International Journal of Innovative Research in Engineering and Management (IJIREM)
ISSN (Online): 2350-0557, Volume-11, Issue-3, June 2024
https://doi.org/10.55524/ijirem.2024.11.3.19
Article ID IJIRD-1341, Pages 121-126
www.ijirem.org

An Optimized Approach for Emotion Detection in Real- Time for Twitter Sentiment Analysis with Natural Language Processing

Sangeeta Devi[1], Munish Saran[2], Rajan Kumar Yadav[4] Pranjal Maurya[4], and Upendra Nath Tripathi[5]

[1,2,3,4,5] Department of Computer Science, DDUGU, Gorakhpur, Uttar Pradesh, India

Correspondence should be addressed to Sangeeta Devi; sangeeta2316@gmail.com

Received: 1 June 2024 Revised: 15 June 2024 Accepted: 29 June 2024

ABSTRACT- The automated method of identifying and deciphering the emotions expressed in written text is called sentiment analysis (SA). In the last ten years, SA has become incredibly popular in the NLP (Natural Language Processing) domain. Web-based social networking websites have become a powerful tool for influencing user perceptions and how businesses are marketed. People's opinions are very important when analyzing the effects of information propagation on people's lives in a large-scale network such as Twitter. The polarity and predisposition of a large population towards a particular topic, item, or entity can be determined by data analysis of the tweets. These days, it's easy to see how such analysis is applied in a variety of contexts, including public elections, movie marketing, brand endorsements, and many more. We will build a program that examines the content of tweets on a specific subject in this project. The main goal is to present an approach for polarity score analysis in Twitter streams with noise.

We suggest an emotive categorization of a large number of tweets in this paper. Here, we categorize an expression's sentiments into positive and negative emotions using deep learning approaches. Motivation, fun, happiness, affection, neutral, relief, and surprise are other categories for positive feelings, while anger, boredom, emptiness, hatred, sadness, and worry are categories for negative emotions. We demonstrated how to attain high emotion classification accuracy by experimenting with and evaluating the approach using recurrent neural networks and long-term short-term memory on three distinct datasets. Based on a comprehensive evaluation, the system achieves 88.47% accuracy for positive/negative classifications and 89.3% and 93.3% accuracy for both positive and negative subclasses, respectively, for emotion prediction using the LSTM model.

KEYWORDS- Twitter Sentiments, Emotion Classification, Deep Learning Techniques, Long Short Term Memory, Recurrent Neural Networks

I. INTRODUCTION

On the social networking site Twitter, users can share messages in the format known as "tweets." People can express their thoughts and feelings about a wide range of topics, disciplines, or themes on this platform. It is an assemblage of user opinions and attitudes about a range of subjects, including ordinary online articles and net weblog. When comparing Twitter to older social networking and blogging platforms, there is more relevant data available. The response time on Twitter is significantly faster than on other blogs. Many parties, including consumers and marketers, frequently use sentiment analysis to learn more about products or comprehend market trends[1].

The foundational concepts of sentiment analysis, including sentiment lexicons, feature extraction, and sentiment classification, are covered in this study first. The examines the benefits and drawbacks of Rule-based, Machine Learning, and deep learning approaches for SA. Research also focuses on the challenges of handling sentiment analysis across languages and cultures, as well as the legal and privacy issues that sentiment analysis raises.

Furthermore, this is essential for forecasting the value of the currency or the product grade of a certain company. This is accomplished by examining how the general public feels about the company in relation to time and location[2]. A social science instrument is required to comprehend the significance of public opinions and, consequently, the market worth of the companies. Additionally, businesses are able to gauge how effectively their product is selling.

Additionally, it will be beneficial to examine both favorable and unfavorable product reviews. Twitter makes it easy for enterprises to discover geographical trends by allowing a transfer stream of geo-tagged tweets from specific regions. With the use of this data, the company will be able to analyze various reactions and promote its products more effectively[4].

The prediction of emotions inside a word, sentence, or library of texts is known as sentiment analysis. It is meant to function as a tool for deciphering the viewpoint, attitudes, and feelings conveyed in an online remark. Gaining or gaining access to a description of the broader public opinion on particular themes is the aim. It is a paradigm that specifically divides interactions into labels that are good, negative, or neutral. Social media platforms are widely used by users to network with others and keep up with news and current affairs. These websites (Facebook, Instagram,

Twitter, Google+) give users a forum to express their thoughts. For instance, as soon as they watched a movie, people immediately submit their opinions online and start a discussion about the performing prowess shown in the film. This kind of data serves as a foundation for individuals to assess and appraise the performance of other items as well as movies in order to determine their likelihood of success. These websites have a wealth of material that is useful for social studies and marketing. As a result, sentiment analysis, which encompasses influence analysis, polarity, classification, and emotion mining, has many uses. Twitter is an online social media platform powered by tweets, which are messages with a character restriction of 380. As a result, the character limit makes it mandatory to classify material using hash tags. At this moment, about 8500 tweets are posted every second, or 668.6 million tweets every day. These noisy tweet streams typically represent a variety of topics and shifting opinions in an unorganized and unfiltered manner.

Natural language processing is used in Twitter sentiment analysis to extract, identify, and characterize the sentiment material[4,5].

II. OBJECTIVE

- To implement an algorithm for automatic classification of text into positive or negative sentiment, and to determine the opinion of the masses towards a subject of interest with graphical representation
- To put into practice an algorithm that automatically classifies text as either positive or negative.
- To ascertain whether the general public has positive or negative opinions about the topic at hand.
- Graphical depiction of the analysis using a pie chart or bar graph.

III. NAÏVE BAYES

Naive Bayes categorization: Fortunately, our classes are far simpler to define than Borges', although many language processing jobs include categorization. The naive Bayes algorithm is shown here and is applied to a significant classification task: text categorization. This involves classifying an entire text by giving it a text categorization label selected from a collection of labels. We concentrate on sentiment analysis, often known as the sentiment extraction problem, which involves identifying the writer's positive or negative attitude toward an object[6].

The simplest and fastest classifier is the naive Bayes model. Several researchers assert that this classifier has produced the greatest results for them. If we need to identify the label for a certain tweet, we first determine the probabilities of each label given that feature, and then we choose the label with the highest likelihood. Unigrams has the lowest accuracy rate, with 84.67%. Using greater order n-gram (87.68%) or negativity detection (85.66%) further improves accuracy. We observe that the accuracy is slightly lower when utilizing higher order n-grams alone (88.92%) than when use both Negation detection and high order n-grams. Additionally, we can see that the double step classifier's accuracy is lower than the matching single step classifier's.

IV. NATURAL LANGUAGE PROCESSING

The most straightforward and quick classifier is the naive Bayes model. This classifier is said by several academics to produce the best results. If we want to assign a label to a particular tweet, we first determine the probabilities of each label given that attribute, and then we choose the label with the highest likelihood. With 84.67% accuracy, Unigrams has the lowest accuracy rate. If we further employ higher order n-grams (87.68%) or negative detection (85.66%), the accuracy rises. It can be observed that the accuracy is slightly lower while utilizing both Negation recognition and higher order n-grams (88.92%) as opposed to solely employing the latter. It's also possible to observe that the double step classifier's accuracy is lower than the matching single step's [7,8].

V. SENTIMENT ANALYSIS

The NLP method for obtaining subjective information from textual data is called sentiment analysis. It also involves classifying text into sentences that are good, neutral, or negative. It entails figuring out the underlying attitude, sentiment, idea, and emotion expressed in any natural language. Understanding the underlying sentiment of ideas, attitudes, feelings, and sentiments expressed in any natural language is the goal of sentiment analysis (SA). Numerous textual data sources, such as social media posts, customer reviews, news articles, and product descriptions, can benefit from sentiment analysis.

The overall framework of the SA process is shown in Figure 1, where the text data is input initially, and then the data is pre-processed in three steps: tokenization, stop word filtering, and textual data stemming. The selection of the categorization method is the next and most important step [9].

The goal of SA is to understand a text's overall sentiment as well as the sentiment of specific elements or entities that are mentioned in the text. Although the terms "feeling," "view," "opinion," and "belief" are sometimes used interchangeably, they have different meanings. A view is an individual's own opinion, a belief is an acknowledgement and intellectual consent, and an emotion is a view that conveys one's feelings. Opinions are conclusions that are disputed due to differing expert opinions.

VI. EXPERIMENT

In order to assess the suggested system, the Twitter data set has been gathered and applied to various data sets for both positive and negative emotions. To prepare the data for pre-processing, we normalized the tweets. To prevent undesired symbols and obtain the tweets in a normalized format, pre-processing processes are essential. In order to improve accuracy, the combined RNN and LSTM model is finally applied to classification [10]. The most common procedures used in sentiment classification are depicted in Figure 2

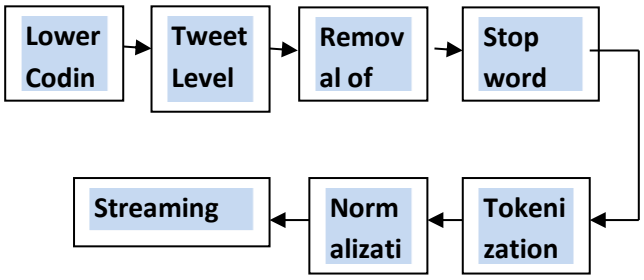

Figure 1: Processing Step

A. Dataset

Three data sets were used for our research and experiments in the study. One lakh tuples made up the original dataset, which was collected from an already-existing dataset. Additionally, data for both positive and negative categorization are included in the dataset. There are 25938 tuples in the second data set, which is for the positive categorization. Seven class labels—enthusiasm, fun, love, happiness, neutral, comfort and surprise—are used to categorize the positive data. There are 27889 tuples in the third and final data set, which is used the negative classification. The six categories into which the negative statistics are divided are: angry, bored, hateful, empty, sad, and worried. Table 1 shows how the sentiment values in the datasets are grouped [11].

This algorithm clarify the steps which are included in the cell state and hidden state on each steps:

Algorithm:

- *xt: Input at time step*
- *ht-1: Hidden state from the previous time step*
- *Ct-1: Cell state from the previous time step*
- *Weight matrices: Wf, Wi, WC, Wo*
- *Bias vectors: bf, bi, bc, bo*

i) Forget Gate Calculation: ft=σ(Wf·[ht−1,xt]+bf)

In an LSTM cell, the forget gate ft determines the amount of the prior cell state (Ct−1) that should be forgotten. Based on the present input xt and the earlier hidden state (ht−1), this choice has been made. The forget gate's output values are regulated by the sigmoid activation function, which guarantees that they fall between 0 and 1. This regulates the information flow inside the cell state. To optimize this gating mechanism, the weights Wf and biased bf are learned along the training phase.

ii) Input Gate Calculation: it=σ(Wi·[ht−1,xt]+bi)

In an LSTM cell, the input gate determines the amount of new data that should be put to the cell state Ct. Based on the present input xt and the earlier hidden state ht−1, this choice has been made. The flow of fresh information into the cell state is regulated by the sigmoid activation function, which makes sure the input gate outputs values that range from zero to one. To optimize this gating mechanism, the weights Wi and biased bi are learned throughout the training process phase.

iii) Candidate Cell State Calculation: Ct~=tanh(WC·[ht−1,xt]+bc)

In an LSTM cell, the additional information that might have been given to the cell state Ct is represented by the candidate cell state C^t. The present input xt and the preceding concealed state ht−1 serve as the foundation for this new data. The potential cell state values are eligible for modifying the cell state since the function that activates tanh makes sure they are between -1 and 1. To optimize this mechanism, the training process yields the weights Wc and biased bc.

iv) Cell State Update: Ct=ft·Ct−1+it·Ct~

An LSTM cell's cell state Ct is updated by merging the newly created candidates cell state C~t with its prior cell state Ct−1. The input gate determines what additional data should be added, while the forget gate determines how much of the previous information should be kept. With this approach, information is updated and selectively retained between time steps, enabling the LSTM to preserve long-term dependencies. To maximize this process, training yields new biases bf, bi, and bc as well as weights Wf, Wf, Wi, and Wc.

v) Output Gate Calculation: ot=σ(Wo·[ht−1,xt]+bo)

How much of the cell state Ct will be used to compute the state that is hidden ht is controlled by the output gate ot in an LSTM cell. By ensuring that the LSTM can control the information flow across the cell state to the hiding state, this gate enables the LSTM to concentrate on the data that is most pertinent to the current time step. To maximize this gating mechanism, training yields the weights Wo and biased bo.

vi) Hidden State Update: ht=ot·tanh(Ct)

An LSTM cell's hidden state (ht) is calculated by modulating the newly generated cell state (tanh) (Ct) using the output gate (ot). With the help of this mechanism, the output gate determines what data from the cell state is most significant to reflect in the concealed state. The LSTM may continuously filter the information that it passes along because of the output gate's ability to regulate over the state that is hidden. This is essential for identifying long-term dependencies and making precise predictions in sequential data jobs.

B. Feature Selection

- **TF-IDF**

In natural language processing, TF-IDF (Term Frequency Inverse Document Frequency) is used to find significant or uncommon words in text data. Term frequency transforms string structured words into mathematical formatted data so that machine learning models can interpret the data. Frequency of Terms The frequency of recurrence of the words we have for the classes is determined using TF. Words are the feature in our data set. every word's frequency in the dataset, which is determined by term frequency [12].

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

where the phrase frequency in document j is represented by i. In memory, a table containing the phrases that appear frequently in the document will be constructed. To determine

which terms are more significant or least common in the document, utilize IDF (Inverse Document Frequency). IDF facilitates our extraction of important terms from the text. The highest degree words are provided by TF, and by calculating the logarithm of the values, IDF assists us in obtaining the lowest occurring words.

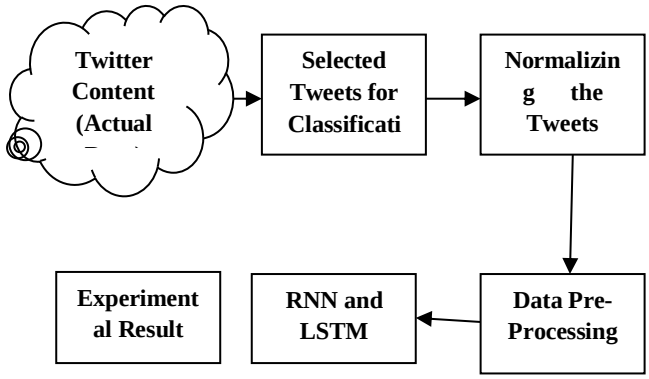

Figure 2: Experimental Setup

Table 1: Description of Datasets

Dataset	No of instance	Positive			Negative			
Dataset1	110000	65467			45633			
Dataset2	12104	Angry	Boredom	Empty	Hate	Sadness		Worry
		655	1768	1856	1078	1607		4565
Dataset3	12265	Enthusiasm	Fun	Happiness	Love	Neutral	Relief	Surprise
		876	1876	6208	3945	986	1656	3211

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

- **Doc2Vec**

Ultimately, we discover that the normalized weights, or TF-IDF output, are the product of the TF and IDF matrices. We obtain the mathematical input data for a machine learning model in this manner. Text represented with a BoW (Bag of Words) is done using TF-IDF. The sentence similarity challenges are performed quite well by the doc2vec vector technique. However, this technique might not be the best option if the input corpus has a large number of words that include misspellings similar to tweets. We employed the Doc2Vec approach, which is a document victimization technique. This is Word2Vec in an enhanced version. This approach was unfavorable in the works because there were numerous instances of word misspellings. It is preferable to turn words into vectors and then utilize those vectors to build the document's overall vector format. Word vectors are used for expressing text in Doc2Vec [13][14].

- **Classification**

With more memory features, the Long Short-Term network is introduced as an extension of RNNs. Repeated cells in LSTMs are connected in a very particular way to prevent vanishing and expanding gradient issues. An enhanced version of a recurrent neural network is called an LSTM. The memory units in an LSTM are blocks that are connected recurrently. The LSTM modifies or adds data to the state of the cell. Gates are the structures that control them. The structures called gates determine whether or not information is passed through them. Thus, by utilizing a wide range of dependencies, LSTM aids in the classification of our tweets [15].

Gates are used to limit the LSTM's flexibility in adding or removing information from the state information tracking cell. Information can flow through the network thanks to gates. The information traveling through is shown by the segment layer's outputs, which vary from 1 to 0. A number of 1 indicates that everything can pass through the gate, whereas a value of 0 indicates that nothing is passing beyond it. First, we need to determine which features should be supplied to the cell state. The LSTM architecture's forget layer makes the choice of whether to accept or forget the features. Two values will be produced by taking into account the values of ht-1 and xt. That is, for every cell state, the output values can be either 0 or 1. A number of 1 indicates that the characteristic is accepted, whereas a value of 0 indicates that the information is avoided. To categorize the emotion in this work, we have to decide whether to accept or reject the characteristic words from the data sets. Depending on the tweets' emotions, each cell state must decide whether to accept or reject the new emotion word.

$$ft = \sigma (Xf .[ht-1, wt] + bf) \quad (1)$$

The identification of new data that must be kept in the cell state is the next step in the categorization process. The sigmoid layer will determine which data has to be updated in conjunction using the input gate layer. The ten layer creates and stores the vector for the recently updated data ct in the

cell. We then modify the cell state after that. In other words, sentimental analysis involves updating the state with newly discovered emotions that were overlooked in earlier stages.

$$it = \sigma (Xi . [ht-1, wt] + bi) \quad (2)$$

$$Ct = \tanh (XC ,[ht – 1, wt] + b_C) \quad (3)$$

Finally, we update the cell state ct-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 it*ct to the previously obtained value. This is the final step for forgetting and updating the state.

$$Ct = ft * Ct-1 + it * Ct \quad (4)$$

The filtered state of cells serves as the basis for the final output. The output that we intend to produce is provided by the sigmoid layer when it is executed. The only way we can output the required portion is to multiply it by the sigmoid layer result after it has passed through the tanh layer. This makes it easier for us to produce the results we want to.

$$ot = \sigma (Xo , [ht-1 , wt] + bo) \quad (5)$$

$$ht = \tanh (Ct) * Ot \quad (6)$$

This is where test cases for both positive and negative emotions begin with an illustration.

- ***Case Positive***

For instance: "whoa! It is regarded as positive. It is unexpected. It is also divided into groups according to the percentage that corresponds to each feeling. Here, the percentages for surprise (46.91%), relief (38.27%), fun (14.64%), neutral (6.22%), and other emotions (less than 5%) indicate that surprise is the most prevalent emotion in this comment. It is therefore classified as unexpected.

- ***Case Negative***

An example of a negative emotion statement would be, "I am so panic these days." It is further divided into groups according to the percentage of each feeling that it corresponds to. Here, concern makes up 78.43 percent of the total, compared to only 14.82% for melancholy, 5.49% for empty, and 1.26% for the other three feelings combined. It falls under the category of worry.

VII. RESULTS AND DISCUSSIONS

For our experimental setup, we made use of the Kaggle dataset. The accuracy with which positive and negative tweets were classified was the evaluation metric employed in our experiment. We compared the different accuracy obtained for other datasets used by Ming-Hsiang Su et al., 2023, and Zhao Jianqiang et al., 2022, using the same models, RNN and LSTM. Accuracy metrics of both LSTM and RNN across various categories are displayed in Table 2. The experiment findings unequivocally suggest that the LSTM model achieve higher accuracy when compared to RNN and CNN classifiers. The twitter dataset proved to be an effective training set for the LSTM model. For positive-negative, negative, and positive classification, respectively, we have three datasets. We used the accuracy of the performance metric to examine the system [15]. The following formula was used to gauge accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (7)$$

There is a potential for incorrect prediction interpretation when calculating accuracy. In other words, the real defective results are recognized as genuine cases and are denoted by the symbol TN, or True Positive. Furthermore, certain right cases are labeled as negative, or false positives, and are denoted by the symbol FP. The accuracy comparison between RNN and LSTM is shown in Figure 3. Our training accuracy for the positive/negative classification in binary terms was 88.47%, while our testing accuracy 83.86%. We achieved a testing accuracy of 89.46% and a training accuracy of 91.13% for the negative classification. We achieved a testing accuracy of 92.75% and a training accuracy of 95.32% for the positive classification. The outcomes clearly demonstrate the superiority of LSTM-based sentiment analysis over RNN [16].

The system's performance is noticeably better since the suggested approach included both the semantic and emotional word vectors. Doc2Vect was utilized by the LSTM-based model learning to extract features from the word sequence. Because of this, it performs better than the RNN-based structure that attempted to model the word sequence's spatial relationship. The experimental results indicate that RNN has a vanishing gradient issue, which prevents the model from learning and adapting even when there are large 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 [16].

VIII. CONCLUSION

In this research, we modeled a Twitter message sentiment analysis system. The tweets we take into account for the study are a combination of various terms and emoticons. We created a model for the classifier. with deep learning methods like LSTM and RNN.

Table 2: Accuracy of different models for Positive/Negative using LSTM, CNN and RNN

Method	Model	Classification	Accuracy (%)
Proposed Method	LSTM	Positive/Negative	88.67
		Positive subclasses	91.13
		Negative subclasses	92.32
	RNN	Positive/Negative	85.21
		Positive subclasses	84.24
		Negative subclasses	89.02
Ming-Hsiang Su et al.2023	LSTM	Positive/Negative	75.66
	CNN	Positive/Negative	68.33
Zhao Jianqiang et al.2022	CNN	Positive/Negative	89.62

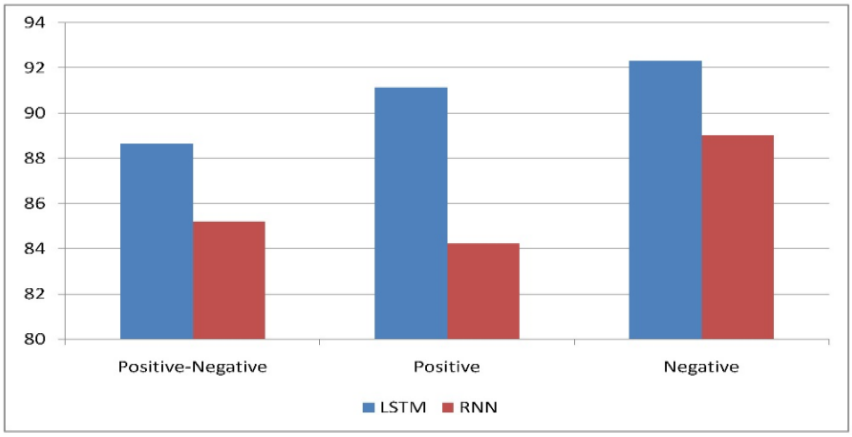

Figure 3: Accuracy obtained for sentiments with LSTM and RNN

We used various feature selection techniques such as TF-ID and Doc2Vect to improve accuracy. The classification model receives a vector as input from the feature extraction process. When it came to the task of classifying the emotional tweets, our model performed better. To make the system more individualized, future research on the analysis of users' personalities from their tweets is required.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

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. Available from: https://doi.org/10.1109/ICICT48043.2020.9112381

[2] D. Ramachandran and R. Parvathi, "Analysis of twitter specific preprocessing technique for tweets," Procedia Computer Science, vol. 165, pp. 245–251, 2019. Available from: https://doi.org/10.1016/j.procs.2020.01.083

[3] A. Sungheetha and R. Sharma, "Transcapsule model for sentiment classification," Journal of Artificial Intelligence, vol. 2, no. 03, pp. 163–169, 2020. Available from: http://dx.doi.org/10.36548/jaicn.2020.3.003

[4] 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. Available from: https://doi.org/10.1142/S0218488598000094

[5] N. F. Alshammari an 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. Available from: https://doi.org/10.1109/CAIS.2019.8769465

[6] 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. Available from: https://doi.org/10.1007/978-3-319-58750-9_64

[7] L. M. Rojas-Barahona, "Deep learning for sentiment analysis," Language and Linguistics Compass, vol. 10, no. 12, pp. 701–719, 2016. Available from: https://doi.org/10.1007/s10462-023-10651-9

[8] 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. Available from: http://dx.doi.org/10.18653/v1/S15-2079

[9] M. Cliche, "Bb twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms," arXiv preprint arXiv:1704.06125, 2017. Available from: https://doi.org/10.18653/v1/S17-2094

[10] 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. Available from: https://doi.org/10.18653/v1/S17-2126

[11] 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. Available from: https://doi.org/10.1016/j.procs.2020.06.038

[12] 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. Available from: http://dx.doi.org/10.1109/ACIIAsia.2018.8470378

[13] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for twitter sentiment analysis," IEEE Access, vol. 6, pp. 23253–23260, 2018. Available from: https://doi.org/10.1109/ACCESS.2017.2776930

[14] 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. Available from: https://doi.org/10.1145/3110025.3116206

[15] 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. Available from: https://doi.org/10.1109/ICICCT.2017.7975191

[16] L. M. Rojas-Barahona, "Deep learning for sentiment analysis," Language and Linguistics Compass, vol. 10, no. 12, pp. 701–719, 2016. https://doi.org/10.1111/lnc3.12228