JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# Tracking Attention of Social Media Events using Bi-LSTM

**CH.Madhavi Sudha[#1], K.Ganesh[*2], B.Sai nath[#3]**

*[#]CSE-CSE Department, CBIT(A),Gandipet*

[1]madhavisudha_cse@cbit.ac.in

[2]ugs18306_cse.battu@cbit.org.in

[3]keshaboinaganesh665@gmail.com

***Abstract*—** The technique of following a real-life scenario, extracting emotion from unstructured text is an active and challenging area of research. It has diverse applications in various aspects of our daily life To overcome various challenges involved in detecting emotion from text, researchers from diverse fields applied various Machine learning algorithms i.e knn algorithm. However, deep learning methods such as Bi Directional Long Short-Term Memory are effective to detect emotion by maintaining the sequence structure of the text. In this work, we use Bi-directional long short-term memory with attention layer for emotion detection for better accuracy for prediction.In addition, we employ a text preprocessing method to improve further results. We perform the experiments on three data and the models are evaluated based on the classification accuracy.

***Keywords*— Machine learning. Deep learning, K nearest neighbour, Emotion Detection, Bi directional LSTM,**

## I. Introduction

The affective computing, the field of involving emotions in computing, emotion detection from text has emerged as an important domain. Its applications are wide-ranging, including measuring the emotional closeness of interpersonal ties using affective language in social networks, in marketing, predicting purchase intentions of customers and gauging brand reputation using emotional states, removing inappropriate posts from social media. Emotion detection from text neatly fits as a crucial intermediate step in many applications. The process of emotion detection starts from defining what emotions are exactly. There is no consensus among psychologists as to how to define and categorize emotions. There are various models like the categorical and dimensional models of emotions Over the years many computational approaches for this problem have been proposed, such knn- machine learning based algorithm. However, these models are not capable of maintaining the sequence structure of text to detect emotion effectively with better accuracy. A deep learning model Bi Directional Long Short-term memory (Bi-LSTM) has the capability to maintain sequence structure of the text and give better accuracy than a KNN machine learning algorithm.

## II. Literature Survey

This study of emotion detection from text pipeline primarily consists of three parts, choosing an emotion model to follow, identifying and aggregating relevant datasets for the emotion model chosen and applying a computational approach to perform the task of accurately determining emotions on given text. Machine learning can be broadly defined as inference of decision rules from a database of labelled training samples for the task of recognizing emotions..Knn and random forest are commonly used as models in this approach. We used a new approach as the emergence of deep learning in the recent past has motivated us to try it out in a variety of domains, including natural language processing (NLP).use a recurrent neural network architecture (LSTM in uni and bidirectional),with dropout and a weighted loss function, trained on word embedding's (GloVe). Note that these word embedding's do not take sentiment into account unlike works such as which attempt to incorporate sentiment features also into the embedding.

## III. Proposed System

**Bi LSTM:**

We used a deep learning architecture shown in Figure 1 Proposed Architecture (Bi-LSTM) to accurately identify emotions in datasets. In order to convert the words into a Suitable intermediate representation that can be fed into the neural network, we used word embedding's. We also Incorporated latent sentiment features into these embedding's as in to improve performance but due to potential instabilities during optimization, shown in Figure 2 system Design we followed common choices that advocate the use of Bi Directional long short-term memory networks. We extended the recurrent neural network architecture presented in as in to include dropout layers for regularization and use a weighted loss function to deal with class imbalance. We also incorporated the attention mechanism, to increase the accuracy. Finally, a fully connected dense layer was used for the voting process, the output of which was then passed through an activation function. Changes in this layer made our model flexible enough to adapt to both categories (using a SoftMax activation function).

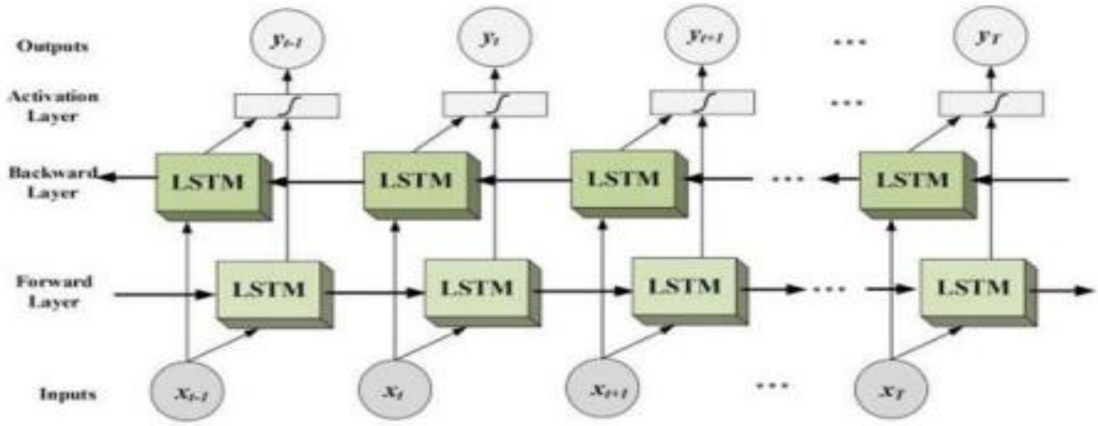



Figure 1: Proposed Architecture ( Bi-LSTM)

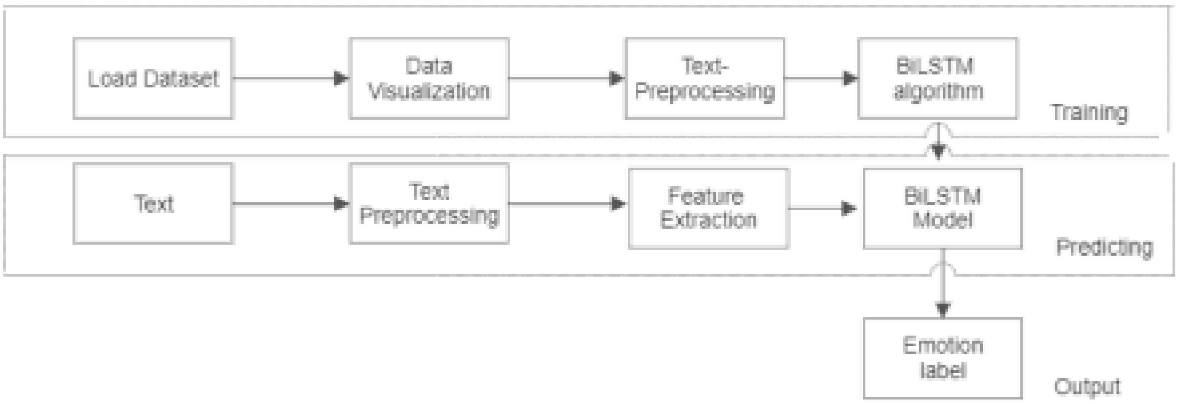



Figure 2: System Design

## IV. Implementation

### *A. Pre processing*

There are three stages to pre-processing. Cleaning input from a text file, which involves cleaning the text and converting it into clauses. Padding is done because, as we all know, all neural networks require inputs that are comparable in shape and size. All of the padding clauses are converted into unique ids using Building Vocab.

### *B. Model Building*

We use a BiLSTM model with two layers. The lowest layer is made up of a number of word-level Bi LSTM modules, each of which corresponds to a single phrase and stores context information for each of the clause's words. A bi-directional LSTM is used to uncover the concealed state of the $j^{th}$ word in the $i^{th}$ clause $h_{i,j}$. After that, an attention method is used to obtain a clause representation. The clause representations are now fed into two BiLSTMs, one for emotion extraction and the other for cause extraction. Softmax is used to classify emotion and cause clauses in the output. The two BiLSTMs' hidden states are regarded as context-aware representations of the clauses. We concatenate the relative distance of each pair with a cartesian product over hidden states and train a logistic regression model to exclude the pairs that do not have a casual link..

When our training data and validating data is ready we can do the training. For that training we have to build the model. We create an algorithm with Sequential. Model has three layers Input layer,Hidden layers,output layer.

***DataSet Description***

Train Data

df_train = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/project_datasets_train/train.txt', header =None, sep =';', names = ['Input','Emotion'], encoding='utf-8')

Test Data

df_test = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/project_dataset_test/test.txt', header = None, sep =';', names = ['Input','Emotion'],encoding='utf-8')

Validation Data

df_val=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/project_dataset_validate/val.txt',header=None,sep=';',names=['Input','Emotion'],encoding='utf-8')

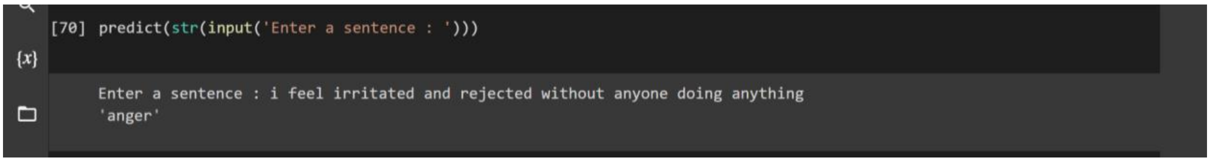
We have 3 datasets.One of them is used to test the model. We are going to do the prediction with the text data.Already,the model is trained. We are going to evaluate the model which gives complete loss and accuracy.

Real Time Prediction

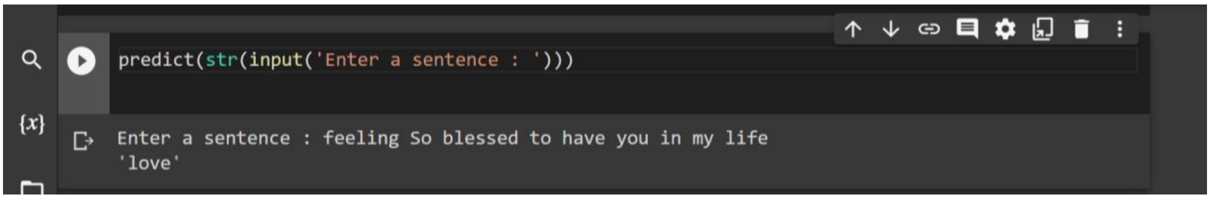
```
def get_key(value):
    dictionary={'joy':0,'anger':1,'love':2,'sadness':3,'fear':4,'surprise':5}
    for key,val in dictionary.items():
        if (val==value):
            return key

def predict(sentence):
    sentence_lst=[]
    sentence_lst.append(sentence)
    sentence_seq=tokenizer.texts_to_sequences(sentence_lst)
    sentence_padded=pad_sequences(sentence_seq,maxlen=80,padding='post')
    ans=get_key(np.argmax(model.predict(sentence_padded),axis=1))
    return ans
```

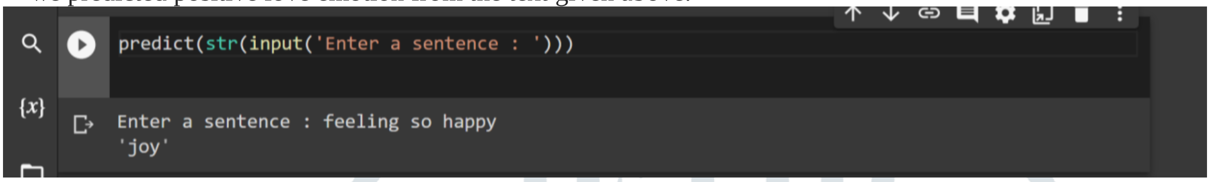
predict(str(input('Enter a sentence : ')))



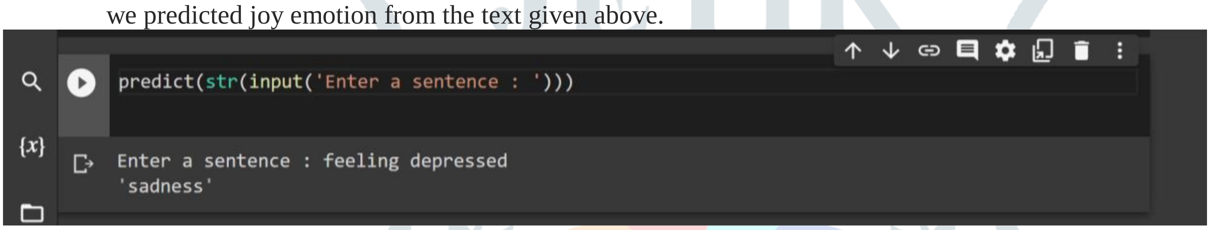

we predicted anger emotion from the text given above.




we predicted positive love emotion from the text given above.




we predicted joy emotion from the text given above.




we predicted sad emotion from the text given above.

we have predicted most common emotion from the text story.and here we connected twitter API .In this api we will have specific keys to access the data.and from the twitter data text we have predicted an emotion.

## V. Conclusions

We employed Bi-LSTM to perform emotion detection effectively over other selected deep learning methods as BiLSTM has bi-directional learning capability. We further observe that a suitable text pre-processing gives a high impact on the performance of the model. As a limitation of this work, the detection of emotions is mainly focused on Ekman's model of six emotion categories, the work may be extended further for more categories of emotion detection. The model may be improved further for emotion detection and the work may be further used better accuracy.

## References

[1] CM Suneera and Jay Prakash. "Performance Analysis of Machine Learning and Deep Learning Models for Text Classification". In: *2020 IEEE 17th India Coun cil International Conference (INDICON)*. IEEE. 2020, pp. 1–6.

[2 ] Kashfia Sailunaz et al. "Emotion detection from text and speech: a survey". In:*Social Network Analysis (2018).*

[3] Ondˇrej Bruna, Hakob Avetisyan, and Jan Holub. "Emo tion models for textual emotion classification". In: *Jour nal of physics: conference series*. Vol. 772. 1. IOP Publishing. 2016, p. 012063.