Sentiment Analysis of IMDB Movie Reviews Using Long

Short-Term Memory

Swraj Kant Sharma

*Department Of Computer Science and Engineering*

*Delhi Technological University*

New Delhi, India

[Swrakant28@gamil.com](mailto:Swrakant28@gamil.com)

***Abstract****\_\_***With the rapid increasing rate at which textual data is being generated, it has become major research subject to analyse the data and extract the valuable information. Data collection from the internet can be used to solve various objectives such as building of recommendation system, doing the sentiment analysis etc. The data which we acquire from the internet is very unstructured. To handle such unstructured data, Deep learning models are suitable in handling such data and make some valuable prediction by analysing the data. In this paper the Long Short-Term Memory (LSTM) based model/classifier has been proposed for inspecting sentiments of IMDB movie review dataset. LSTM is based on Recurrent Neural Networks (RNN) algorithms. The dataset has been pre-processed using the technique of Natural Language Processing (NLP) and has been effectively portioned to enhance the accuracy of the model. Results show an accuracy of around 90.10% on the test data. This model can further be integrated on text base sentiment analysers.**

Keywords- ***Natural Language Processing (NLP), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), IMDB Movie Review Dataset***.

I. INTRODUCTION

In recent years, due to sudden increase in the use of internet and social media, sentiment analysis gained a wide range of popularity among people and especially among various organizations and businesses. The sentiment analysis is the process of using Natural Language Processing (NLP) combined with some machine learning / Deep learning models to extract, identify and categorize different opinions expressed in the form text. Two of the most popular methods that are used for sentiment analysis are: Lexicon Based Sentiment Analysis and Machine Learning / Deep Learning based sentiment Analysis.

In the deep learning method, a more complex set of neural networks is being trained on the data referred as training data set. The training followed by an evaluation step by testing the pre-trained deep learning model on the testing data set.

Although these classification techniques have rapidly advanced in last couple of years, there are still several ambiguities that has not been solved yet and it leads to the poor performance of the model.

The main limitation that includes are: Keywords which hold various different meanings as per the context may lead to ambiguity, and the model is incapable of classifying such sentences which does not include clear emotional keywords. Therefore, the proposed system should account for these drawbacks and provide accurate measures for the accurate classification of the data since these models may get by deployed on sensitive and important applications so the misclassification and ambiguities need to minimized as much as possible.

In this paper, I am going to propose a RNN (Recurrent Neural Networks) based model that used LSTM (Long Short-Term Memory) to analyse the IMDB movie review dataset. The entire dataset is divided into training and testing parts. The training dataset has been used to train the LSTM based model and the testing set has been used to quantify the accuracy. Confusion matrix has also been plotted to better understand the results.

II. Methods and Materials

The propose Deep learning-based sentiment analysis model has been presented in Fig. 1.

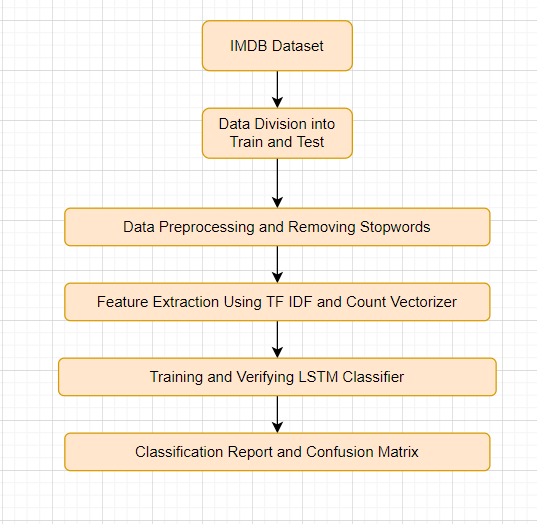


Fig. 1: - The proposed System Flowchart

1. *IMDB Dataset*

The IMDB Dataset contains 50k movie reviews. The dataset has already been divided into train and test set by Keras. The training set contains 25k reviews and the test set also contains 25k reviews. In addition, both training and test contains 12.5k positive and negative reviews. The reviews are classified into positive and negative in reference to the IMDB rating system. A viewer can rate a movie on a scale of 1 to 10, and according to the creator the review <=4 stars has been labelled as negative and the reviews >=7 has been labelled as positive. From the dataset it can be seen that each movie has at most 30 reviews. Overall, the dataset contains 88585 different words.

1. *Division of Data into Train and Test Set*

Out of the 50k reviews 80-20 split has been done which gives 80% data to the training and 20% of the data to the testing. The training data has been used to train the LSTM Classifier. Out of the 80% of the training data 20% of the data has been used for Validation Purposes. The Validation Data is a subset of the Training Data that does not train, but prevents the model from Overfitting. The Test Data has been used to evaluate the model accuracy. The split of the Training, Validation and Testing can occur in many ways. However, a general notion exist that Training gets the most Data.

1. *Data Preprocessing and Removing of Stop Words*

Before I discuss the process of removal of Stop Words first let’s understand a bit about stop words. Stop Words refers to the some commonly Occurring words in a sentence that do not help in determining the sentiment of that sentence. More precisely, Stop Words are the irrelevant Words. So before feeding the data to the Deep Learning we should remove these Stop Words. The most common stop words used in a text are “I”,” a”,” an”,” the”,” while”,” for”,” where”,” to”,” from” etc. Consider the following example “I was happy and I loved acting in the movie”. Now the words “I”,” was”,” and”,” in” adds no meaning to the sentence. Whereas the words “happy”,” loved”,” acting”, and “movie” are the unique keywords and on parsing them we can get useful results. NLTK generally provides us with a bunch of stop words almost from 16 different languages. To be more precise we can add more stop words on our own that may increase the efficiency of the model.

1. Feature Extraction using TF IDF Vectorizer

In this project we are dealing with text, but we need to understand that the Deep Learning models can work with numbers. Therefore, to associate the text with numbers we have used the natural language.

Term Frequency (TF) – It is a physical quantity defines the number of times a word occurred in a given corpus/ document.

(1)

: frequency of word i in document j.

: number of words in document j.

: Tern Frequency

Inverse Data Frequency (IDF) – It is a physical quantity used to calculate that how many times a particular term appeared across all the document. It basically predicts whether a particular term is common or rare in a given set of corpus/ document.

(2)

*N*: total number of documents.

: document frequency.

: inverse data frequency.

TF-IDF is the product of equation (1) and equation (2)

(3)

: Tern Frequency.

: inverse data frequency.

: TF-IDF formulae.

1. *Count Vectorizer*

Count Vectorizer basically takes a collection of text document and converts it to a vector of terms containing the count of each kind of tokens. By doing Count Vectorizer we basically feed Vector containing numerical data instead of text words that helps in making workable representation of the important features of the dataset.

1. *LSTM Architecture*

The Proposed LSTM neural network architecture is shown in Fig. 2. In the architecture I have use a word embedding in first layer, in which our embedding learns all words from IMDB Movie Review Dataset. The size of the vocabulary is 10k and the output dimension of the embedding is (64 x 10000) matrix, in which the maximum length each batch is 64.

The Output generated from the embedding layer is then fed to the LSTM layer. When the output generated from the embedding layer is passed through the LSTM layer it produces a context vector of dimension 32. After this the 32-dimensional vector is then feed into the Fully Connected Dense Layer with Sigmoid activation producing the vector as input to predict with 2 units (Positive, Negative). The Output Vector has a dimension of (32 x 1).

I have used **Binary Cross Entropy** as loss function and **rmsprop** Optimizer having a learning rate of 0.001 and the total number of epochs that I used to train the model is 10 and the data if feed to the model with a batch size of 128.

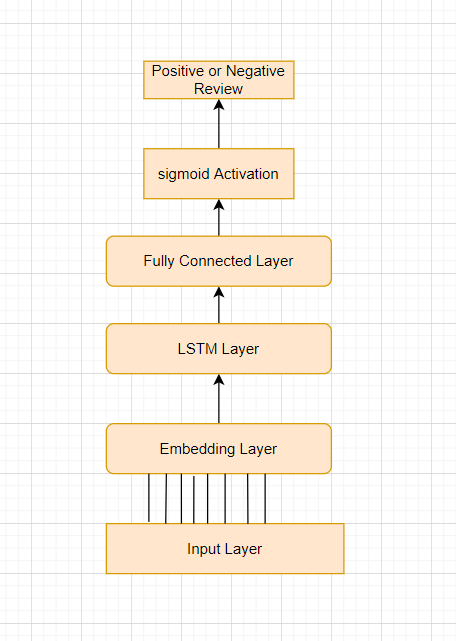


Fig. 2: - Proposed Model for Sentiment Analysis (LSTM)

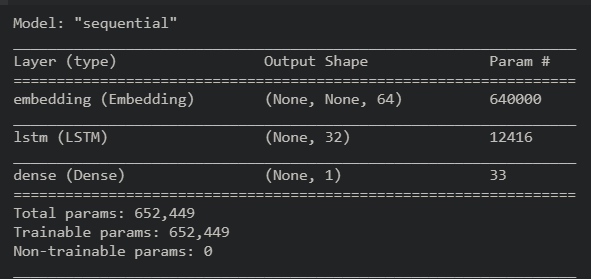


Fig. 3: - Details of LSTM model

1. Classification Accuracy

The Classification Accuracy basically predicts how accurate the model performed on the testing data after training has been done. It is the measure of percentage of text that have been correctly classified to their respective labels. The mathematical formulation for accuracy has been given in Equation (4).

(4)

TP – Texts that have truly classified in the positive class (True Positive)

TN – Texts that have truly classified in the negative class (True Negative)

FP ­- Texts that have falsely classified in the positive class (False Positive)

FN – Texts that have falsely classified in the negative class (False Negative)

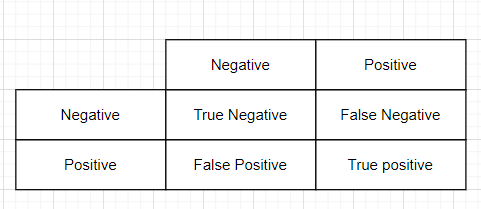


Table I. Confusion Matrix Outline

III. RESULTS AND ANALYSIS

After training the model, once the model was tested on test to predict the sentiment of a given text, the accuracy the maximum accuracy that I achieved on the test data was **90.10%** and, on the training, data was **93.10%**.

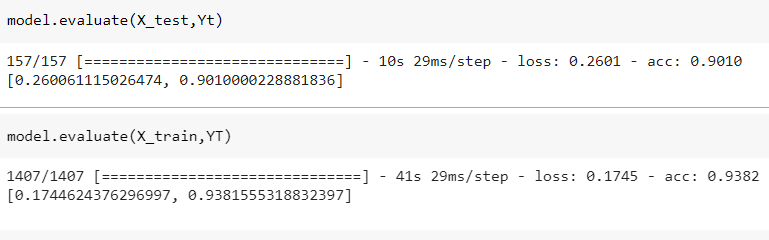


Fig. 4. – Loss and Accuracy of the test and the train data

As part of result analysis, the Fig .5. represents the training and validation accuracies over 10 epochs. Fig .6. represents the training and validation loss over 10 epochs.

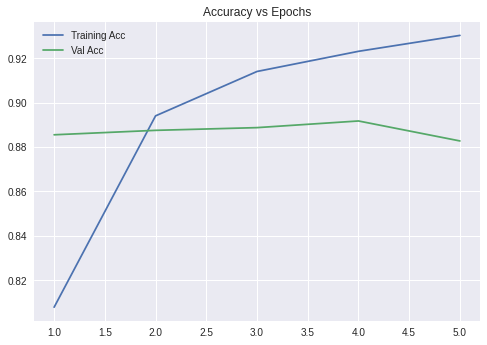


Fig. 5.- Training and Validation Accuracies

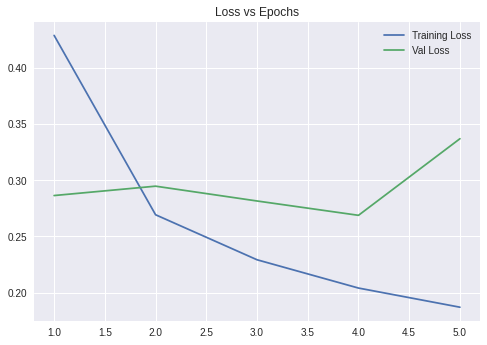


Fig. 6. – Training and Validation Loss

For the test data of IMDB Movie Reviews, the confusion matrix has been generated to analyse the accuracy. Confusion Matrix Provides us about the correct and incorrection classification of the sentiments.

The result of the test data has been represented in Table. 2 as the confusion Matrix. The accuracy of the test data was found around **90.10%.**

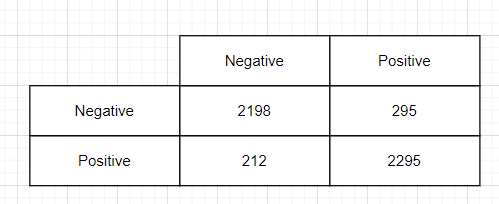


Table 2 – Confusion Matrix Result

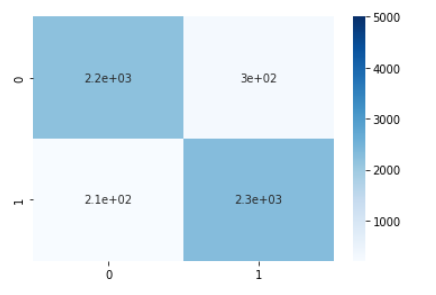


Fig. 7. – Confusion Matrix

By looking at the accuracy on the entire dataset, the classifier has achieved sufficient precision on the given dataset. A better accuracy can be obtained by employing more text cleaning approaches and ensembling various classification models.

IV. CONCLUSION

In this paper, I tried to provide an overview on the sentiment analysis on the IMDB Movie Reviews Dataset. The approach proposed by me classified the reviews into positive and negative class using the RNN-LSTM model and doing Natural Language Processing. By tuning the hyperparameter of the model the maximum accuracy that I achieved is **90.10%**.

A better accuracy can be achieved by some advanced data Preprocessing and preconditioning techniques. In future I have planned to train this model on some hybrid deep learning models and also by ensembling various classification models.

References

[1] S. M. Qaisar, "Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory," 2020 2nd International Conference on Computer and Information Sciences (ICCIS), 2020, pp. 1-4, doi: 10.1109/ICCIS49240.2020.9257657.

[2] R. Agarwal, D. Jain, G. Sharma and J. Tiwari, "Machine Learning based Click Bait Detection System," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 1714-1720, doi: 10.1109/ICAIS50930.2021.9395898.

[3] D. Goularas and S. Kamis, "Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data," 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), 2019, pp. 12-17, doi: 10.1109/Deep-ML.2019.00011.

[4] V. Tyagi, A. Kumar and S. Das, "Sentiment Analysis on Twitter Data Using Deep Learning approach," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020, pp. 187-190, doi: 10.1109/ICACCCN51052.2020.9362853.

[5] B. Nath Saha and A. Senapati, "Long Short Term Memory (LSTM) based Deep Learning for Sentiment Analysis of English and Spanish Data," 2020 International Conference on Computational Performance Evaluation (ComPE), 2020, pp. 442-446, doi: 10.1109/ComPE49325.2020.9200054.

[6] D. Li and J. Qian, "Text sentiment analysis based on long short-term memory," 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI), 2016, pp. 471-475, doi: 10.1109/CCI.2016.7778967.

[7] A. Aziz Sharfuddin, M. Nafis Tihami and M. Saiful Islam, "A Deep Recurrent Neural Network with BiLSTM model for Sentiment Classification," 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), 2018, pp. 1-4, doi: 10.1109/ICBSLP.2018.8554396.