International Journal for Modern Trends in Science and Technology, 8(06): 503-510, 2022

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0806087
Available online at: http://www.ijmtst.com/vol8issue06.html

Sentiment Analysis of COVID data extracted via Twitter

Rugved Mone | Bhakti Palkar

Department of Computer Engineering, K.J. Somaiya College of Engineering, Ghatkopar East, Mumbai, India.
Email: rugved.mone@somaiya.edu

To Cite this Article
Rugved Mone and Bhakti Palkar. Sentiment Analysis of COVID data extracted via Twitter. International Journal for Modern Trends in Science and Technology 2022, 8(06), pp. 503-510. https://doi.org/10.46501/IJMTST0806087

Article Info
Received: 18 May 2022; Accepted: 17 June 2022; Published: 22 June 2022.

ABSTRACT

Different types of social media sites exist, wherein some of them are LinkedIn, Twitter, Facebook, Instagram, WhatsApp, etc. As the number of social media users increases, the opportunity for the user to express their feelings also increases. Twitter is a choice of many users as it not only allows the users to express their thoughts but to interact with official accounts (PMO, Defense Ministry) which can be seen with a verified tick on the website.

In this thesis titled 'Sentiment Analysis of COVID data extracted via Twitter', multiple machine learning and deep learning techniques have been researched and implemented to perform sentiment analysis. Moreover, a novel approach using deep learning architecture has been proposed. It is based on a combination of Bidirectional Long Short Term (BiLSTM) neural networks and Convolution Neural Networks (CNN). Prior to implementing the algorithms, the data is acquired by using web-scraping techniques and/or public APIs pertaining to Twitter. A comparative analysis of the efficiency and performance of the proposed technique along with other existing approaches discovered during the literature review phase is also presented.

KEYWORDS: ***Sentiment analysis, machine learning, deep learning, Natural Language Processing.***

1. INTRODUCTION

Sentiment analysis classifies text messages with respect to polarity. It labels an examined text message as positive, negative, or even more degrees. Sentiment analysis has been widely applied in social media because social media provides convenient platforms for people to express their opinions, most of which are public messages. Twitter is a unique social media platform. Its microblog service limits the length of each Twitter message(called a tweet) to 280 characters. COVID19 began to spread in December 2019 and now it has affected almost all countries over the globe. The pandemic peaked in May 2020. For the past 20+ months, the number of tweets on COVID-19 are increasing at an unprecedented rate by including positive, negative, and neutral tweets. This diversified nature of tweets has attracted researchers to perform sentiment analysis and analyze the varied emotions of a large public towards COVID-19. Moreover, it is of extreme importance to study and understand the impact which COVID19 has created on the mindset of the world. This analysis will help to predict the emotions of the masses upon the arrival of the next wave of COVID19, or any other epidemic also.

The next section(Section II) consists of a review of existing methodologies considered to tackle this problem statement. These methodologies include both machine learning and deep learning techniques. Section III discusses the proposed deep learning approach and its implementation. The process of implementation

right from data acquisition to final testing and comparison has been detailed. The conclusion related to this issue and the prospect of future work have been presented in Section IV.

2. Related Work

Sentiment analysis is a popular topic; numerous recent works have been looked at before finalizing the proposed approach. This analysis has also been performed on few issues apart from COVID19 also where textual data is applicable. Some other problem statements include analysis of YouTube comments, Facebook posts, etc. About building sentiment analyzers specifically for data related to COVID19, many machine learning as well as deep learning techniques have been reviewed.

Looking at various machine learning approaches for sentiment analysis of COVID data, the use of a plain Naive Bayes classifier is proposed by [1]. The dataset considered by the authors is binary which is restrictive as neutral opinions of people are also frequently found on the internet. The accuracy of the proposed system (84%) is compared with that of Recurrent Neural Networks (77%) in the same publication. The authors of [2] have demonstrated the capabilities of Support Vector Machine (SVM) and K-Nearest Neighbours(KNN) for the same problem statement albeit with a larger dataset than [1]. TFIDF has been employed to generate the initial features from the cleaned data. The accuracy yielded by both these approaches (88% and 78% respectively) is similar to the Naïve Bayes Classifier shown in [1].

Logistic regression as a potential solution has been proposed in [3]. The cleaning techniques are similar to the ones implemented in previously mentioned works. The use of vectorization is considered a preprocessing step. Count vector is also referred to as vocabulary of words which is a common encoding scheme of a given word in a document while TF-IDF is a numerical statistic that shows how important a word is in a document from a collection of corpora. Corpus is a large set of structured text and languages. However, this seems to have its flaws related to the size of the vocabulary. As only vectorization is used, the vector or the sparse matrix consists of all the unique words found across the entire dataset. This in turn exponentially increases the training time and resources required for the training of the model. In [3], the accuracy of the proposed logistic regression approach (83%) is compared with other pre-trained models such as BERT (92%) and VADER (88%) as well. A similar comparison of multiple machine learning techniques in combination with 2 preprocessing techniques is presented by [4]. [5] and [6] show the comparison of ML approaches on textual problem statements but not related to COVID. It is important to understand how the preprocessing and cleaning measures change as a different set of textual data is being considered here.

In addition to machine learning techniques mentioned above, deep learning techniques have also proved to be beneficial when tackling this problem statement. The architectures include making use of Convolution Neural Networks (CNNs), Gated Recurrent Units (GRU), Recurrent Neural Networks (RNNs), Long ShortTerm Networks (LSTMs), and more.

[7] consists of the implementation of a CNN model on a multiclass dataset (Positive, Negative, Neutral). The comparison presents two preprocessing approaches - with and without one-hot encoding for embedding. It is seen that the combination of CNN, one-hot encoding, and vectorization yields a better accuracy of 84.3%. The authors of [8] have employed multi-layered LSTMs and compared the results by using the pre-trained word embedding model of BERT. The dataset size of the Senwave Dataset [13] is relatively small and comprised of 10,000 tweets only. These tweets have more than 10 target classes namely “Optimistic”, “Thankful”, “Empathetic”, “Pessimistic” and more. The two-fold idea of this publication is also to understand the majority of emotions expressed by humans in the era of COVID 19, along with building a realistic sentiment analyzer. The authors of [9] have proposed a similar architecture with changes in hyperparameter tuning but on larger data and a lesser number of target classes. Along with the actual content and tokens present in the tweet, additional information has been derived from the metadata such as demography, timeline, and culture as well. This architecture yields an accuracy of 84%. [10] presents a novel approach by developing an extended version of LSTM named SAB-LSTM. This approach can be considered close to our proposed approach. But it is less complex as only unidirectional LSTM is used. SAB-LSTM makes use of its custom embedding layer. Moreover, a comparative study of SAB-LSTM with the

traditional LSTM has also been shown. All of this has been experimented on a dataset comprising of approximately 80,000 tweets and 5 target classes (Sad, Happy, Neutral, Depressed, Fear) instead of the standard 3.

Apart from these, there are multiple publications referenced. These publications compare the performance of traditional CNNs, and LSTM with the standard machine learning techniques such as Naïve Bayes Classification, Random Forests, and SVMs. In most of these cases, the Deep Learning techniques have shown to be more effective

3. Proposed Methodology

After performing a detailed literature review of various techniques used, we have developed a novel hybrid deep learning model consisting of multiple layers (CNN + BiLSTM) which we will explain in depth later. The overall proposed architecture of our implementation consists of many steps. These steps start with data acquisition, followed by data cleaning, data preprocessing, designing the model, training the model, and concluding with evaluating the model by comparing it with existing architectures. The pictorial representation of said architecture can be seen in Figure 1.

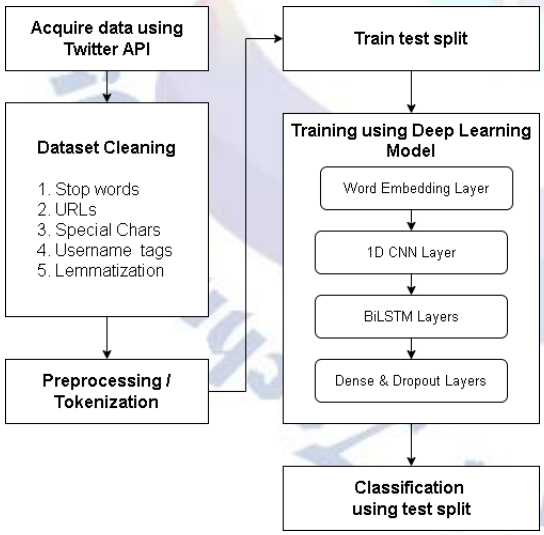

Fig 1. Proposed overall architecture

A. Dataset

The dataset has been compiled by extracting data from two sources. We have performed extraction from Twitter using the open API provided by Twitter. Multiple relevant hashtags have been used to scrape tweets related to COVID19. These tweets have been labeled manually with the sentiments namely 'Positive', 'Negative', and 'Neutral'. Apart from that, a large chunk of data is compiled from a dataset acquired from Kaggle [17]. After the initial cleaning i.e removal of empty/null tweets from the compiled dataset, the distribution of sentiment counts can be seen in Figure 2. The total number of tweets in the dataset is 130,222.

B. Data cleaning

After the dataset was collected from multiple sources, data cleaning was performed. The steps performed for cleaning are a mix of generalized steps as well as Twitter-centric measures. The cleaning process involved the following:

1. Removal of NULL data
2. Removal of stop-words
3. Digits and punctuation characters
4. Discarding URLs
5. Special characters handling
6. Handling tagging of other usernames in tweets.
7. Lemmatization

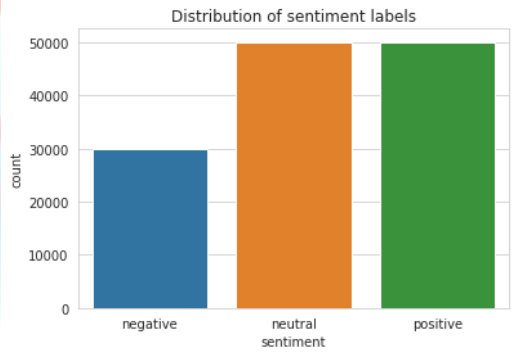

Fig1. Sentiment distributions

C. Data pre-processing

After the initial sanitization has been performed on the dataset, data pre-processing is required. The objective of the data pre-processing stage is to convert the cleaned tweet into a list of tokens, which will be used for training and classification.

Tokenization is best described as the process of creating a vocabulary out of the data which will be used to generate the word embeddings. To carry out this process, the Tokenizer class from the Keras library has been used. In order to tokenize the entire dataset, the total number of unique words/tokens in the dataset should be known, which is obtained as 96000 for our dataset. Out of this, a certain number of words have to be fit into the vocabulary, and the rest are considered out of vocabulary (OOV). This is an experimental

number that can be decided by considering many factors such as the size of the dataset, infrastructure, and complexity of one's deep learning model. We have experimented by considering 60,000 words into our vocabulary. This is achieved by using the 'fit_on_texts()' method. After the vocabulary has been generated, the individual tweets have to be mapped to the vocabulary with respect to the words present in them.

The tokenized data for every tweet should now be padded to create a consistent shape of data that can be passed on to the model. The length of the padded text is experimental and should be decided by looking at the distribution of lengths of tweets across the dataset. The distribution for the dataset in question is shown in the Figure 3. The average word count of the tweets is 14 words. Hence the padding limit has been selected as 20 after multiple trials of values (15, 20, and 25).

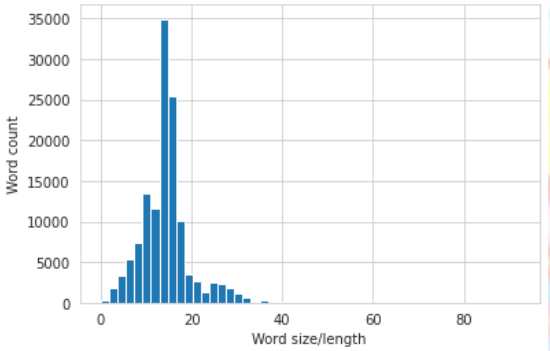

Fig 2. Word length distribution

D.Proposed deep learning architecture

The layers of the proposed deep learning model can be seen in Figure 4.

Input layer: As seen in the Figure 4, the first layer of the neural network is the input layer, which is a standard layer. After the tokenization phase is complete, the dataset is of the shape [130,000 x 20], as we have considered 20 to be the padding limit. Here 130,000 is the number of tweets considered for training. It is 80% of the total number of cleaned tweets. Multiple ratios for the training testing split have been experimented with and will be presented in a latter section.

Embedding layer: The embedding layer is the first hidden layer in the architecture, wherein it aims to generate an embedding (i.e the output vector) for each word in the dataset. The 3 required parameters are (sequentially) vocabulary size (60000), output vector size, and the input length size (20). The output vector size determines the length of the vector/number of features for each word. This number is experimental and depends upon the type of problem being tackled. We have considered 15 and 25 for comparative analysis. Thus the output vector/embedding for each tweet is generated.

CNN layer: The embeddings are then fed to a 1-dimensional convolution network layer (1D CNN). The main idea of introducing a 1D CNN layer is to consider the grouping of sequential word embeddings to generate features. This enables the model to learn information that is best acquired and understood when multiple words in a tweet are considered simultaneously. The kernel size used in the 1D CNN determines the number of word embeddings that will be grouped. In this study, we have experimented with 4, and 5. Hence, a greater number of features depending on the filter size (64 here) are generated by clubbing 3-word embeddings at a time over the sliding window.

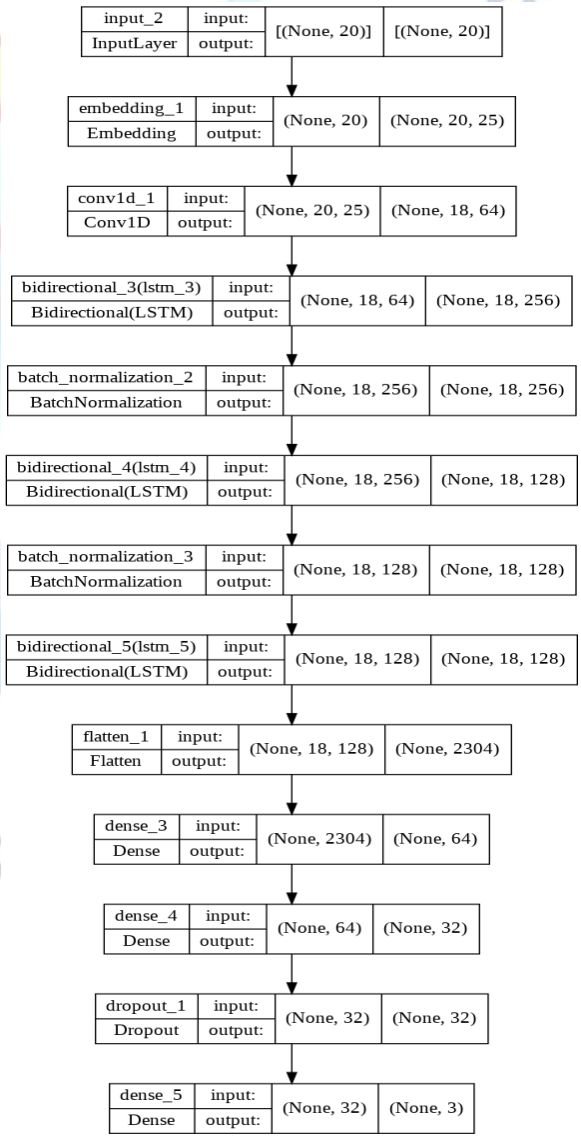

Fig 3. Proposed model

BiLSTM layer: Bidirectional long-short term memory (BiLSTM) layer processes sequenctial information in both directions. With the regular LSTM, the input flow is only in one direction, either backward or forward. However, in bi-directional, the input flow can be facilitated in both directions to preserve the future and the past information.

These features generated by 1D CNN layer are then supplied to a series of BiLSTM layers provided by the Keras library. The output dimensions are lowered as the layers progress. Using return_sequences=True ensures that the output vector for BiLSTM is returned after each word/embedding in the tweet is processed. If return_sequences is not set to True, only one output vector is returned after all the words/embeddings in the tweet are processed.

Batch Normalization layers: The key issue that batch normalization tackles is the internal covariate shift. Internal covariate shift occurs due to the inherent nature of neural networks. At every epoch of training, weights are updated and different data is processed, which means that the inputs to a neuron is slightly different every time. As these changes get passed on to the next neuron, it creates a situation where the input distribution of every neuron is different at every epoch. Another issue that batch normalization tackles is vanishing or exploding gradients. Gradients are the factors due to which the weights are updated at every hidden layer of the neural network. These small gradients get even smaller when multiplied together deeper into the network. When using backpropagation, the gradient gets exponentially closer to 0. This "vanishing" gradient severely limits the depth of networks.

Batch normalization normalizes a layer input by subtracting the mini-batch mean and dividing it by the mini-batch standard deviation. Mini-batch refers to one batch of data supplied for any given epoch, a subset of the whole training data. The normalization ensures that the inputs have a mean of 0 and a standard deviation of 1, meaning that the input distribution to every neuron will be the same, thereby fixing the problem of internal covariate shift and providing regularisation.

Dense and Dropout layers: To move towards convergence, Flatten layer and a series of Dense layers are used. A Dropout layer with a 25% drop rate has been added to tackle overfitting. In theory, one Dense layer with the output dimensions equal to the number of target classes should be enough. However, in practice, it has been observed that adding multiple dense layers with gradually decreasing output space is beneficial. Since the target classes are more than 2 (3), the 'categorical_crossentropy' loss has been used in the final Dense layer.

4. Training, Classification, And Results

Various training and testing split ratios ranging from 65:35 to 80:20 have been experimented with. Also, by analysing the dataset, we have experimented with multiple epochs of training (10, 15, and 20), and the output size of the embedding layer (15,20, and 25) as well. Maximum accuracy is achieved for the 80:20 training split and 20 epochs, along with the output size of the embeddings being 25. The graph depicting the training accuracy and validation accuracy is shown in Figure 5.

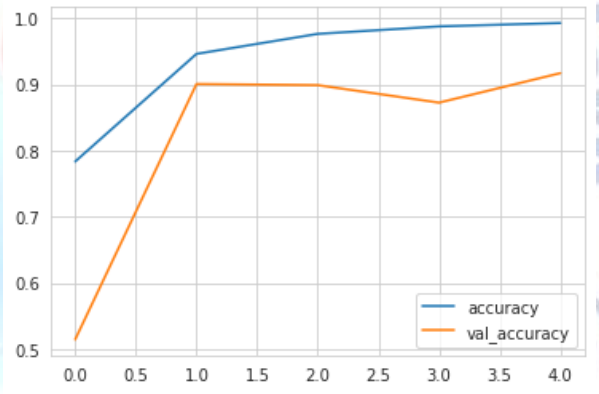

Fig 4. Training and validation accuracy

Using the optimum combination, the proposed model yields a training accuracy of 98.81% and a validation accuracy of 94.43%. To analyse the performance of the model on every class, the confusion matrix (precision, recall, f1 score) and a heatmap have been plotted.

Fig 5. Proposed approach heatmap

Table I: Confusion matrix (proposed approach)

	precision	*recall*	*F1-score*
Neutral	0.95	0.87	0.91
Negative	0.94	0.89	0.91
Positive	0.92	0.95	0.89

Thus it can be seen that the proposed approach performs well in the classification of each target class. Accuracy being the most intuitive performance measure is simply the ratio of the correct predictions to the total number of predictions. Precision is the ratio of correct positive predictions count for any class to the total positive predictions for the same class. Recall is the ratio of correct positive predictions count for any class to all predictions in the same class. F1 score is the weighted average of precision and recall. It is most beneficial to look at when there is an uneven distribution of target classes across the dataset. For our dataset, the tweet count for every target class is comparable to other classes. This leveled distribution of labeled data also plays a role in training the model for predicting without bias.

5. Model Comparison

To gauge the effectiveness of the proposed deep learning model, we have implemented the architectures of multiple traditional approaches presented in the literature survey. These traditional models (machine learning as well as deep learning) have been trained and tested on the same dataset as that of the proposed model. The models implemented for traditional approaches have also been trained by testing with multiple values of hyperparameters considered during the designing of the proposed model. A comparative study of performance is shown in Table II.

Table II shows the optimum results for each approach along with the values of the terms in the confusion matrix. However, while implementing and testing each architecture (proposed as well as traditional), we have performed extensive tuning of the hyper-parameters of each model to understand their impact. In Table II it can be observed that

Table II: Comparison with traditional models

Method	Validation accuracy	Precision	Recall	F1-Score
CNN+ Multilayer BiLSTM (Proposed approach)	**0.944**	**0.936**	**0.903**	**0.919**
Bidirectional GRUs	0.891	0.874	0.887	0.880
BiLSTM	0.912	0.900	0.916	0.907
LSTM	0.918	0.906	0.893	0.899
CNN	0.914	0.896	0.896	0.896
Naïve Bayes	0.840	0.806	0.756	0.780
KNN	0.827	0.825	0.827	0.826

the proposed hybrid architecture (CNN + BiLSTM) performs comparably as compared to the existing machine learning as well as deep learning approaches. For the machine learning approaches mentioned in Table II, the techniques term frequency-inverse document frequency (TFIDF) and vectorization have been implemented as pre-processing steps. This was done by referring to methodologies discussed in [1] and [2] to accurately emulate the architectures proposed by them and then train those on our dataset. Tables III, IV, V, and VI demonstrate the effect of various hyper-parameters on the proposed approach as well as traditional approaches.

Table III. Training of **proposed model (CNN + BILSTM)** with various hyper-parameters (20 epochs)

Embedding vector size	Training split (%)	Training accuracy	Validation accuracy
15	70	0.953	0.919
15	75	0.967	0.913
15	80	0.966	0.927
25	70	0.973	0.902
25	75	0.976	0.927
25	80	0.988	0.943

Table IV. Training of **traditional BILSTM** with various hyper-parameters (20 epochs)

Embedding vector size	Training split (%)	Training accuracy	Validation accuracy
15	70	0.948	0.873
15	75	0.957	0.881
15	80	0.961	0.912
25	70	0.955	0.89
25	75	0.972	0.902
25	80	0.978	0.894

Table V. Training of **traditional LSTM** with various hyper-parameters (20 epochs)

Embedding vector size	Training split (%)	Training accuracy	Validation accuracy
15	70	0.941	0.881
15	75	0.949	0.884
15	80	0.966	0.905
25	70	0.953	0.908
25	75	0.958	0.911
25	80	0.972	0.918

Table VI. Training of **traditional CNN** with various hyper-parameters (20 epochs)

Embedding vector size	Training split (%)	Training accuracy	Validation accuracy
15	70	0.941	0.861
15	75	0.961	0.882
15	80	0.966	0.897
25	70	0.971	0.893
25	75	0.976	0.914
25	80	0.989	0.901

6. Future ScopeAndConclusion

As a part of this research, we have proposed a novel approach to perform sentiment analysis on extracted tweets related to COVID19. This novel deep learning approach is based on a hybrid model consisting of CNN, BiLSTM, and Batch Normalization layers. As is seen in the previous two sections, the performance of our model is comparable with, and in most cases better than the traditional deep learning and machine learning architectures. Having a validation accuracy of 94.42%, we have compared the performance of existing architectures by implementing, training, and testing those on the same dataset. Hence an extensive homogenous comparison has been exhibited. Moreover, the model has been evaluated by considering multiple reliable parameters namely precision, recall, and F1-Score.

The choice of pre-processing techniques is also relevant in combination with the architecture considered. For instance, the Naïve Bayes classifier yields optimum results when combined with TF-IDF and vectorization. However, In case of the proposed deep learning approach, the preprocessing techniques such as TF-IDF/Count Vectorizer are replaced by using self-generated word embeddings by adding an embedding layer immediately after the input layer of the neural network. By experimenting with more layers and fine-tuning of hyperparameters, it may be possible to improve the performance of the proposed deep learning approach.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

References

[1] A. Radaideh, F. Dweiri and M. Obaidat, "A Novel Approach to Predict the Real Time Sentimental Analysis by Naive Bayes & RNN Algorithm during the COVID Pandemic in UAE" 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics.

[2] H. Adamu, M. J. Bin Mat Jiran, K. H. Gan and N. -H. Samsudin, "Text Analytics on Twitter Text-based Public Sentiment for Covid-19 Vaccine: A Machine Learning Approach," 2021 IEEE IICAIET

[3] A. J. Nair, V. G and A. Vinayak, "Comparative study of Twitter Sentiment On COVID - 19 Tweets" 2021 5th International Conference on Computing Methodologies and Communication

[4] G. M. Raza, Z. S. Butt, S. Latif and A. Wahid, "Sentiment Analysis on COVID Tweets: An Experimental Analysis on the Impact of Count Vectorizer and TF-IDF on Sentiment Predictions using Deep Learning Models," 2021 International Conference on Digital Futures and Transformative Technologies

[5] M. Wongkar and A. Angdresey, "Sentiment Analysis Using Naive Bayes Algorithm Of The Data Crawler: Twitter" 2019 Fourth International Conference on Informatics and Computing (ICIC), 2019

[6] N. Muhammad, S. Bukhori and P. Pandunata, "Sentiment Analysis of Positive and Negative of YouTube Comments Using Naïve Bayes – Support Vector Machine (NBSVM) Classifier," 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering

[7] E. Nugraheni, P. H. Khotimah, A. Arisal, A. F. Rozie, D. Riswantini and A. Purwarianti, "Classifying aggravation status of COVID-19 event from short-text using CNN," 2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), 2020, pp. 240-245, doi: 10.1109/ICRAMET51080.2020.9298674.

[8] Chandra R, Krishna A (2021) COVID-19 sentiment analysis via deep learning during the rise of novel cases. PLoS ONE 16(8)

[9] Singh C, Imam T, Wibowo S, Grandhi S. A Deep Learning Approach for Sentiment Analysis of COVID-19 Reviews. Applied Sciences. 2022

[10] D. A. Kumar and A. Chinnalagu, "Sentiment and Emotion in Social Media COVID-19 Conversations: SAB-LSTM Approach," 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART), 2020, pp. 463-467, doi: 10.1109/SMART50582.2020.9337098.

[11] T. Rahman and S. Aktar, "A Machine Learning Approach to Track COVID-19 Pandemic using Sentiment Analysis," 2021 3rd International Conference on Electrical & Electronic Engineering (ICEEE), 2021, pp. 145-148, doi: 10.1109/ICEEE54059.2021.9718770.

[12] O. Baker, J. Liu, M. Gosai and S. Sitoula, "Twitter Sentiment Analysis using Machine Learning Algorithms for COVID-19 Outbreak in New Zealand," 2021 IEEE 11th International Conference on System Engineering and Technology (ICSET), 2021, pp. 286-291, doi: 10.1109/ICSET53708.2021.9612431.

[13] Yang Q, Alamro H, Albaradei S, Salhi A, Lv X, Ma C, et al. SenWave: Monitoring the Global Sentiments

[14] N. S. Devi and K. Sharmila, "Fine Grainded Sentiment Analysis on COVID-19 Vaccine," 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), 2021, pp. 707-711, doi: 10.1109/SMART52563.2021.9676205.

[15] G. A. Sandag, A. M. Manueke and M. Walean, "Sentiment Analysis of COVID-19 Vaccine Tweets in Indonesia Using Recurrent Neural Network (RNN) Approach," 2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS), 2021, pp. 1-7, doi: 10.1109/ICORIS52787.2021.9649648.

[16] P. Vyas, M. Reisslein, B. P. Rimal, G. Vyas, G. P. Basyal and P. Muzumdar, "Automated Classification of Societal Sentiments on Twitter with Machine Learning," in IEEE Transactions on Technology and Society, doi: 10.1109/TTS.2021.3108963.

[17] https://www.kaggle.com/datasets/abhaydhiman/covid19-sentiments

[18] https://www.kaggle.com/datasets/datasciencetool/covid19-vaccine-tweets-with-sentiment-annotation