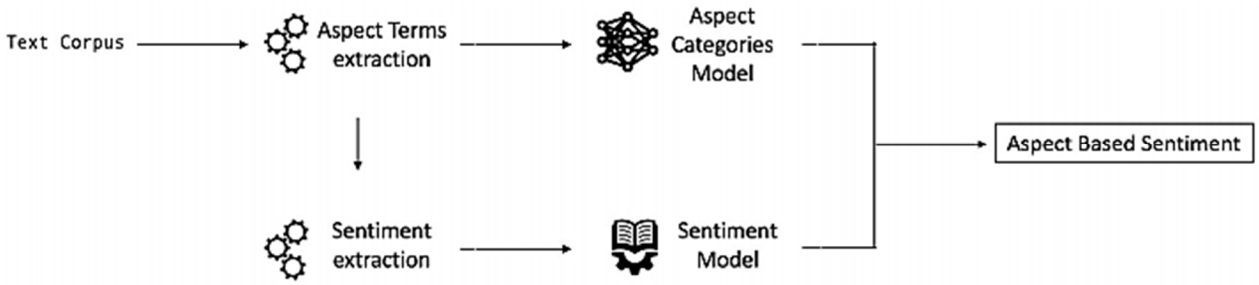
Data Science: NLP (2021) 12:42
https://www.societyofai.in/

Detection of depression and mental illness in Twitter

1 Introduction

Sentiment Analysis is an important task in Natural Language Processing and has a wide range of real-word applications. Typical Sentiment Analysis methods focus on predicting the polarity of a given sentence i.e., whether the sentence reflects a positive or negative sentiment. Building on this, a more complex subtask in Sentiment Analysis is predicting the sentiment towards a certain aspect or word mentioned in a sentence. This subtask is known as Aspect-Based Sentiment Analysis. ABSA comes in two variants that are implemented as two-step procedures as illustrated in Fig. 1.



In the context of our project, which is the application of Sentiment Analysis for Detection of depression and mental illness we focus on the primarily, ATSC Aspect-Target Sentiment Classification. Our dataset already consisting of target column having target value in 0 = negative, 2 = neutral, 4 = positive.

2 Related Work

The below figure is of our sample dataset Fig. 2.

	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Third Day to do it. ;D
0	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
1	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
2	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
3	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
4	0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew

Data Science: NLP (2021) 12:42
https://www.societyofai.in/

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1600000 entries, 0 to 1599999
Data columns (total 2 columns):
 #   Column  Non-Null Count    Dtype
---  ------  --------------    -----
 0   target  1600000 non-null  int32
 1   text    1600000 non-null  object
dtypes: int32(1), object(1)
memory usage: 18.3+ MB
```

```
df.target.value_counts()

4    800000
0    800000
Name: target, dtype: int64
```

The Sample dataset is of shape size 1600000 entries with 5 columns. But As per our project we can choose only. Hence, we have removed unnecessary columns i.e., 'id', 'date', 'query', 'users'

After the observation it's really important to work on data preprocessing and performing EDA. There are many steps for data cleaning as Expand contraction, Cleaning and removing punctuations, Cleaning and removing punctuations, Cleaning and removing repeating characters, Cleaning and removing Numeric numbers, Applying Stemming, Applying Lemmatizer. In our project we have removed hyperlinks, punctuation, contraction and apply stemming method for data cleaning. For EDA we have checked below points.

- Checked the info of the data and how much of memory and space it's taking.
- Checked for the null values and haven't found any.
- Checked the distribution of tweet lengths.
- Generated the word cloud for the Positive and Negative tweets.

Fig 3. Positive Tweet

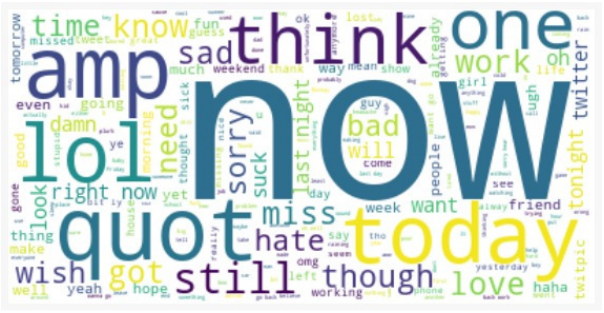


Fig 4. Negative Tweet

Data Science: NLP (2021) 12:42
https://www.societyofai.in/

3 Technical Approach

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feed-forward neural networks, LSTM has feedback connections. It can process not only single data points, but also entire sequences of data. we used LSTM (Long Term Short Memory) for this Classification task as it is a type of RNN Architecture (Recurrent Neural Network) which are extensively used nowadays for NLP because it handles long sequence dependencies well. For more details about LSTM, you can follow the reference link in the last of our report.

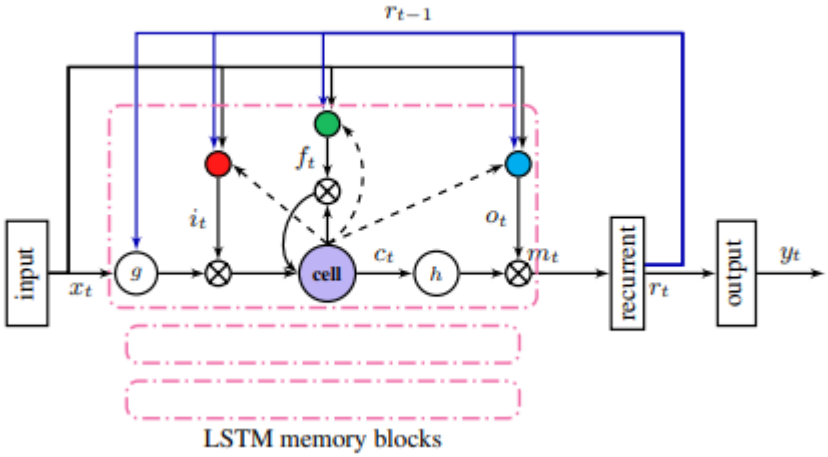


Fig 5: LSTMP RNN architecture. A single memory block is shown for clarity.

Model Summary:

```
1 lst_model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 30, 64)            640000
_________________________________________________________________
lstm (LSTM)                  (None, 130)               101400
_________________________________________________________________
dense (Dense)                (None, 1)                 131
=================================================================
Total params: 741,531
Trainable params: 741,531
Non-trainable params: 0
_________________________________________________________________
```

We have created a model using various layers.

Sequential: - In this problem we have used Sequential model as we have to understand the pattern in a long sequence of data, we need our network to analyze patterns in those long sentences.

Embedding layer: - As an input we will be getting sentences in textual form but our neural network won't understand textual form. So, for our Neural Networks to understand the sentences and ingest it, we need to convert them into some numeric form. We can do this by the method of Word Embedding. Word embedding represent words in continuous vectors. These vectors represent any word in few dimensions mostly based on the number of unique words in our text. we have used `pad-sequences` in this because our neural network assumes a vectorized representation of data. In this case we need our data to be transformed such that each sequence has the same length.

Sigmoid: -As our problem is a binary classification so we have used `Sigmoid` activation function here which turns all our output values in a value between 0 and 1.

Result

Model Accuracy: - We have achieved a Validation **Accuracy of 83%** and loss 0.394.

Performance Matrix: Below is the Classification Report showing the performance of our model.

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_preds.round()))
```

```
              precision    recall  f1-score   support

           0       0.83      0.81      0.82    159815
           1       0.81      0.83      0.82    160185

    accuracy                           0.82    320000
   macro avg       0.82      0.82      0.82    320000
weighted avg       0.82      0.82      0.82    320000
```

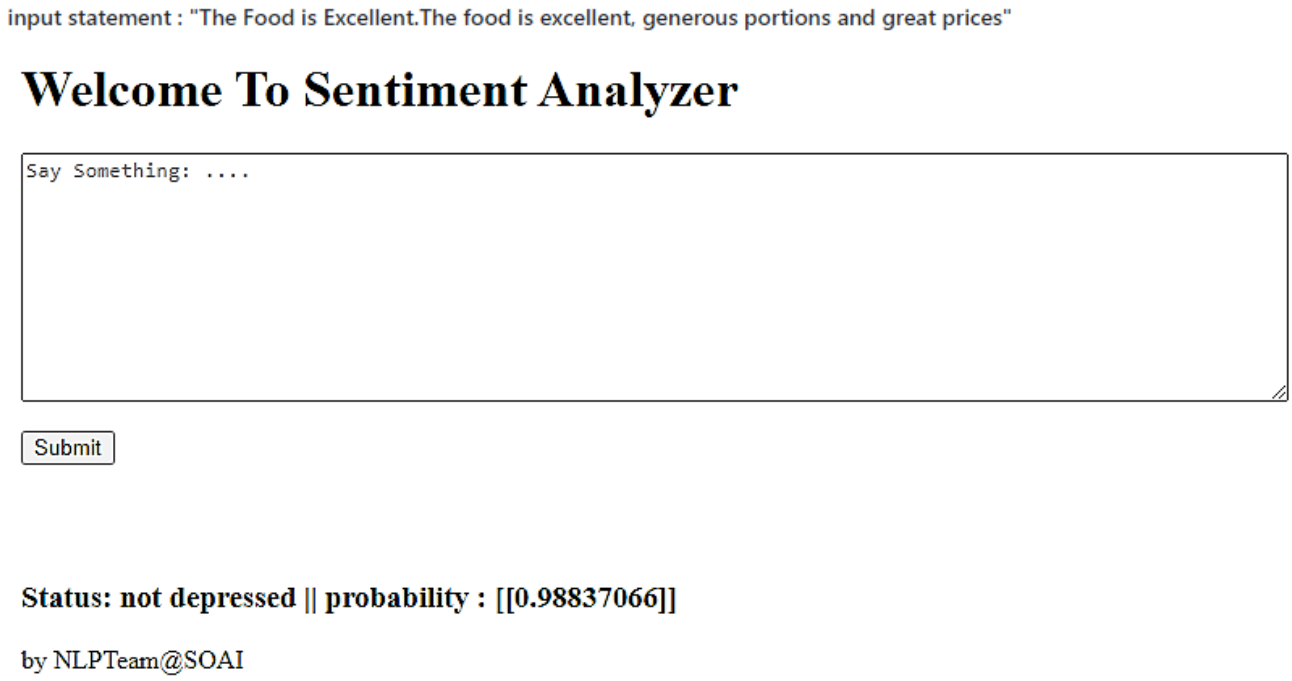
```
The review is
 ***It is a happy sunny day ***
 The model predcited that the review is POSITIVE and probablity is [0.9993554]
```

Data Science: NLP (2021) 12:42
https://www.societyofai.in/

4 Model Deployment

We have deployed our model on Heroku platform.
Link: https://nlplstm.herokuapp.com/

WEB UI:



5 Conclusions

We showed that deep LSTM RNN architectures achieve state of-the-art performance for large scale acoustic modeling. The proposed deep LSTM RNN architecture outperforms standard LSTM networks and DNNs and makes more effective use of the model parameters by addressing the computational efficiency needed for training large networks. We also show for the first time that LSTM RNN models can be quickly trained using NN.

6 Created By [NLP Team]

- *Naman Dixit*
- *Makarand Rayate*
- *Monish R*
- *Aakash Godara*

7 References

- https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf
- https://arxiv.org/abs/1808.03314
- https://paperswithcode.com/method/lstm