Sentiment Analysis on Movie Reviews

Gurpal Singh

CSE Department

Nirma University

Ahmedabad, India

Abstract:

Movie reviews help users decide if the movie is worth their time. A summary of all reviews for a movie can help users make this decision by not wasting their time reading all reviews. Movie-rating websites are often used by critics to post comments and rate movies which help viewers decide if the movie is worth watching. Sentiment analysis can determine the attitude of critics depending on their reviews. Sentiment analysis of a movie review can rate how positive or negative a movie review is and hence the overall rating for a movie. Therefore, the process of understanding if a review is positive or negative can be automated as the machine learns through training and testing the data.

Keywords: Random-forest, Sentiment analysis, KNN, Naïve Bayes, LSTM, Movie Reviews

Introduction:

Humans are subjective creatures and their opinions are important because they reflect their satisfaction with products, services and available technologies having the ability to interact with people on its level has many advantages for information systems; like enhancing products quality, adjusting marketing and business strategies, improving customer services, managing crisis, and monitoring-performances.

A movie review is a commentary reflecting its writers' opinion a couple of certain movie and criticizing it positively or negatively, which enables everyone to know the idea of that movie and make the choice whether to watch it or not. A movie review can affect the full crew who worked on it. A study illustrates that in some cases, the success or the failure of a movie depends on its reviews. Therefore, a significant challenge is to be able to classify movies reviews to capture, retrieve, quantify and analyze watchers more effectively
Movie reviews classification into positive or negative reviews is connected with words occurrences from the reviews text, and weather those words are used before in an exceedingly positive or a negative context. These factors help enhance the review understanding process using Sentiment Analysis where SA has become the gateway to understanding consumer needs.

There are following phases of Sentiment Analysis:

1) Pre-Processing Phase: The data is first cleaned to reduce noise.
2) Feature Extraction: A token is given to the keywords and this token is now put under analysis.
3) Classification Phase: Based on different algorithms these keywords are put under certain category.

About Dataset:

IMDB dataset having 50K movie reviews for natural language processing or Text analytics. This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We have used around 10% of training data 5% of validation data and 5% of test data. So, predict the number of positive and negative reviews using either classification or deep learning algorithms. For more dataset information, please go through the following-link:

http://ai.stanford.edu/~amaas/data/sentiment/

Dataset Reading:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
...	...	...
49995	I thought this movie did a down right good job...	positive
49996	Bad plot, bad dialogue, bad acting, idiotic di...	negative
49997	I am a Catholic taught in parochial elementary...	negative
49998	I'm going to have to disagree with the previou...	negative
49999	No one expects the Star Trek movies to be high...	negative

50000 rows × 2 columns

Data Pre-processing:


```
positive    25000
negative    25000
Name: sentiment, dtype: int64
```

As seen in the plot we can say that in our dataset there are 25k positive and 25k negative sentiments.

1) Case: In data pre-processing first we need to convert the data into one case(upper/lower). There is a common approach to lowercasing everything for the sake of simplicity. It helps to maintain the consistency flow during the NLP tasks and text mining. The **lower()** function makes the whole process quite straightforward.

2) HTML tag should be removed because it is not related with the sentiment prediction. The web generates tons of text data and this text might have HTML tags in it. These HTML tags do not add any value to text data and only enable proper browser rendering.

3) **Removing URLs** is also important. URLs (or Uniform Resource Locators) in a text are references to a location on the web, but do not provide any additional information. We thus, remove these too using the library named re, which provides regular expression matching operations. We take our sample text and analyse each word, removing words or strings starting with https or http.

4) Removing punctuations: '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'

Punctuation should be removed because they do not contribute to predict the sentiments.

5) Chat-Words Conversion: in reviews most of the times people use short words which need to be replaced with full form for proper sentiment analysis.

6) textBlob(incorrect text): sometimes because of speed typing people do mistakes in spelling and this need to be corrected for sentiment analysis in python **TextBlob module** is useful for this.

7) Removing Stopwords: English is one of the most common languages, especially in the world of social media. For instance, "a," "our," "for," "in," etc. are in the set of most commonly used words. Removing these words help the model to consider only key features. These words also don't carry much information. By eliminating them, data scientists can focus on the important words. By using nltk module we can do this task.

8) removing special characters and digits is more important because they not contributing for sentiment analysis and need to be removed.

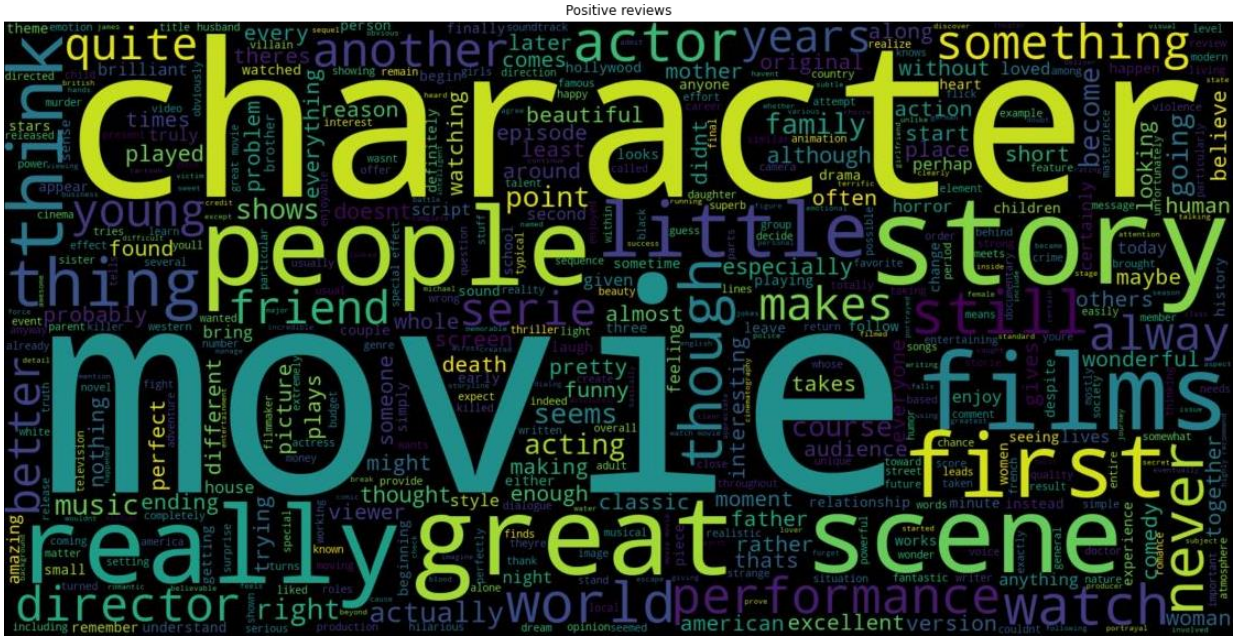
9) Stemming: There are many variations of words that do not bring any new information and create redundancy, ultimately bringing ambiguity when training machine learning models for predictions. Take "He likes to walk" and "He likes walking," for example. Both have the same meaning, so the stemming function will remove the suffix and convert "walking" to "walk." The example in this guide uses the **PorterStemmer module** to conduct the process. You can use the snowball module for different languages.

Word cloud after preprocessing:

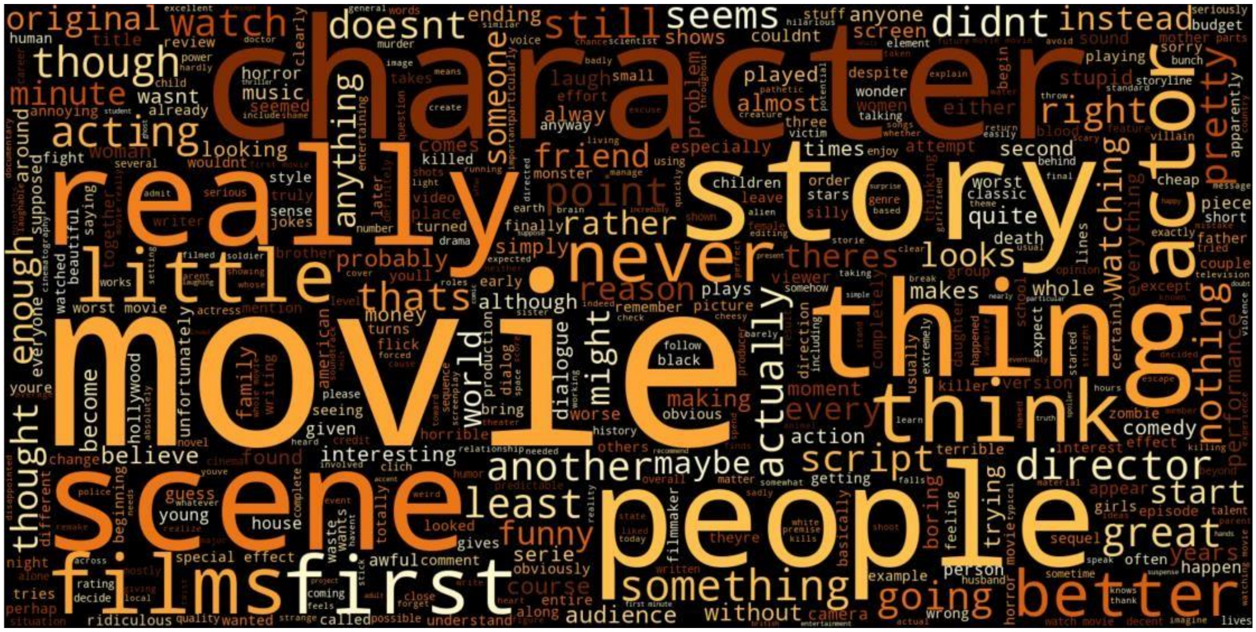
Word Clouds came out to be a game-changer visualization technique for understanding and determining patterns and evolving trends. Whether to discover the political agendas of aspiring election candidates of a country or to analyze the customer reviews on the recently launched product, one can get a visual representation by plotting the Word Cloud.

Word Clouds are widely being used to analyze the feedback received from end-users like from people who have watched the movie. Word Cloud would play a key role in getting top keywords among the feedback.

Word Cloud of Positive reviews



Word Cloud of Negative reviews



Problems:

Sentiment analysis is tough because a same topic can be expressed in different ways. Also the words used to express a positive sentiment would be negative in other statements. The movie reviews posted on the inter-net are unstructured form of grammar and expressing opinions on a topic are never standardized, one person's appreciation may differ from others.

In this section we have described the problem statement on Sentiment Analysis of Movie Reviews:

1. Extracting Sentiment Words - It is the heart of sentiment analysis; all the review statement contains sentiment words which have a major contribution in determining the polarity of the review. Example, “The movie was good and interesting”, here the sentiment words good and interesting tells us that the polarity of the movie is positive.

2. Sarcasm - It is really difficult to know the tone of author in textual sentences, we can't definitely say that bad means bad or good. For example, “The movie was supposed to be hilarious?”

3. Parsing - What does the verb and/or adjective of a subjective or objective textual sentence really refer to?

4. Scaling - What is the quantity of data input as a proportion of the total universe of users? 10% of the IMDB corpus gives you a rough idea of what's going on but the result are nowhere close to the resolution you get with 50% of the reviews.

Models we have used:

And after preprocessing we have implemented our models.

We have implemented both ML and DL models to do the sentiment analysis of movie reviews.

We have used Random Forest, Naïve Bayes, KNN models of ML implementation. And have implemented LSTM for DL implementation.

We have compared all the models we have tested. And LSTM model shows the highest accuracy among all.

Where ML models fail:

Now, let’s say we have given the input to the system i.e. movie review like this- The movie was bad but story was good. Here, ml models will give output as negative as they have no dependency among each other. But in LSTM, in DL, it will recognize good with previous dependency and it will give some threshold like in sigmoid if threshold is let’s say greater than 0.5 than will output positive otherwise negative.

ML models implementation:

1) Random Forest:

The underlying data structure of the forest classifier is a decision tree, but with random selection of features to split on.

2) Naive bayes:

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

3) KNN:

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry.

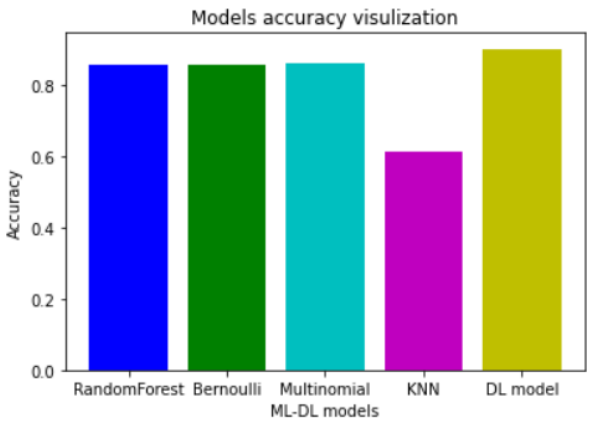
DL models implementation:

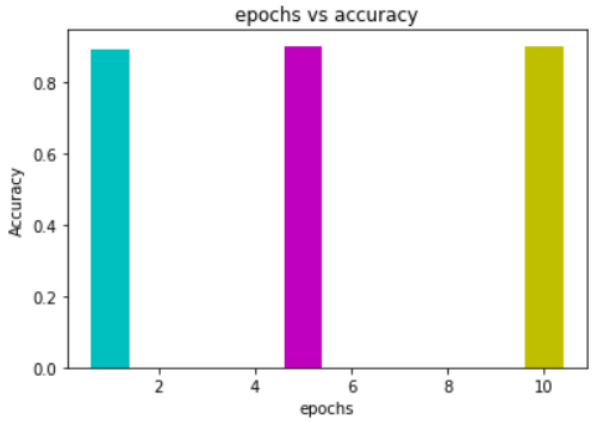
LSTM(long short term memory):

We will map each movie review into a real vector domain, a popular technique when working with text called word embedding. This is a technique where words are encoded as real-valued vectors in a high dimensional space, where the similarity between words in terms of meaning translates to closeness in the vector space. We can develop a small LSTM for the IMDB problem and achieve good accuracy.

The first layer is the Embedded layer that uses 32 length vectors to represent each word. The next layer is the LSTM layer with 32 memory units (smart neurons). Finally, because this is a classification problem we use a Dense output layer with a single neuron and a sigmoid activation function to make 0 or 1 predictions for the two classes positive and negative) in the problem. Because it is a binary classification problem, log loss is used as the loss function binary_crossentropy in Keras. The efficient ADAM optimization algorithm is used. The model is fit for only 2-5 epochs because it quickly overfits the problem.
We have tried at epochs -> 1 , 5 and 10. . At epoch = 1 , accuracy we achieved is 89. At epoch=5, accuracy is 90. At epoch=10, it overfits , accuracy = 86

Accuracy plots:





Sr No.	Model	Accuracy
1	Random Forest	85.68%
2	Naïve Bayes	85.43%
3	KNN	61.13%
4	LSTM	89.98%

Conclusion:

As we can see from this table, Random Forest gives enough accuracy – 85.68% and on the other hand KNN gives very less efficient. DL model LSTM gives the best accuracy – 89.98%.