Sentiment Analysis Using Deep Learning

Conference Paper · February 2021

DOI: 10.1109/ICICV50876.2021.9388382

4 authors, including:

Shilpa C p.c
SCMS Group of Educational Institutions

Susmi Jacob
SCMS School of engg & Technology

Proceedings of the Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV 2021).
DVD Part Number: CFP21ONG-DVD, 978-0-7381-1182-7

SENTIMENT ANALYSIS USING DEEP LEARNING

Shilpa P C
Computer Science & Engineering
SCMS School Of Engineering and Technology
shilpapc@scmsgroup.org

Rissa Shereen
Computer Science & Engineering
SCMS School Of Engineering and Technology
risasherinc@gmail.com

Susmi Jacob
Computer Science & Engineering
SCMS School Of Engineering and Technology
susmijacob@scmsgroup.org

Vinod P
Computer Science & Engineering
SCMS School Of Engineering and Technology
vinodp@scmsgroup.org

***Abstract*— Emotion recognition from text is crucial Natural Language Processing task which can contribute enormous benefits to different areas such as artificial intelligence, human interaction with computers etc. Emotions are physiologic thoughts engendered in human reactions to the events. Analysis of these emotions without facial and voice modulation are critical and requires a supervisory approach for proper interpretation of emotions. In spite of these challenges, it's essential to acknowledge the human emotions as they progressively communicate using mistreatment text through social media applications such as Facebook, Twitter etc. In this paper, we propose a sentimental classification of multitude of tweets. Here, we use deep learning techniques to classify the sentiments of an expression into positive or negative emotions. The positive emotions are further classified into enthusiasm, fun, happiness, love, neutral, relief, surprise and negative emotions are classified into anger, boredom, emptiness, hate, sadness, worry. We experimented and evaluated the method using Recurrent Neural Networks and Long short-term memory on three different datasets to show how to achieve high emotion classification accuracy. A through evaluation shows that the system gains emotion prediction on LSTM model with 88.47% accuracy for positive/negative classification and 89.13% and 91.3% accuracy for positive and negative subclass respectively.**

keywords-Twitter Sentiments, Emotion Classification, Deep Learning techniques, Long Short Term Memory, Recurrent Neural Networks

I. INTRODUCTION

Twitter is a social networking web site where members can post messages in the form of "tweets". This is a platform where individuals can share ideas or sentiments on diverse subjects, fields or themes.It is a collection of user thoughts and sentiments spanning across various topics including standard net articles and net blogs. The quantity of pertinent data is bigger for twitter, when contrasted with former social media and blogging platforms. When compared to other blogging sites, the response rate on Twitter is much more quicker. Sentiment analysis is widely utilized by different parties such as shoppers or marketers to gain insights into merchandise or understand the market trends[1].

Also, this is critical in predicting the exchange rate or the product rating of a selected organization. This is done by analyzing the sentiment of the public towards the corporation with regard to time and positioning[2]. There is a need for a social science tool in order to understand the importance of public sentiments and thereby the market value of the companies. Firms can also estimate how well their product is moving in the market. Furthermore, it will help to analyze the positive and negative feedbacks about the products. Since Twitter permits transfer stream of geotagged tweets from particular areas of explicit locations, organizations can easily understand area trends. This information will assist the corporation in analyzing different responses so that they can market their products in an optimized manner. The firms can also market their products using appropriate techniques, understand latest trends and develop market demanding products.

As the machine learning algorithms have drastically improved in the recent past, we can find a better way to enhance the accuracy of our sentiment analysis predictions. Machine learning is strongly connected to statistical analysis which focus on creating predictions with the help of digital computers. The investigation of a numerical improvement made the application to change to the circle of AI (Artificial Intelligence). Data processing has a major role in research areas and in the field of study among AI. While considering its application across various businesses and research problems, AI is also considered as prognostic systematic.

Sentiment analysis on Twitter uses several approaches where Deep learning have gained great results in emotion recognition. This paper focus on classifying user emotion in Twitter messages using deep learning models such as LSTM (Long short Term Memory) and RNN (Recurrent Neural network). From each word of the input tweets, the semantic word vectors were take out from the lexical words using the method word2vec. The features extracted were combined and fed into an LSTM and RNN to train and predict the emotion classification labels of the tweet into negative or positive sentiments. The experiment on the data shows LSTM technique has a better capability of predicting sentiment classification accuracy.

II. Background And Related Work

Various studies were carried out on fully automated system that perform feature extraction from datasets without human intervention [3, 4]. The resultant studies classified the features as positive, negative and neutral[5]. The sentiment analysis for the classification was done on the phrasal level, using new features like DAL scores and n-grams etc. The polarity [6, 7] of syntactic details were used as features. However, this system came with a limitation; it required an accurate expression boundary to capture the precise emotion prediction. Also, it does not handle polysemy owing to the difference in producing words using DAL which is not the part of the speech.

An alternative classification used is the distant supervision for sentiment analysis [8, 9]. The dataset used include twitter messages with different emoticons. These emoticons are considered as noisy label. The classification algorithms used are Naive Bayes, Maximum Entropy and Support Vector Machine. The main limitation noticed here was the metric of accuracy as compared to other learning algorithms. The classification process done with TreeTagger for POS(part of speech)tagging along with Naive Bayes and N-Grams have different distribution for the emotion categories. Yet, the multilingual collection of data is not included for the classification [10].

Sentiment classification of twitter messages [11] by using polarity predictions to categorize reviews are taken from different sites and termed as noisy labels. These labels were used for developing a training model. Also, 1000 tweets were grouped manually for tuning and another 1000 tweets were taken for testing. However, they did not explain about the data collection for testing. Nevertheless, they planned the utilization of features for tweets following particular patterns like re-posting of tweets (retweet), hashtags, link of replies, punctuation and exclamations. Additional options like previous polarity of words and POS tags were used.

In 2004, a sentiment analysis was done on the feedback data collected from a survey. They set their sights on analysis for identifying the importance of semantic features like push tags in Natural language Processing. They performed substantial feature selections and analysis, which could help demonstrate that the analysis of verbal features contributes to the classifier accuracy to make it more ideal [12].

In 2007, the authors developed a corpus by making use of different emotional texts from web blogs for sentiment analysis and use the emotional blogs from web posts as indicators of users' mood. The categorization of sentimental emotions at sentence level went through analysis phase with the assistance of SVM(Support Vector Machine) and CRF(Conditional Random Fields). Also, some investigations created with many methods to work out the sentiment of messages at document level [13, 14].

Another prominent initiative, which have been done on the sentiment analysis was the usage of word lengthening to identify emotional sentiments in micro blogs such as Twitter [15]. For the purpose of word lengthening they created a method which was fully independent of the user intervention that provides a great support to word lengthening. They focused their study on the sentimental part of Twitter messages as well as other social media messaging applications.

Despite the traditional machine learning classification algorithms, LSTM has proved the efficiency in obtaining excellent accuracy for emotion classification [16]. If the classification is done using simple RNN there occur some exploding and decaying of result accuracy during the phases of back propagation. This problem was rectified using an advanced RNN, LSTM with its more complex internal structure which includes memory cells that permits architecture to recollect the information that have been stored in the memory cells long back [17].

Various studies have been done on sentimental and content analysis in a combined manner for identifying and interpreting human emotional messages [18]. In the course of few years, the sentimental computing has set foot in the area of machine learning as the social media have been used by different abusers. Also, these are required abilities for many Human computer interaction applications.

Apart from the traditional parsers that was based on normal searching, a greedy parsing technique named Transition based dependency is used for the classification. This parser will give more accuracy of the classification results. However, due to the error propagation the result become worse when compared to traditional searching parser. Another limitation is that the classification can only be performed on a small size of data [19]. The social media has become very popular and inevitable medium for the companies to market their product according to the needs [20]. Hence, it is very important to analyse the correct interpretation of public opinion [21]. Different deep learning techniques are available for finding the accurate polarises opinion. This paper did the analysis of different deep learning classification based on the accuracy [22].

In fact, the iterations of the SemEval, Twitter sentiment analysis vying have already entrenched their potential over other approaches [23]. To build an accurate sentimental classifier model for emotion recognition. They combined the features of both CNN and RNN models in the message level and phrase level.

The studies were conducted on models which include the most advanced twitter sentiment classifiers using CNN (convolutional neural networks) as well as LSTMs networks. The analysis shows a noticeable change in the accuracy levels for emotion recognition [24]. This analysis was based on LSTM paid attention to the information in the message-level and topic level while doing the sentiment analysis [25].

As twitter becomes an important platform for microblooging, it is also used to analyze the public emotions and opinions of various fields. Recent studies were conducted on tweets for collecting the public opinion about elections in India using NRC Dictionary based approach and Lexicon based approach [26].

We studied various models that classify the emotions as

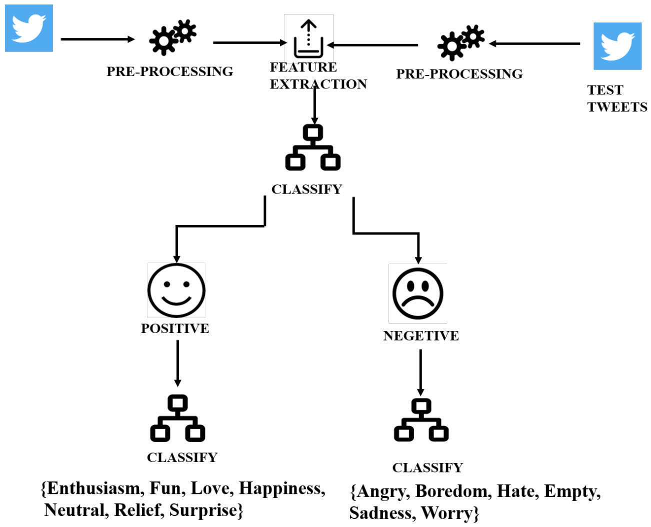

Fig. 1: System Architecture

positive, negative and neutral or other models classifying sentiments into many emotions. In our model we combined both positive and negative models. Our model first classifies the emotion to positive or negative and then further to other emotions accordingly.

III. System Architecture

Pre-processing of train data uses NLP, which is a technique accustomed to perceive the computer information and handle the human interactions. The text comments that are given to the model is additionally being pre-processed. Both informations are passed to the sentiment library where the feature extraction of the pre-processed information is being done. From this, we tend to get the trained model for the knowledge sets. The system architecture is depicted in Figure 1. The classification is finished using LSTM that is another version of Recurrent Neural Network.

The classifier takes the input and then classify them as positive and negative emotions. These data are then passed onto the another classifier for further classification of positive and negative emotions. The positive features are then classified as enthusiasm, fun, love, happiness, neutral, relief and surprise. The negative features are classified into angry, boredom, hate, emptiness, sadness and worry.

The classifier then predicts the output of the test input, which provides the results of the model. The text comments from the tweets undergo the pre-processing, since it contains URL id. Since we tend not to think about any address, we eliminate all the URLs and avoid all unwanted areas. These processes are done in the pre–processing stage.

IV. Methodology

A. Dataset

Though data sets for the experiments are widely available across various social networking platforms, the manual determination of sentiments of different tweets is a challenging task. It requires a high level of expertise in the area to perform a vigilant analysis of these tweets and generate accurate results. Existing public datasets having url are rather limited due to these challenges.

For the evaluation of experiments described in this paper, we used three different datasets. Primarily, we classified the given comments of the first dataset as either positive or negative. The second dataset contains only positive comments with labels of different positive emotions. This dataset is used to further classify the positive comments. The third and the last dataset contains only negative comments with labels of different negative emotions which is used to further classify the comments that are classified as negative. We have 100000 datum in our positive-negative classification.

For positive emotions, we further classified the tweets into Enthusiasm, Fun, Love, Happiness, Neutral, Relief, Surprise. For negative emotion, they are again categorized into Angry, Boredom, Hate, Empty, Sadness, Worry.

B. Pre-processing

The messages from Twitter are too informal and has different styles of using tweets based on the nationality, origin, age and the gender of the user. Therefore, tweets taken from twitter generally result in a noisy data set of unwanted emoticons and symbols. Twitter users will use different kinds of special characters like sending the tweets again, which are termed as re-tweets, emoticons, personalized wording etc. all of which are to be suitably extracted. Hence for creating new datasets, raw twitter data has to be normalized for easiness of the classifier algorithms to perform the emotion analysis. There are different pre-processing methods available for tweets [2]. The main reason for using the pre-processing task is to reduce the noise and size of the messages. Initially we perform few general and mandatory pre-processing task which include conversion of words to lower case. Also, there are tweets with more dots as we use informal conversation. So, the additional dot is replaced with a space. At the end, we remove unwanted space and special characters.

Multiple spaces can be avoided by replacing them with a single space. The url from the twitter messages will not be used in the analysis as it leads to a sparse result. Hence, we replaced all the URL in the tweets with the expression `((www\. [\S] +)))`.

Handle is another special character used in tweets to mention other user names in their messages. The users commonly use the tagging of another user as `@name_user`. As we are eliminating all unwanted symbols from the data sets, we replace all user tagging like the `@name_user` with the word USER-MENTION. The equivalent expression used to match user tagging is `@[\s] +.2`.

It is very common to use different types of images in the tweets. These include smiley symbols, hand gestures, etc. Since the social media sites have a wide range of such images, it is very difficult to find equivalent expression for each among them. However, we cannot exclude the images as they play a major role in the emotion transfer and therefore, we use two types of substitutions for positive and negative emotions as EMOJ-POSITIVE and EMOJ-NEGATIVE respectively.

Hashtags are used to categorize Tweets and messages on similar or common area so that people can easily find the area of interest. They are phrases without any space and preceded by a # symbol. When we click on the hashtags it will move to other tweets that also have the same hashtag. So, these are from the same area of interest. For replacing and giving a common format for these hashtags, we will remove the hash symbol and replace the hashtag with the word which comes after the hash symbol. The regular expression used to match the hashtag is `# (+)` .

It is common to resend the tweets which we have already received from other users for many reasons. This gives more flexibility to the use of tweets and make this platform more interesting. Normally these retweets begin with the letter RT and during the pre-processing phase these RT will be removed as it is not relevant for the classification of tweet emotions. The retweets are represented as `\brt\b`.

After the pre-processing of the URL, Hashtags and retweets, the main part of tweet processing is the clearing of punctuations in the tweets. That is, in a tweet message like “Oh my God!! I cannot believe this.” we will remove the punctuation and rephrase the tweets. For getting better features for the classification we need to remove all the punctuation, like `’’‘! ()-[]{};:’"\, <>. ?` @%^*_ ”’ /.

Twitter is a platform where users can convey their feelings through different expressions. Many people will post tweets like “I am sooooo surprised” to highlight their emotions. But during the word level processing of tweets we must ignore these duplications. The other word level processing includes the removal of “-” and “’” which is commonly used in typing t-shirts and can’t. These are represented as ”tshirt” and ”can”t in a more generalized manner. The complete processing steps are shown in the Figure 2.

V. Experiments

The twitter data set has been collected and used across different data sets for positive and negative emotions to evaluate the proposed system. We normalized the tweets to make the data suitable for pre- processing. The pre-processing steps are mandatory to avoid unwanted symbols and get the tweets in a normalized manner. Finally, the combined model of RNN and LSTM are used for classification to get better accuracy. Figure 3 shows the most typical processes involved in a sentiment classification.

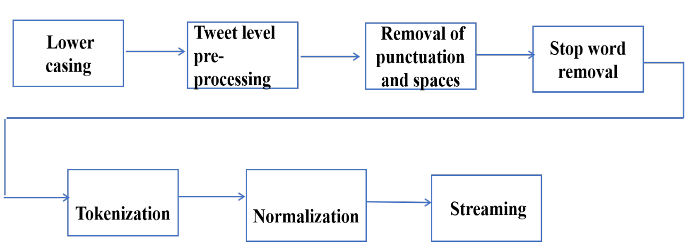

Fig. 2: Processing steps

A. Dataset

For our work and experiments in the paper we took three data sets. The first data set was taken from an existing dataset consisting of one lakh tuples. Moreover, the dataset consists of data for the positive and negative classification. The second data set is for the positive classification which consists of 23938 tuples. The positive data are classified into 7 class labels namely Enthusiasm, Fun, Love, Happiness, Neutral, Relief, Surprise. The last and third data set is for the negative classification which consists of 26889 tuples. The negative data are classified into 6 classes which are Angry, Boredom, Hate, Empty, Sadness, Worry. The grouping of the sentiment labels in datasets are illustrated in Table I.

B. Feature Selection

TF-IDF

TF-IDF (Term Frequency Inverse Document Frequency) is used in natural language processing for identifying the important words or rare words in a text data. Term frequency converts the words that are in string format for numerical formatted data so that the machine learning models can understand the information. Term Frequency TF is used to find out the frequency of occurrence of words that we have for the classes. In our data set the feature is words. The frequency of each word in the dataset which is being calculated using term frequency.

$$\text{TF(I,j)} = \frac{term\ i\ frequency\ in\ document\ j}{total\ words\ in\ doc\ j}$$

Where i is the frequency of terms in document j. A table with frequency of terms in the document will be created in the memory. IDF (Inverse Document Frequency) is used to get the most important or least occurring words in the document. IDF helps us to take meaningful words from the document. TF gives us the highest degree words while IDF

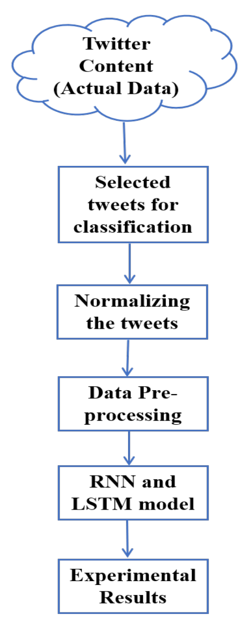

Fig. 3: Experimental setup

TABLE I: Description of datasets

Dataset	**No of instance**	Positive			Negative			
Dataset1	100000	56467			43533			
Dataset2	11102	angry	Boredom	Empty	Hate	Sadness		worry
		611	1678	1826	1022	1507		4458
Dataset3	11261	Enthusiasm	Fun	Happiness	Love	Neutral	Relief	surprise
		760	1775	5208	3844	962	1526	2186

helps us to get the lowest occurred words by taking the logarithm of the values.

$$\text{IDF(i)} = \log 2\left(\frac{(total\ number\ of\ documents)}{(number\ of\ documents\ with\ term\ i)}\right)$$

Finally, we find that the product of both the TF and IDF matrices is the normalized weights which is the TF-IDF output. In this way we get the numerical input for the machine learning model. TF-IDF is used to represent text with a BoW (Bag of Words).

Doc2vec

The doc2vec vector algorithm performs considerably well for sentence similarity tasks. However, if the input corpus includes a lot of words with misspellings like tweets, this algorithm may not be an ideal choice. We used Doc2Vec method which is used for the vectorization of documents. This is an improved version of Word2Vec. This method was not desirable in the works where a lot of misspellings of words occur. It is better to convert words to vectors and then use these vectors to create the vector format of the whole document. Doc2Vec is used to represent text with word vectors.

C. Classification

As an extension of RNNs, Long Short-Term network is introduced with additional memory features. The architecture and the internal structure of LSTM is shown in Figure 4. In LSTMs, repeated cells are connected in an exceedingly specific method to avoid vanishing and exploding gradient problems. LSTM is an advanced form of Recurrent Neural Network. LSTM includes recurrently connected blocks which are the memory units. The LSTM remove or add information to the cell state. They are regulated by structures called gates. Gates are the structures that decides whether to pass the information or not. So LSTM helps us to classify our tweets by also using the long range of dependencies.

The LSTM has flexibility to get rid of or append information to the cell which keeps track of the sate information, and these are controlled by gates. Gates allow the information to pass through the network. The segment layer has the outputs range between 1 and 0 which denotes the information which are passing through. If the obtained value is 0, it tells us that nothing is passing through the gate and a value of 1 describes that everything can be pass through.

Initially, we have to identify the features to be fed to the cell state. The forget layer in the LSTM architecture is responsible for this decision whether to forget or accept the features. By considering the value of h_{t-1} and x_t it will give two values as output. That is, the output values can either be a 0 or a 1 for each cell state. If the value is 1, it represents the acceptance of the feature and 0 represents the avoidance of the information. In this work we must accept or reject the feature words from the data sets to classify the emotion. In each cell state, the new emotion word has to be accepted or

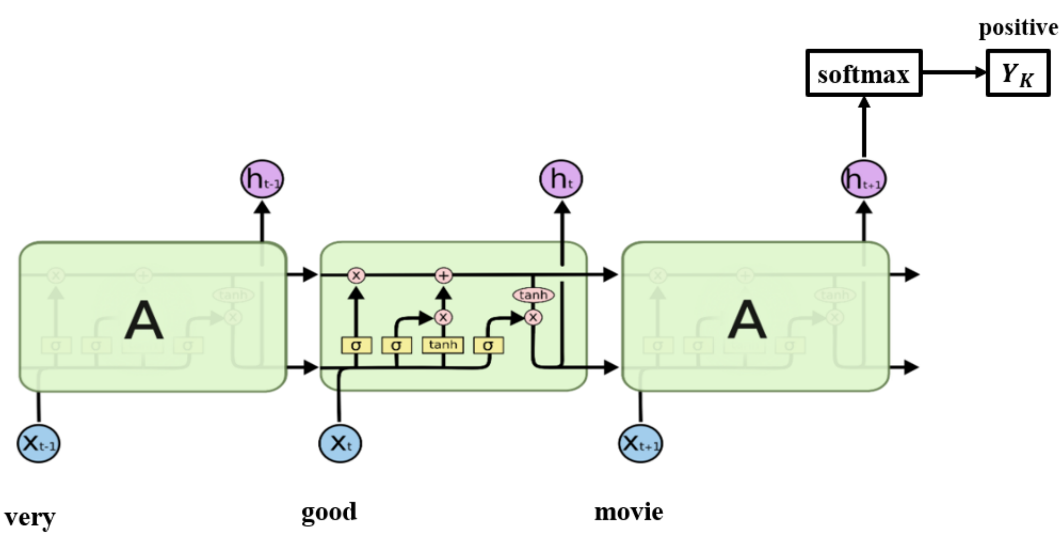

Fig. 4: LSTM Architecture.

rejected based on the emotion of the tweets.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{1}$$

The next part of classification is the identification of new information that need to be stored in the cell state. In combination with the input gate layer, the sigmoid layer will decide which data to be updated. The vector for the newly updated data c_t is created by the tanh layer and stored in the cell. After that we update the cell state. That is, in the case of sentimental analysis, we update the state with the new emotion which have forgotten in the previous stage.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\check{C}_t = \tanh(W_C.[h_{t-1}, x_t] + b_C) \tag{3}$$

Finally, we update the cell state $c_{t\text{-}1}$ with the new state CT which have already updated. After forgetting the old state we multiply the state by ft. For getting the new candidate value we add i_t*c_t to the previously obtained value. This is the final step for forgetting and updating the state.

$$C_t = f_t * C_{t-1} + i_t * \check{C}_t \tag{4}$$

The final output is based on the filtered cell state. When we execute the sigmoid layer, it gives the output which we are going to produce. We can only output, the desired part by passing it through the tanh layer and multiplying with the result of the sigmoid layer. This facilitates us to generate the output that we desire to produce.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = \tanh(C_t) * O_t \tag{6}$$

Test cases for positve and negative emotions are demonstarted here with an example.

Case 1: Positive

For example: “omg!!! It is surprising,” it is categorized as positive. Further, it is categorized based on the percentage in correspondence to each emotion. Here we get 40.91% for surprise, 28.27% of relief, 11.64% for fun, 7.22% of neutral and less than 5 percentage for another three emotions each, which shows that the emotion surprise is more dominant in this comment. Hence it is categorized as a surprise.

Case 2: Negative

For example: “I am so panic these days”, it is categorized as negative emotion statement. Further, it is categorized based on the percentage of correspondence to each emotion. Here we get 78.43 percentage to worry, only 14.82% is for sadness, 5.49% of empty and other three emotions together obtain only 1.26% of the total. So, it is categorized as worry.

VI. Results and discussions

We used the dataset Kaggle for our experimental setup. The evaluation metrics used in our experiment is the accuracy in classification of positive and negative tweets. For the purpose of comparison, we examined various accuracy obtained with same models, RNN and LSTM for other datasets used by Ming-Hsiang Su et al., 2018 and Zhao Jianqiang et al., 2017. Table II shows the accuracy metrics for LSTM and RNN over different classifications.

The results obtained from the experiments clearly recommend that LSTM model acquire better accuracy compared to RNN, CNN classifiers. The LSTM model was successfully trained with the twitter dataset. We have three datasets for positive-negative, negative and positive classification respectively. We analyzed the system using the performance measure accuracy. The accuracy was measured by the formula:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

In the measurement of accuracy, there have chances of miss interpretation of the prediction. That is, the actual defected results are being identified as true cases and is termed as True Positive, represented as TN. Also, some cases which are correct are being recognized as negative, called as False positive and is represented as FP.

Figure 5 depicts the accuracy comparison of both LSTM and RNN. For the positive/negative binary classification we obtained a training accuracy of 88.47% and testing accuracy of 79.16%. For the negative classification we obtained a training accuracy of 89.13% and testing accuracy of 87.46%. For the positive classification we obtained a training accuracy of 91.32% and testing accuracy of 90.75%. The results significantly proves the efficiency of LSTM based sentiment analysis over RNN.

Since the proposed method incorporated both semantic word vector and the emotional word vector, the performance of the system is improved remarkably. The LSTM-based model learning used Doc2Vect for feature extraction in the word sequence. Hence, its performance is better than the RNN-based structure that modelled the spatial relationship of the word sequence. When RNN is taken into account, the experimental results show that it suffers from a vanishing gradient problem, where the model fails to learn and adapt even with significant changes in the weights. These problems are solved effectively using LSTM, which is an improved version of RNN with robust computing proficiency and storage capacity. Moreover, LSTM classifies emotion of long sentences effectively compared to the conventional RNN.

VII. Conclusion

In this paper, we modelled a system for sentiment analysis of twitter messages. The tweets we consider in the analysis are, a mixture of different words and emoticons. We modeled the classifier With deep learning techniques such as RNN and LSTM. In order to gain better accuracy, we incorporated different feature selection methods like TF-IDF

TABLE II: Accuracy of different models for positive/negative using LSTM, CNN and RNN

Method	Model	Classification	Accuracy (%)
Proposed Method	LSTM	Positive/Negative	88.47
		Positive subclasses	89.13
		Negative subclasses	91.3
	RNN	Positive/Negative	83.21
		Positive subclasses	81.24
		Negative subclasses	87.02
Ming-Hsiang Su et al.,2018[27]	LSTM	Positive/Negative	70.66
	CNN	Positive/Negative	65.33
Zhao Jianqiang et al.,2017 [28]	CNN	Positive/Negative	87.62

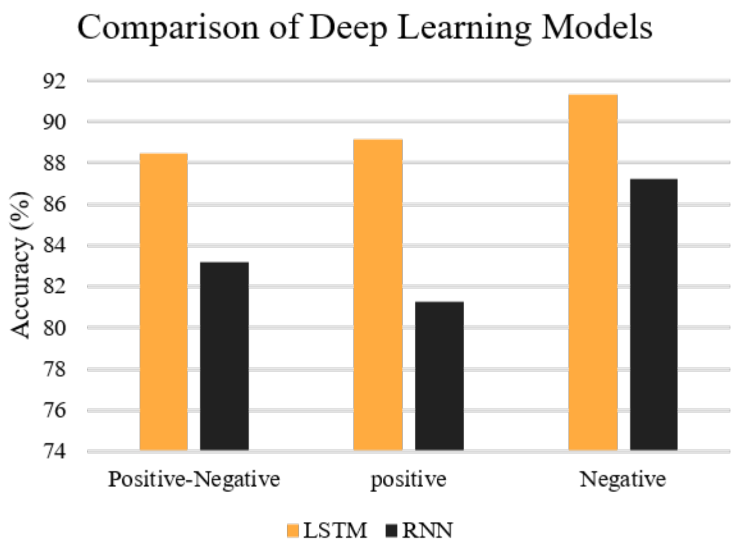

Fig. 5: Accuracy obtained for sentiments with LSTM and RNN.

and Doc2Vect. The feature extraction generates a vector that is been given as input to the classification model. Our model achieved better results in the task of classifying the twitter emotional messages. As a future work, the analysis of the personality of the users from their tweets are need to be investigated, so that the system can be more personalized.

References

[1] N. Yadav, O. Kudale, S. Gupta, A. Rao, and A. Shitole, "Twitter sentiment analysis using machine learning for product evaluation," in *2020 International Conference on Inventive Computation Technologies (ICICT)*, pp. 181–185, IEEE, 2020.

[2] D. Ramachandran and R. Parvathi, "Analysis of twitter specific preprocessing technique for tweets," *Procedia Computer Science*, vol. 165, pp. 245–251, 2019.

[3] A. Agarwal, F. Biadsy, and K. Mckeown, "Contextual phrase-level polarity analysis using lexical affect scoring and syntactic n-grams," in *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pp. 24–32, 2009.

[4] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," *arXiv preprint cs/0409058*, 2004.

[5] S. A. El Rahman, F. A. AlOtaibi, and W. A. AlShehri, "Sentiment analysis of twitter data," in *2019 International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–4, IEEE, 2019.

[6] M. Hu and B. Liu, "Mining and summarizing customer reviews. kdd'04: Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining," 2004.

[7] S.-M. Kim and E. Hovy, "Determining the sentiment of opinions," in *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pp. 1367–1373, 2004.

[8] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N project report, Stanford*, vol. 1, no. 12, p. 2009, 2009.

[9] A. Bermingham and A. F. Smeaton, "Classifying sentiment in microblogs: is brevity an advantage?," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1833–1836, 2010.

[10] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining.," in *LREc*, vol. 10, pp. 1320–1326, 2010.

[11] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Coling 2010: Posters*, pp. 36–44, 2010.

[12] M. Gamon, "Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis," in *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pp. 841–847, 2004.

[13] C. Yang, K. H.-Y. Lin, and H.-H. Chen, "Emotion classification using web blog corpora," in *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, pp. 275–278, IEEE, 2007.

[14] A. Sungheetha and R. Sharma, "Transcapsule model for sentiment classification," *Journal of Artificial Intelligence*, vol. 2, no. 03, pp. 163–169, 2020.

[15] S. Brody and N. Diakopoulos, "Cooooooooooooooollllllllllllll!!!!!!!!!!!!!! using word lengthening to detect sentiment in microblogs," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 562–570, 2011.

[16] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[17] N. F. Alshammari and A. A. AlMansour, "State-of-the-art review on twitter sentiment analysis," in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pp. 1–8, IEEE, 2019.

[18] S. Niklander and G. Niklander, "Combining sentimental and content analysis for recognizing and interpreting human affects," in *International Conference on Human-Computer Interaction*, pp. 465–468, Springer, 2017.

[19] D. Chen and C. D. Manning, "A fast and accurate dependency parser using neural networks," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 740–750, 2014.

[20] C.-C. Wang, M.-Y. Day, C.-C. Chen, and J.-W. Liou, "Temporal and sentimental analysis of a real case of fake reviews in taiwan," in *2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 729–736, IEEE, 2017.

[21] B. Bhavitha, A. P. Rodrigues, and N. N. Chiplunkar, "Comparative study of machine learning techniques in sentimental analysis," in *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 216–221, IEEE, 2017.

[22] L. M. Rojas-Barahona, "Deep learning for sentiment analysis," *Language and Linguistics Compass*, vol. 10, no. 12, pp. 701–719, 2016.

[23] A. Severyn and A. Moschitti, "Unitn: Training deep convolutional neural network for twitter sentiment classification," in *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pp. 464–469, 2015.

[24] M. Cliche, "Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms," *arXiv preprint arXiv:1704.06125*, 2017.

[25] C. Baziotis, N. Pelekis, and C. Doulkeridis, "Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis," in *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pp. 747–754, 2017.

[26] A. Sharma and U. Ghose, "Sentimental analysis of twitter data with respect to general elections in india," *Procedia Computer Science*, vol. 173, pp. 325–334, 2020.

[27] M.-H. Su, C.-H. Wu, K.-Y. Huang, and Q.-B. Hong, "Lstm-based text emotion recognition using semantic and emotional word vectors," in *2018 First Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia)*, pp. 1–6, IEEE, 2018.

[28] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for twitter sentiment analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018.