**Sentiment Classification-sentiment140-B**

**Table of Contents:**

1.Project Description

2.Dataset Description

3.Technologies Used

4.Model Description

5.Performance Description

**1.Project Description:**

The goal of the project is to analyze the twitter data and perform sentimental analysis on them. The Twitter tweets are already scrapped and store in the train and test.csv for analysis . To analyze the text data BILSTM model has been used and it provides an accuracy of about 80% and above. Tweets contains lost of urls , usernames , numbers and emotions . All of these are to be preprocessed for better results . This project includes the preprocessing part as well which will be explained in the further sections. Classification report and accuracy analysis have been done on the Bilstm model . The dataset is about 16,00,000 data points and has to be run on GPU so I have used Google Colab to run this project.

**2.Dataset Description:**

The given dataset contains text tweets which are around 1,600,000 tweets extracted using the Twitter API.The sentiment columns has two values 0 as negetive and 4 as positive which is able of detecting sentiments.

There are total of 6 columns and are descripted below. Later on we convert the output class labels as 0 and 1 for bilstm to train .

1.sentiment: The polarity of the tweet (0 = negative, 4 = positive)

2.ids: The id of the tweet

3.date: The date of the tweet

4.flag: The query value.

5.user: The user account that tweeted

6.text: The tweeted text

**3.Technologies Used:**

Python libraries have been extensively used in this project .

Numpy and Pandas have been used to do array operations and data frame operations. Exploraray data analysis has been done with seaborn and Matplotlib libraries .WordCloud packages also have been used this project for showing the word counts in the twitter data . Text Vectorizer has been performed on the twitter text using Tokenizer and Pad\_Sequences packages. The tensorflow packages contains different hidden layers packages to build the bilstm module.

Flow of the Project :

1)” training.1600000.processed.noemoticon.csv” this file has been read which contains the twitter data and is stored in a dataframe.

2)Exploraray Data Analysis has been done on the project that includes , checking if the dataset is balanced or imbalanced or not. Word Cloud has been plotted.

**3)Data Preprocessing included:**

1.Lower Case: tweet texts are converted into lower cases.

2.Replace URLS: Links containing keywords like 'http' or 'https' we replace with empty string

3.Replace Usernames: Usernames start with @ they are replaced with empty string

4.Replace Consecutive letter: 3 or more consecutive letters are replaces with 2 .(Eg Hiiii to Hii)

5.Replace Emoji:replace with the emotion like 'smile'

6.Remove non-alphabets

7.remove stop words

4)Mostly sentiment and text columns are widely used in this project to train the model. Rest of the columns are dropped from the dataframe.

5)Bar plots to view the count of emotions in the dataset have been have been done.

6)Word2Vec vectorization has been performed on the text data with a vocabulary length of 7000. This is followed by tokenizing the text into words and creating the embedded matrix. Padding concept has been applied to make the train and text matrix length same.

7)Bisltm model has been created which included first layer as the embedded layer having word2vec in them. Followed by 2 bilstm layers and a dense layer having ‘relu’ as the activation function and last layer being ‘sigmoid ’ activation layer.

8)The model has been trained with different learning rates to check the accuracy .the loss function used is binarycrossentropy as the output is 2 classes which are 0 or 1. It has been trained for standard 20 epochs with a batch size of 520 . Optimizer used here is Adam .

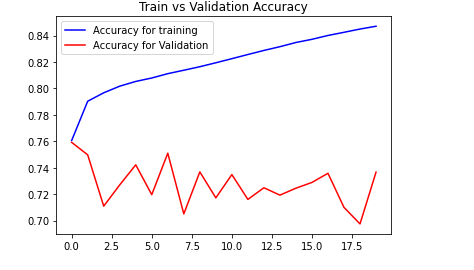
9)Confusion Matrix has been plotted with True Positive and True Negetive rate higher by varying the threshold to 0.25 from 0.5 .

10) Pickle and save the model.

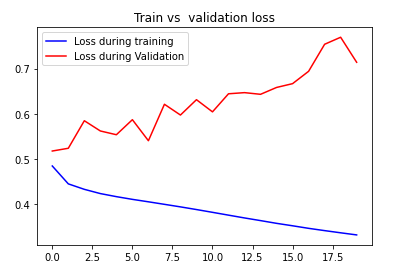
**4.Model Description**

A Bidirectional LSTM, or biLSTM, is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. BiLSTMs effectively increase the amount of information available to the network, improving the context available to the algorithm. Bilstm thus learns faster than traditional Lstm.

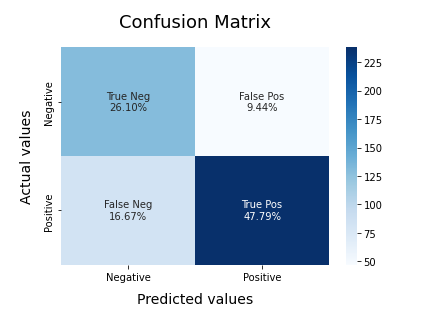
**5.Performance Description**

****

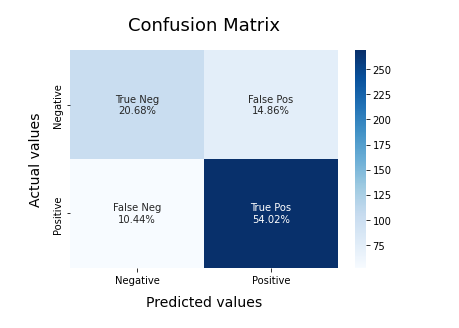
**Here we can see that there is a difference in accuracy of train and validation data by 10% . This may be due to dataset size difference which is quite huge.**

****

**Loss function for the Validation dataset is decreasing which is expected as the epochs increases.**

****

**True Positive rate is 48% when threshold is set to 0.5**

****

**True Positive rate is 55% when threshold is set to 0.25**