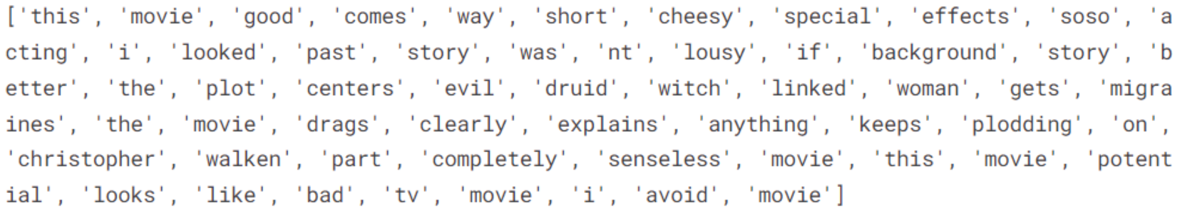
Removing Punctuation:

- Remove unnecessary punctuation marks from the text.

Removing Stop words:

- Remove common words (stop words) that don't contribute much to the meaning of the text.

After removing stop words the data will look like below without any words with noise.



```
['this', 'movie', 'good', 'comes', 'way', 'short', 'cheesy', 'special', 'effects', 'soso', 'a
cting', 'i', 'looked', 'past', 'story', 'was', 'nt', 'lousy', 'if', 'background', 'story', 'b
etter', 'the', 'plot', 'centers', 'evil', 'druid', 'witch', 'linked', 'woman', 'gets', 'migra
ines', 'the', 'movie', 'drags', 'clearly', 'explains', 'anything', 'keeps', 'plodding', 'on',
'christopher', 'walken', 'part', 'completely', 'senseless', 'movie', 'this', 'movie', 'potent
ial', 'looks', 'like', 'bad', 'tv', 'movie', 'i', 'avoid', 'movie']
```

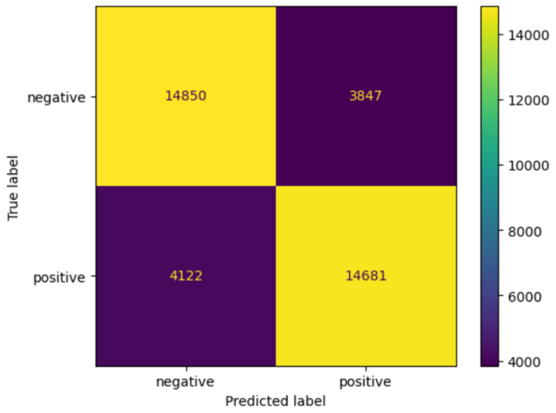
Stemming and Lemmatization:

- Reduce words to their base or root form to normalize variations.

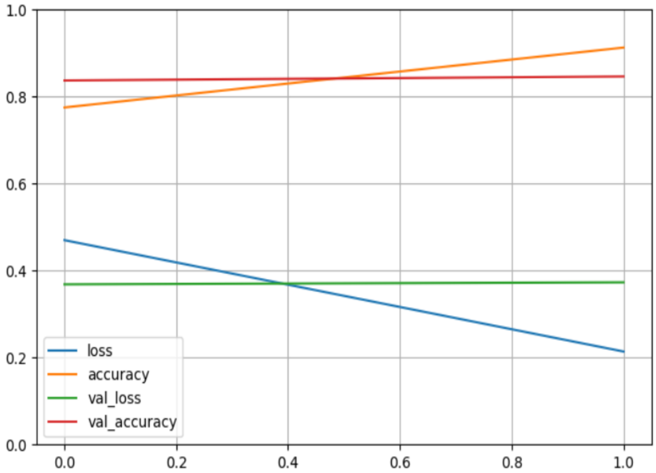
Model Evaluation and Validation

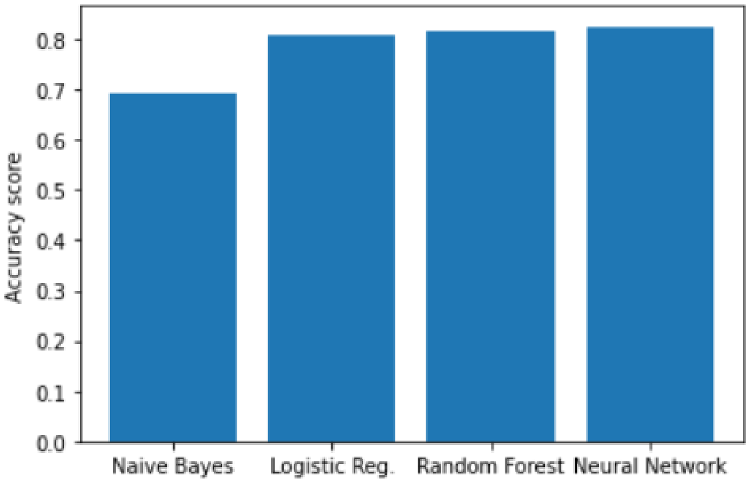
Confusion matrix:

Particularly useful for classification problems, a confusion matrix provides a detailed breakdown of true positive, true negative, false positive, and false negative predictions.



The evaluation part of our work makes use of the Kaggle evaluator. The evaluation consists of 25,000 unlabeled reviews that we classify with each of our models. We submit our results to Kaggle and we receive an AUC value for the corresponding ROC curve. Our evaluation is based on the AUC value, and hence comparison with the reported accuracies in the literature may not be vis-a-vis. For our baseline, we used the bag of words model. We limited our vocabulary to 5000 and used 100 trees in random forest, and obtained a baseline of 84.44. Our evaluation consists of two types of analysis. In the first part, we explore the effect of the parameters of word2vec in the vanilla case (averaging the word vectors). The second part of the evaluation consists of evaluating different models with the same set of parameters used across them.





With respect to clustering and averaging word vectors, we find that clustering performs a bit better. We attribute this to the information gain about the sentiments from the clusters. Our weighted averaging of the word vectors seem to validate our thesis about the placement of sentiments in the paragraph. While the word2vec model is inferior to the paragraph vector model, the weights of the word vectors make up for some of the information loss during averaging.

Conclusion and Future Work:

We observed that the word2vec models significantly outperform the vanilla bag-of-words model. We observe that the variation of the parameters of word2vec have sometimes surprising outcomes. While the AUC increases with the number of trees, it falls sharply as we increase the minimum frequency of words. Amongst the word2vec based models, SoftMax provides the best form of classification, while weighted averaging of the words based on their position, gives a significant boost to the accuracy and comes very close to the state of the art.

We have a few recommendations for future work. We disregard non-alphabet characters in our analysis, but it may be useful to also take those into accounts. For instance, emojis and a repetition of punctuation (e.g., multiple exclamation marks) may infer the sentiment of the review significantly. Other classification methods may also be examined, such as regression and k-nearest neighbour.

References

[1] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. Found. Trends Inf. Retr., 2(1- 2):1-135.

[2] Kaggle competition : Bag of Words Meets Bags of Popcorn

[3] Mikolov, Tomas, et al. ”Efficient estimation of word representations in vector space.” arXiv preprint arXiv:1301.3781 (2013).

[4] Stanford sentiment analysis

[5] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. ”Learning Word Vectors for Sentiment Analysis.” The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).