Emotion extraction from text: Analysis of four different algorithms

Nimisha Jadav[1], Komal Shirsat[2], V.Preethi[3], Prof. Mohan Bonde[4]

[1]Usha Mittal Institute of Technology, SNDT Women's University, Mumbai,
[2]Usha Mittal Institute of Technology, SNDT Women's University, Mumbai,
[3]Usha Mittal Institute of Technology, SNDT Women's University, Mumbai,
[4]Asst. Professor, Usha Mittal Institute of Technology, SNDT Women's University, Mumbai.

Abstract- Emotions play an important role in human life. In today's internet world textual data has been proven as a main medium of communication in human-human and human-machine interaction. Thisincrease in use of text for communication makes it a need to find the emotion present in textual data. This makes it important to provide a framework that can recognize the emotions present in the communication or the emotions of the users. Emotion recognition or sentiment analysis is a natural language processing task that mine information from textual data liketwitter tweets, blogs, movie reviews and reviews from online websites and classify on the basis of polarity. Emotion Recognition Model extracts emotion from text at the sentence level. Our method uses 4 different deep learning algorithm and detects emotion from a text-input. To make the recognition more accurate, emotion-affect-bearing words and phrases were also analyzed. This experiment shows that the method could generate agood result for emotion detection from text input.

Keywords-Natural language processing (NLP), LSTM, SVM, CNN, HAN, tweepy.

I. INTRODUCTION

Emotions play an important role in social media and they are used to stimulate cognitive processes for strategy making. Today, there is a huge amount of data available in text format on web. Marketing aims to stimulate emotions in customer for trying them to brand and to increase the sale of product or service. Analyzing the emotions and sentiments of various textual data over the Internet has its own advantages, we can measure the well-being of a community, we can prevent suicides, we can improvethe quality of the product from its reviews and also it can be very helpful for organizations to measure the degree of satisfaction of their customers by analyzing the comments or the feedback they provide.

Twitter is a rapidly growing social networking website where people express their opinions in a short and simple manner of expressions. Also people can get live news and entertainment about sports, politics or any other event which is happening 1000 miles away from them. It is a platform where merchants sell products on the Web and ask the customers to review the products.

Sufficient amount of research has been done related to speech and facial emotion detection but text based emotion recognition system still requires attraction of researchers and data scientist. While performing Natural language processing tasks on text data the short messaging language have the ability to interrupt and falsify[1].

For example, "wht u doing tonit, we can go to baseball match or we can go fr movie". While going through this sentence, it will recognize some terms which doesn't belong to decent plain text. While going through these messages, human brain will resolve the short messaging language word to a meaningful word or phrase immediately. When human see "wht" and its neighboring words "u" human brain knows that it is "what" and "you". That's because human brain is trained with previous experiences. But in case of Natural Language Processing tools, they are trained and adopted to work properly with plain textual data. Mapping between short messaging language words and plain text words can be very sensitive in some cases.

A wrong mapping can lead to alternations of themeaning or it may destroy semantics under the applied context which will eventually give wrong results. When considering the sub phrase "tonit we can go fr" in above example, "tonit" can be considered as "tonight" .Humans

can understand thatit is "tonight we can go for" and but a direct mapping from a language tool will face difficulties. Hence it depends on the context in which the word has been used.

II. RELATED WORK

Zichao Yang, Diyi Yang Chris Dyer, Xiaodong He, Alex Smola, Eduard Hovy has proposed a hierarchical attention network (HAN) for document classification, A model with two levels as word level and sentence level of attention mechanism to visualized the document in such a way that it highlights and aggregates important words in sentences[2]. Erdenebileg Batbaatar , Meijing Li, And Keun Ho Ryu developed a Semantic Emotion Neural Network (SENN) which makes use of bidirectional Long-Short Term Memory (BiLSTM) to extract emotion as Ekman's six basic emotions[3]. Priyanka H S and Ramya B V proposed a way to find polarity of movie reviews by using logistic regression. They performed feature extraction and classification using machine learning approach and their model achieved accuracy upto 88% [4].

Adarsh S R proposed a way for emotion recognition based on word clustering which helps in reducing dimensions of feature space and shows improvements in results[5]. S.Shaheen, W.El-Hajj, H.Hajj, and S.Elbassuoni, has created model based on KNN classifier which uses automatically generated rules for emotion detection. If the terms and their relation to the meaning of the sentence are found, they can be easily generalized and considered as emotion recognition rules (ERRs). The emotions considered in these study are happiness, sadness, disgust, anger, surprise and fear[6]. Romana Rahman, Tajul Islam, Md. Humayan Ahmed has created a method for Emotion detection from text and emoticon, based on keyword analysis, keyword negation analysis, and by finding proverbs, emoticon and exclamatory word from sentences. Their method achieved upto 80% accuracy[7].

Nida Manzoor Hakak, Mohsin Mohd, MahiraKirmani, Mudasirmohd, Department of CS has surveyed prior works which has been done in the field of emotion analysis from text[1]. Sudhanshu Prakash Tiwari, M. Vijaya Raju, GurbakashPhonsa and Deepak Kumar Deepu have proposed a hybrid approach of emotion recognition which is a combination of both machine learning and keyword based approach. And they concluded that hybrid approach of emotion recognition gives better results than model which is based only on learning based or keyword based[8]. S. N. Shivhare and S. K. Saritha, proposed model that is based on keyword searching technique which also uses the concept of ontology which makes this model more efficient than other methods in recognizing emotions from text input[9]. This has been created to overcome following limitations: Ambiguity in Keyword Definitions, Incapability of Recognizing Sentences without Keywords and Lack of Linguistic Information. Chetan R. Chopade surveyed all emotion detection methods viz.,keyword-based, Lexical affinity method, learning based, and hybrid based approach[10].

F. Calefato, F. Lanubile and N. Novielli,has developed java based "EmoTxt: A toolkit for emotion recognition from text," which supports emotion classifiers from manually annotated training data[11]. It identifies emotions in an input corpus provided as a CSV file, with one text per line with unique identifier. This gives output as a CSV file containing the text id and the predicted label for each item of the input collection. S.L.Ramírez-Ávila, R. Oramas-Bustillos, M.L. Barron-Estrada, R.Zatarain-Cabada describe the development of system to generate a corpus of textual opinions in Spanish, labeled with learning-centered emotions. Here, the corpus generated with the ERAS system contains 851 textual opinions. The system remains available for more participants to express their opinions on educational resources[12]. This model will help Intelligent Tutoring Systems to detect emotions through text and make the teaching process more efficient for students, adjusting the content to the particular needs of each of them.

III. ALGORITHM

a) *LSTM*

Long Short Term Memory is a special kind of Recurrent Neural Network (RNN) desgined by Hochreiter and Schmidhuber. LSTM is used to solve problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long term memory but can give more accurate predictions from the recent information.

LSTM has three gates such as input, output and forget gate. They are composed of a sigmoid neural net layer and a point-wise multiplication operation.

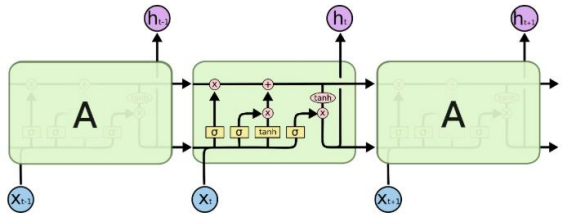


Figure 1. LSTM model structure.

b) CNN

Convolutional Neural Network (CNN) is a class of deep, feed-forward artificial neural networks (where connections between nodes do not form a cycle) and use a variation of multilayer perceptrons designed to require minimal preprocessing.

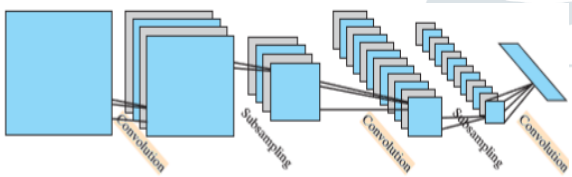


Figure 2.Convolutional network

Convolutional neural networks show outstanding results in image and speech applications. In Text Understanding from Scratch, CNNs can achieve outstanding performance without the knowledge of words, phrases, sentences and any other syntactic or semantic structures with regards to a human language. Semantic parsing, paraphrase detection, speech recognition are also the applications of CNNs.

c) SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems.The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N- the number of features) that distinctly classifies the data points.

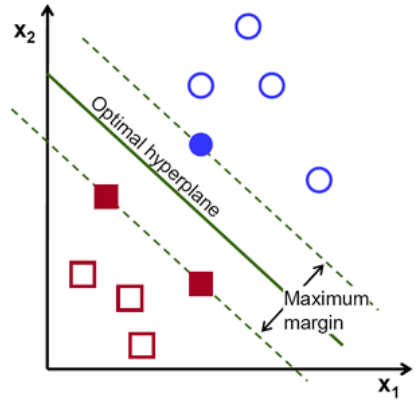


Figure 3. hyperplanes in SVM

d) HAN

The overall architecture of the Hierarchical Attention Network (HAN) is shown in Figure.

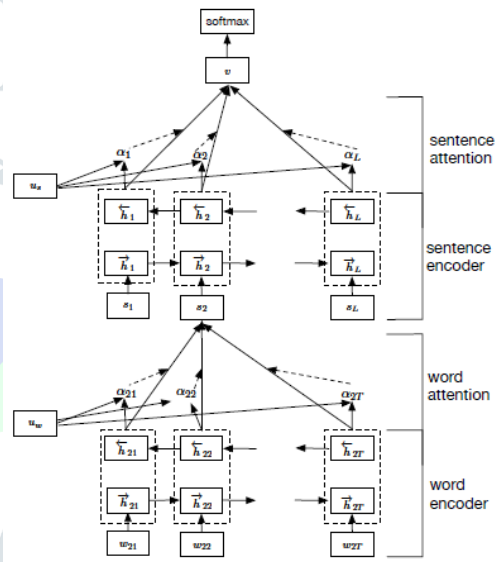


Figure 4. Architecture of HAN

It uses stacked recurrent neural networks on word level followed by attention model to extract such words that are important to the meaning of the sentence and aggregate the representation of those informative words to form a sentence vector[2].

IV. METHODOLOGY

The process of emotion analysis or emotion classification from text can be done in three different ways which are Keyword-based detection, Learning based detection, and Hybrid detection[6].

a) Keyword-based detection:

Here, classification is done by searching for theEmotional keywords in the input sentence. But these methods suffer

from the ambiguity definitions because word can have different meanings according to usage and context.

b) Learning-based detection:

In these methods, the emotion is detected by using classification approaches by applying learning algorithms based on a training dataset.

c) Hybrid detection:

In hybrid methods, emotions are detected by using a combination of emotional keywords and learning patterns collected from training datasets.

i. Implementation steps for offline dataset:

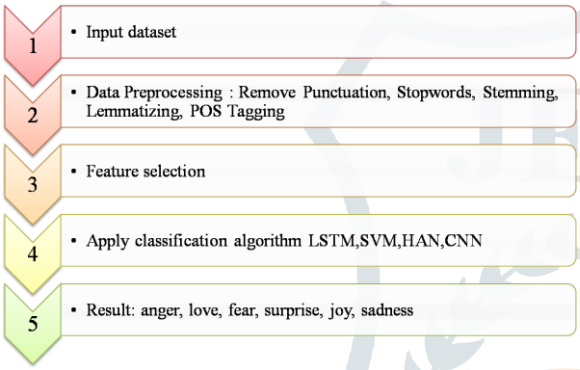


Figure 5. Implementation process flow

We have used Spyder IDE which comes along with the anaconda installation to do all our programming.

A. *Choosing Dataset*

To perform emotion analysis we can get textual data from multiple website like a movie review website, or product reviews from Amazon, Ebay, Nykaa, Zomato, etc. But, in our project we are using labeled text dataset. This dataset contains total 416809 tweets in total, labelled into different human sentiments. We import the basic libraries and then read the dataset.

B. *Preprocessing the Data*

The dataset which we are using has tweets which includes special characters, punctuation, plural forms, short forms and also past participles of words and stop words like prepositions.We need to remove all these words from tweets to make this entire process easy. Also we need to breakdown sentences into parts likes small tokens.We can perform this tasks using different Natural Language Processing and Natural language toolkit. To do this preprocessing and for making all sentences uniform we have applied following steps:

- Making all letters lowercase
- Removing Punctuation, Symbols
- Lemmatisation
- Stemming
- Finding antonyms and synonyms
- Finding Stop words
- Tokenizing Text
- Speech Tagging

C. *Feature Extraction*

After doing data preprocessing we make the data clean, error-free, precise and we represent each sentence by a group of keywords. After this step we use Feature Extraction method, i.e. We extract parameters from the data which can be presentednumerically. For this we are using two different features which are TF-IDF and Count Vectors. Before applying feature extraction features we will split the dataset into training and testing parts.

- TF-IDF:

The Term Frequency-Inverse Document Frequency(TFIDF) feature is used by data scientist to check therelative importance of a term in the data and to measure how frequently and rarely it appears in the text.

$$TF(term) = \frac{no\_of\_times\_term\_appears\_in\_doc}{total\_no\_of\_terms\_in\_doc} \quad (1)$$

$$IDF(term) = \frac{total\_no\_of\_doc}{no\_of\_doc\_with\_term\_in\_it} \quad (2)$$

$$TFIDF = TF(term) \times IDF(term) \quad (3)$$

- Count Vectors:

We are using this feature to transform our tweet ortext sentences into an array having the number of appearances of each word in document. We use this features considering the possibility that sentences that conveys similar emotions can have the same words repeated over and over again.

D. *Training Our Models*

After doing data preprocessing and feature extraction we get the numerical representations of sentences from dataset, we can apply them as inputs for deep learning models.

In our project we trained our learning models to classify given sentence using 4 different algorithms such as Long Short-Term Memory (LSTM), Convolutional Neural Network(CNN), Support vector machine(SVM) and Hierarchical Attention Network (HAN) . These methods can be used for tackling any kind of classification

problem. In our case, we want to classify a given tweet into one of the 6 emotions. For this we are using a dataset which contains 416809 tweets to train our model.We also check accuracy of these four models.

E. Testing

After performing all the above steps successfully we can test this model to check if it performs in reality by giving some random text input.

ii. Implementation steps for real time data

We have also perform the experiments on real-time tweets. For this we had connected our twitter API to python program. In order to use Twitter's API, we have to create a developer account on the Twitter apps site as per following steps.

- If you already have a twitter account then Log in or make a Twitter account at https://developer.twitter.com
- Create a new app.
- Fill in the app creation page with a unique name, a website name, and a project description.
- Once the application is created , open the 'Keys and Access Token' tab.
- Copy 'Consumer Key', 'Consumer Secret', 'Access token' and 'Access Token Secret'.
- Establish the connection with Twitter API.
- Calculate the polarity.
- Plot the pie chart with all the 6 polarities

V. RESULTS

a) Results for offline data:

After training the models using 4 different algorithms we get output as follows which show the iterations and time taken by model to perform each iteration, Also accuracy and loss at each iteration. These learning curves show how accuracy and loss varies from 1st iteration to Nth iteration.

```
In [5]: runfile('C:/Users/NimishaJadav/Desktop/project/emotion_csv.py', wdir='C:/Users/
NimishaJadav/Desktop/project')
C:/Users/NimishaJadav/Desktop/project/emotion_csv.py:92: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy
  data.text[i] = words
              precision    recall  f1-score   support

       anger       0.87      0.89      0.88      5658
        fear       0.81      0.86      0.83      4809
         joy       0.91      0.91      0.91     14155
        love       0.76      0.72      0.74      3455
     sadness       0.93      0.92      0.92     12088
    surprise       0.71      0.68      0.70      1516

    accuracy                           0.88     41681
   macro avg       0.83      0.83      0.83     41681
weighted avg       0.88      0.88      0.88     41681

processing time: 17992.54545068741 seconds
```

Figure 6. Iterations after applying SVM

After training dataset by Hierarchical Attention Network (HAN), we got results as

```
In [18]: cp=ModelCheckpoint('model_han_.hdf5',monitor='val_acc',verbose=1,save_best_only=True)
         history=model.fit(x_train, y_train, validation_data=(x_val, y_val),
                   epochs=3, batch_size=5,callbacks=[cp])

Train on 375129 samples, validate on 41680 samples
Epoch 1/3
375129/375129 [==============================] - 167525s 447ms/step - loss: 0.1615 - acc: 0.9240 - val_loss: 0.1171 - val_acc:
0.9373

Epoch 00001: val_acc improved from -inf to 0.93731, saving model to model_han_.hdf5
Epoch 2/3
375129/375129 [==============================] - 173336s 462ms/step - loss: 0.1160 - acc: 0.9381 - val_loss: 0.1108 - val_acc:
0.9368

Epoch 00002: val_acc did not improve from 0.93731
Epoch 3/3
375129/375129 [==============================] - 169453s 452ms/step - loss: 0.1195 - acc: 0.9389 - val_loss: 0.1381 - val_acc:
0.9360

Epoch 00003: val_acc did not improve from 0.93731
```

Figure 7. Iterations after applying HAN

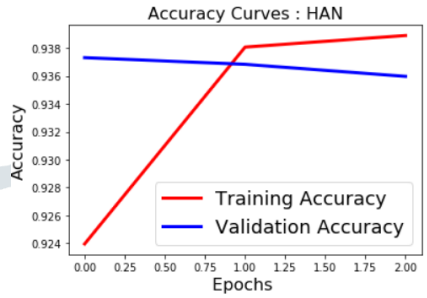


Figure 8. Accuracy curve of HAN

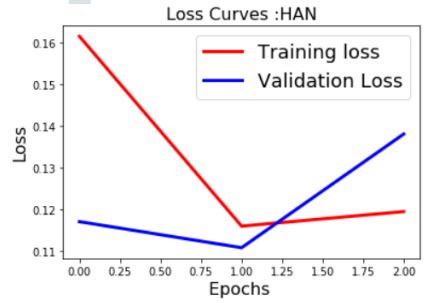


Figure 9. loss curves of HAN

```
In [20]: history=model.fit(x_train, y_train, validation_data=(x_val, y_val),epochs=3, batch_size=5,callbacks=[cp])

Train on 375129 samples, validate on 41680 samples
Epoch 1/3
375129/375129 [==============================] - 144415 384ms/step - loss: 0.7354 - acc: 0.7942 - val_loss: 0.7433 - val_acc: 0.
7944

Epoch 00001: val_acc improved from -inf to 0.79441, saving model to model_cnn.hdf5
Epoch 2/3
375129/375129 [==============================] - 424825 1134ms/step - loss: 0.9837 - acc: 0.7856 - val_loss: 1.0717 - val_acc:
0.7958

Epoch 00002: val_acc improved from 0.79441 to 0.79578, saving model to model_cnn.hdf5
Epoch 3/3
375129/375129 [==============================] - 15220s 41ms/step - loss: 1.1924 - acc: 0.7789 - val_loss: 1.1172 - val_acc: 0.
6983

Epoch 00003: val_acc did not improve from 0.79578
```

Figure 10. Iterations after applying CNN

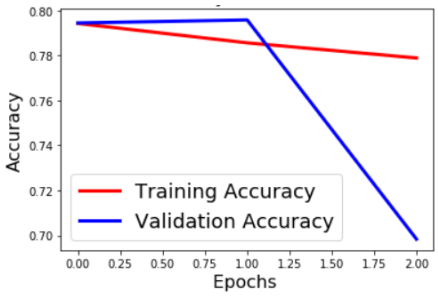


Figure 11. Accuracy curves for CNN

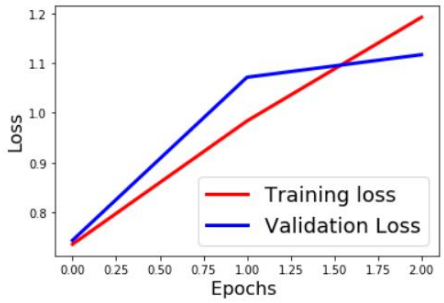
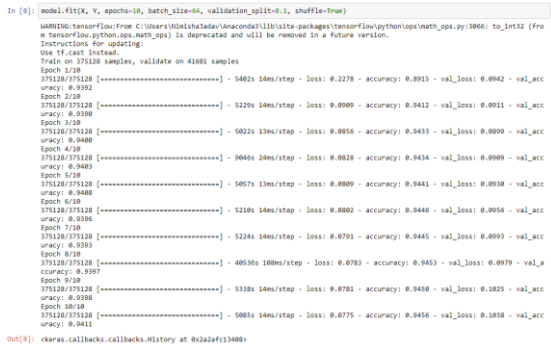


Figure 12. loss curves for CNN



```
In [8]: model.fit(X, Y, epochs=10, batch_size=64, validation_split=0.1, shuffle=True)

WARNING:tensorflow:From C:\Users\Himisha3adav\Anaconda3\lib\site-packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (fro
m tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 375128 samples, validate on 41681 samples
Epoch 1/10
375128/375128 [==============================] - 5402s 14ms/step - loss: 0.2278 - accuracy: 0.8915 - val_loss: 0.0942 - val_acc
uracy: 0.9392
Epoch 2/10
375128/375128 [==============================] - 5229s 14ms/step - loss: 0.0909 - accuracy: 0.9412 - val_loss: 0.0911 - val_acc
uracy: 0.9390
Epoch 3/10
375128/375128 [==============================] - 5022s 13ms/step - loss: 0.0856 - accuracy: 0.9433 - val_loss: 0.0899 - val_acc
uracy: 0.9400
Epoch 4/10
375128/375128 [==============================] - 9046s 24ms/step - loss: 0.0828 - accuracy: 0.9434 - val_loss: 0.0909 - val_acc
uracy: 0.9403
Epoch 5/10
375128/375128 [==============================] - 5057s 13ms/step - loss: 0.0809 - accuracy: 0.9441 - val_loss: 0.0930 - val_acc
uracy: 0.9408
Epoch 6/10
375128/375128 [==============================] - 5210s 14ms/step - loss: 0.0802 - accuracy: 0.9446 - val_loss: 0.0956 - val_acc
uracy: 0.9396
Epoch 7/10
375128/375128 [==============================] - 5224s 14ms/step - loss: 0.0791 - accuracy: 0.9445 - val_loss: 0.0993 - val_acc
uracy: 0.9393
Epoch 8/10
375128/375128 [==============================] - 40536s 108ms/step - loss: 0.0783 - accuracy: 0.9453 - val_loss: 0.0979 - val_a
ccuracy: 0.9397
Epoch 9/10
375128/375128 [==============================] - 5338s 14ms/step - loss: 0.0781 - accuracy: 0.9450 - val_loss: 0.1025 - val_acc
uracy: 0.9398
Epoch 10/10
375128/375128 [==============================] - 5085s 14ms/step - loss: 0.0775 - accuracy: 0.9456 - val_loss: 0.1038 - val_acc
uracy: 0.9411
Out[8]: <keras.callbacks.callbacks.History at 0x2a2afc13408>
```

Figure 13. Iterations after applying LSTM

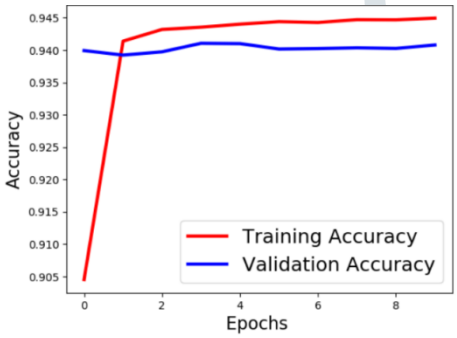


Figure 14. Accuracy curves for LSTM

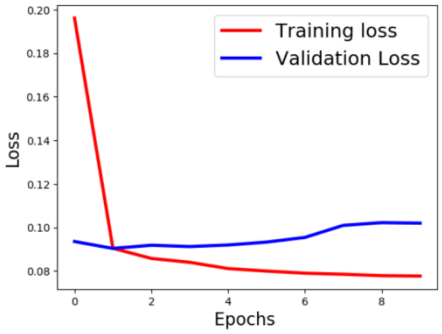


Figure15. loss curves for LSTM

Algorithm	Time/Epoch	Training Accuracy	Validation Accuracy
LSTM	9113s-10epochs	94.56	94.11
HAN	159274s-1 epoch	93.56	93.80
SVM	23856s	88.00	-
CNN	24047s-3 epochs	77.89	69.83

Table 1. Summary of training and validating accuracy of 4 algorithms

By looking at this table we can conclude that Out of the four models, LSTM gives us output in less amount of time with highest validation and training accuracy.

After training when we test our model we got output as shown in following figure.

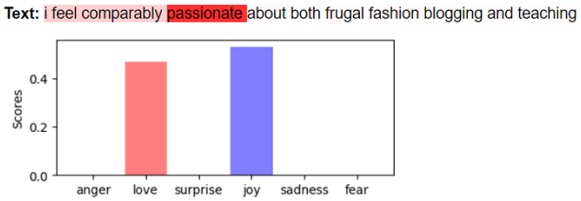


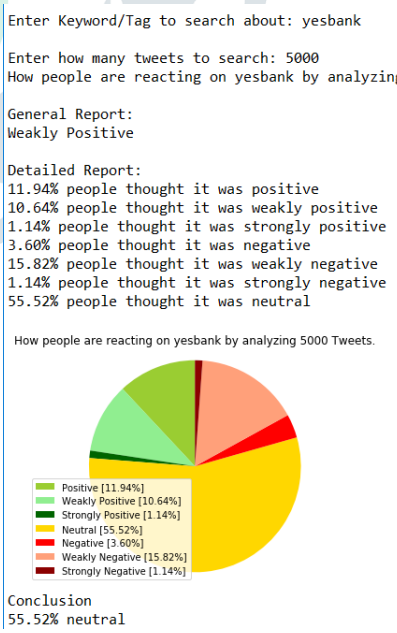
Figure 16. Extracting emotion from sentence

b) Results for real-time data:

We have performed emotion analysis for real time data as well, for different topics and we change the tweet count every time. For this real-time data we choose current most discussed topic on twitter like CAA bill, Coronavirus outbreak and Yesbank controversy.

The Twitter API platform offers two options for streaming real-time Tweets. Each option offers a varying number of filters and filtering capabilities.

After performing emotion analysis on real time data we will get output as shown in following figures.



```
Enter Keyword/Tag to search about: yesbank

Enter how many tweets to search: 5000
How people are reacting on yesbank by analyzing 5000 tweets.

General Report:
Weakly Positive

Detailed Report:
11.94% people thought it was positive
10.64% people thought it was weakly positive
1.14% people thought it was strongly positive
3.60% people thought it was negative
15.82% people thought it was weakly negative
1.14% people thought it was strongly negative
55.52% people thought it was neutral
```



```
Conclusion
55.52% neutral
```

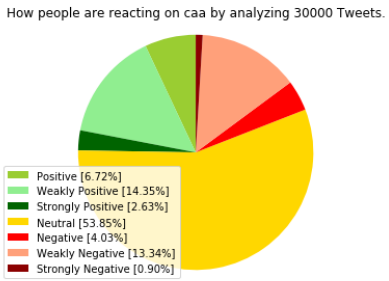
Figure 17. Result for 5000 real-time tweets

```
Enter Keyword/Tag to search about: caa

Enter how many tweets to search: 30000
How people are reacting on caa by analyzing 30000 tweets.

General Report:
Weakly Negative

Detailed Report:
6.72% people thought it was positive
14.35% people thought it was weakly positive
2.63% people thought it was strongly positive
4.03% people thought it was negative
13.34% people thought it was weakly negative
0.90% people thought it was strongly negative
53.85% people thought it was neutral
```




```
Conclusion
53.85% neutral
```

Figure 18. Result for 30,000 real-time tweet

Topic → No of tweets↓	CAA	Coronovirus	yesbank
5000	65.38 % neutral	40.22% neutral	55.52% neutral
10000	62.88% neutral	44.77% neutral	53.84% neutral
20000	54.16% neutral	40.58% neutral	58.84% neutral
30000	53.85% neutral	39.71% neutral	60.06% neutral

Table 2. Emotion analysis for real-time data

VI. CONCLUSION AND FUTURE SCOPE

We have implemented emotion detection method to extract emotion from text input. At first, we preprocess the input data using different NLP tasks then we apply feature selection function to our learning models for more accuracy. While performing the experiment we observe the significant change in accuracy rate of models before and after applying counter vector function. Before applying counter vector to learning models the accuracy rate of linear SVM was 54\% but after applying counter vector it increased upto 78.61\%.

Experiments have proved that process of extractinghuman emotion from textual data is dependent on the context for which word has been used in the sentence. As we know, Emotion Recognition also introduces different challenges to the work in the sense of emotion and the ways to express emotion is subjective.

In the future works, one can extend this emotion extracting to other tasks such as affective computing and sentiment analysis. It is possible to improve the performance of this model by using the larger emotion word embeddings. Also same emotion recognition model can be implement for data which is in audio format.

VII. REFERENCES

[1] MahiraKirmani, MudasirMohd, Nnida Manzoor Hakak, Mohsin Mohd, "Emotion analysis: A survey," 2017, IEEE.

[2] Chris Dyer, Xiaodong He, Alex Smola, Eduard Hovy, Zichao Yang, Diyi Yang, "Hierarchical attention networks for document classification," Proceedings of the 2016 conference of the North American chapter ofthe association for computational linguistics: human language technologies, 2016.

[3] Keun Ho Ryu,ErdenebilegBatbaatar, Meijing Li,"Semantic-emotion neural network for emotion recognitionfrom text," 2019, IEEE

[4] Dr. Ashok Kumar R, Priyanka H S, Ramya B V, "Classificationmodel to determine the polarity of moviereview using logistic regression," International ResearchJournal of Computer Science (IRJCS), 2019.

[5] Adarsh S R., "Enhancement of text based emotionrecognition performances using word clusters," 2019.

[6] Hazem Hajj, Shady Elbassuoni, Shadi Shaheen, Wassim El-Hajj, "Emotion recognition from text based on automatically generated rules," International Conference on Data Mining Workshop, 2014.

[7] Md. Humayan Ahmed, Romana Rahman, Tajul Islam,"Detecting emotion from text and emoticon," 2017.

[8] Gurbakash P, Deepak K D, Sudhanshu T, M. Vijaya Raju, "A novel approach for detecting emotion in text," Indian Journal of Science and Technology,Vol 9(29), 2016.

[9] Prof. Saritha Khethawat, Shiv Naresh Shivhare, "Emotiondetection from text,".

[10] Chetan R. Chopade, "Text based emotion recognition:A survey," International Journal of Science and Research(IJSR), 2015.

[11] Nicole Novielli, Fabio Calefato, Filippo Lanubile,"Emotxt: A toolkit for emotion recognition from text,"Seventh International Conference on Affective Computingand Intelligent Interaction Workshops and Demos(ACIIW), 2017.

[12] Raúl Oramas-Bustillos, Maria Lucia Barron-Estrada, Ramon Zatarain-Cabada, Sandra Lucia Ramírez-Ávila, "A corpus for sentiment analysis and emotion recognitionfor a learning environment," IEEE, 2018.