Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

BERT for Twitter Sentiment Analysis: Achieving High Accuracy and Balanced Performance

Oladri Renuka[1], Niranchana Radhakrishnan[2]

[1]Data Science and Artificial Intelligence, School of Technology, Woxsen University, Hyderabad, India

[2]Department of Computer Science and Engineering, Alliance College of Engineering and Design, Alliance University, Bengaluru, India

Email: [1]renuka.oladri_2025@woxsen.edu.in, [2]niranchana.r@alliance.edu.in

Abstract

The Bidirectional Encoder Representations from Transformers (BERT) model is used in this work to analyse sentiment on Twitter data. A Kaggle dataset of manually annotated and anonymized COVID-19-related tweets was used to refine the model. Location, tweet date, original tweet content, and sentiment labels are all included in the dataset. When compared to the Multinomial Naive Bayes (MNB) baseline, BERT's performance was assessed, and it achieved an overall accuracy of 87% on the test set. The results indicated that for negative feelings, the accuracy was 0.93, the recall was 0.84, and the F1-score was 0.88; for neutral sentiments, the precision was 0.86, the recall was 0.78, and the F1-score was 0.82; and for positive sentiments, the precision was 0.82, the recall was 0.94, and the F1-score was 0.88. The model's proficiency with the linguistic nuances of Twitter, including slang and sarcasm, was demonstrated. This study also identifies the flaws of BERT and makes recommendations for future research paths, such as the integration of external knowledge and alternative designs.

Keywords: BERT, Natural Language Processing, Machine Learning, Sentiment analysis, Multinomial Naive Bayes.

DOI: https://doi.org/10.36548/jtcsst.2024.1.003
Received: 03.02.2024, received in revised form: 24.02.2024, accepted: 08.03.2024, published: 21.03.2024

1. Introduction

Twitter is a well-known website where users may share their daily activities and opinions on significant issues, as well as express themselves through messages. Consequently, individuals employ emojis, slang, acronyms, and condensed forms to make their points more concise. In addition, people use polysemy and sarcasm to convey their ideas [1][2].

In the field of sentiment analysis, many machine learning algorithms have replaced lexicon-based techniques, but even these traditional methods fail to capture the slang and sarcasm in the text [3][4]. To bridge this gap our study uses the large language models specifically BERT, to understand the text better [5].

The scarcity of research employing BERT for Twitter data analysis is notable, by using BERT to perform sentiment analysis on Twitter data and comparing it's results with common baseline Multinomial Naïve Bayes (MNB), our study closes the gap [6]. Our results state that BERT performs well with informal data, identifies slang and sarcasm properly, and classifies tweets accurately.

The document's remaining sections are organised as follows: In Section 2, pertinent research on BERT and sentiment analysis is discussed. Section 3 outlines the recommended approach and its components. Section 4 presents the results and comments. Section 5 concludes the paper.

2. Related Work

We review the relevant literature on sentiment analysis and BERT in this section and divide it into four subsections: 2.1 approaches that rely on rules and lexicons; 2.2 machine learning techniques; 2.3 deep learning techniques; and 2.4 BERT-based techniques.

2.1 Approaches that Rely on Rules and Lexicons

The earliest and most basic techniques for sentiment analysis are rule-based and lexicon-based approaches, which use dictionaries and predefined rules to identify sentiment words and expressions. These methods have low coverage, scalability, and language and domain adaptability, but they do not require labelled data or a training process. A lexicon of words annotated with their semantic orientation (polarity and strength) and several linguistic

rules to handle negation, intensification, and modulation are used in Taboada, M., et al. [7] proposed method, the Semantic Orientation Calculator (SO-CAL). Using a Twitter dataset, they implemented their method and obtained a 65.5% accuracy rate. Thelwall, M., et al. [8] proposed a technique called SentiStrength that handles punctuation, emoticons, spelling correction, and booster words using a lexicon of words with positive and negative scores along with many heuristics. They used a Twitter dataset to test their method, and their accuracy was 60.6%. A technique known as NRC-Canada was proposed by Mohammad, S.M., et al [9]. It makes use of a lexicon of words and phrases along with their sentiment and emotion scores, as well as some rules to deal with hashtags, negation, and modifiers. Using a Twitter dataset, they implemented their method and obtained 69.2% accuracy.

2.2 Machine Learning Techniques

The most widely used and well-liked techniques for sentiment analysis are machine learning approaches, which automatically extract features and patterns for sentiment analysis from labelled data. These methods can achieve higher accuracy and adaptability than rule-based and lexicon-based methods, but they require large and high-quality datasets as well as computational resources [10]. To evaluate the new coronavirus explosion, Dubey et al. [11] gathered tweets from four nations in Europe over 20 days in March. To analyse the coronavirus data that was gathered, Medford et al [12]. used unsupervised machine-learning approaches. In a different study, Alhajji et al. [13] analysed and classified the data they gathered from Twitter using the Naive Bayes classifier.

2.3 Deep Learning Techniques

The most sophisticated and cutting-edge techniques for sentiment analysis are deep learning approaches, which employ neural networks to acquire sophisticated and high-level features and representations for sentiment analysis. These methods can outperform machine learning methods in terms of accuracy and robustness, but they also require more data and processing power, and they may have problems with overfitting and interpretability. A CNN is used by Severyn, A., and Moschitti, A. to learn local and global features from word embeddings, and a SoftMax layer is used to classify the sentiment polarity. Additionally, they initialised the word embeddings using word2vec, a pre-trained word embedding model. 88.3% accuracy was attained using CNN and word2vec features [14]. Preethi et al. [15] applied deep

learning to sentiment analysis for a cloud-based recommender system using the food dataset from Amazon. Another method uses an LSTM model based on emotion detection to integrate sentiment and semantic information. [16].

2.4 BERT-based Techniques

Since pre-trained language models can use vast amounts of unlabelled data to produce broad language representations, they have become more important in a range of NLP applications; among of the greatest examples are Elmo [17], GPT [18], and BERT [19]. Because of its unrivalled bidirectionality and attention mechanism, the BERT model is the one that gets the most attention among all [20]. Researchers are monitoring its impact on subsequent NLP tasks as a result.

3. Proposed Work

This section provides examples of the suggested method for categorizing tweets and conducting sentiment analysis. Figure 1 shows the flow of proposed work.

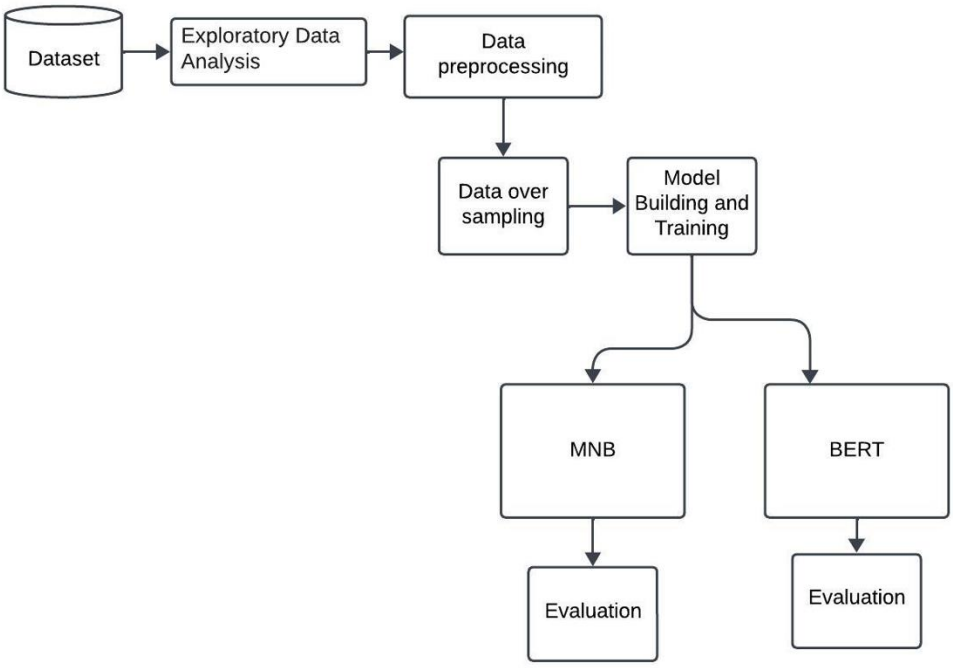

Figure 1. Flowchart of Proposed Work

3.1 Data Collection

For our study, we utilized the “COVID-19 NLP Text Classification” dataset available on Kaggle. This dataset was specifically compiled to facilitate research on sentiment analysis related to the COVID-19 pandemic. It encompasses a diverse collection of tweets, manually tagged to ensure the accuracy of sentiment classification. To maintain user privacy, all names and usernames within the dataset have been coded.

The dataset is structured into six columns of attributes including the username, screen name, location, tweet date and time, actual textual content of the tweet, and the sentiment label assigned to the tweet, and around 45000 tweets. This dataset serves as a valuable resource for our sentiment analysis, offering a snapshot of global sentiment during a significant period of recent history. Its rich content allows for an in-depth exploration of public opinion, captured through the lens of social media communication during the pandemic.

3.2 Data Preprocessing

For data preprocessing we followed following steps. First, we assigned numerical values to the sentiment labels: 0 represented negative and extremely negative, 1 represented neutral, and 2 represented positive and extremely positive. Subsequently, we eliminated extra spaces, URLs, special characters, emojis, and hashtags from the tweet text. Third, we used the BERT tokenizer to tokenize the text of the tweet. This involves breaking the text up into smaller words and appending special tokens like [CLS] and [SEP]. Fourth, we discovered that the majority of tweets have fewer than 80 tokens after examining the distribution of tweet lengths as we can see in Figure 2. As a result, we padded or truncated the tweets by the 80-character maximum length for tokenization.

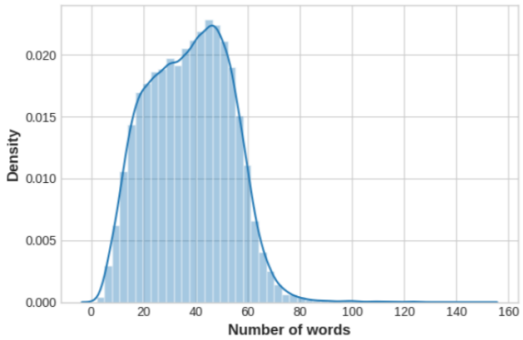

Figure 2. Number of Words in a Sentence

Fifth, we plotted the daily and location-specific tweet counts after converting the tweet date column to Date Time format.

3.3 Data Oversampling

The majority of tweets are classified as neutral, followed by positive, negative, and extremely positive or negative, which leads us to conclude that the dataset is unbalanced. We used the RandomOverSampler method from the imbalanced-learn library to oversample the training data to eliminate the bias brought on by the class imbalance. This method duplicates the minority class samples at random until the number of samples in each class is the same [21].

3.4 Data Splitting and One-Hot Encoding

Using 80% of the oversampled data for training and 20% for validation, we divided the data into training and validation sets. Additionally, we prepared the testing data by rearranging the data and retaining only the pertinent columns (sentiment and tweet text). We used the TensorFlow library's categorical function to perform one-hot encoding on the target labels. The numerical categories are transformed into binary vectors of length four by this function, where a sentiment class is represented by each position.

3.5 Model Building – Multinomial Naïve Bayes

We utilised MNB, a baseline model that is frequently used for text classification tasks [22]. The Multinomial Naive Bayes (MNB) model is a probabilistic classifier based on Bayes' theorem, and it operates under the assumption of feature independence. MNB models the word counts and assumes the likelihood of observing a word count is given by a multinomial distribution. The parameters of the model are calculated using the frequency of the words in the documents. The architecture of the MNB model in Figure 3 can be described as follows:

The input text data is pre-processed using techniques like tokenization, stop-word removal, and lemmatization.

For feature extraction, we utilized CountVectorizer to convert the tweet text into a matrix of token counts, followed by Tf-idfTransformer to reflect the importance of words

within the documents. The term frequency-inverse document frequency (Tf-idf) is calculated using the following equations:

tf(w, d) = frequency of word w in document d (1)

Tf-idf (w, d)= tf (w, d)*idf(w) (2)

where the inverse document frequency, or idf(w), indicates how frequently a word appears in all of the corpus's documents. It is computed as:

$$\text{idf(w)} = \log(\frac{N}{df(W)}) \quad (3)$$

where df(w) is the number of documents that include it) and N is the total number of documents in the corpus.

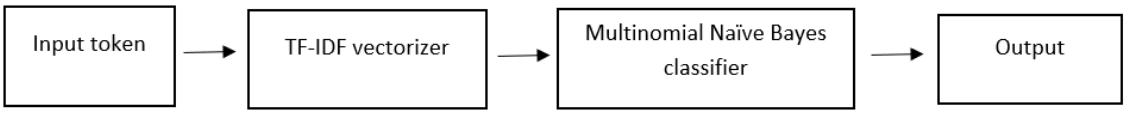

Figure 3. Multinomial Naïve Bayes Architecture

Training data was used to create feature vectors for a Multinomial Naive Bayes classifier. Using the assumption of conditional independence between features (words), MNB applies the Bayes theorem to determine the likelihood that, given the observed features (words), a document d will belong to a class c (negative, neutral, or positive). The formula used to get the posterior probability P(c|d) is:

$$\text{P(c|d)} = \text{P(c)} * \frac{\left(\text{P(d|c)}\right)}{P(d)} \quad (4)$$

where P(c) is the class c prior probability, which is calculated using the training data's class frequencies.

The probability of document d given class c is P(d|c), which is computed as the product of the probabilities of each individual word assuming independence:

$$P(d|c) = \pi\ (P(w_i|\ c))\ \text{ for all words } w_i \text{ in document d} \quad (5)$$

Since our goal is to maximise the posterior probability, P(d), the total probability of the document, is a constant value for all classes and can be disregarded during classification.

A document d is assigned by the MNB model to the class c that has the largest posterior probability P(c|d).

Using the testing data, the trained MNB model was assessed. The model's performance was assessed by computing accuracy, precision, recall, and F1-score for each sentiment class (positive, neutral, and negative), as well as overall.

3.6 Model Building – BERT

We investigated a deep learning strategy utilising BERT, a potent pre-trained transformer model for natural language processing tasks, in light of MNB's limitations [23]. BERT uses a sophisticated neural network architecture to capture the contextual relationships between words in a sentence, in contrast to MNB, which depends on statistical features. The architecture of BERT in Figure 4 is described as follows:

Two input layers, each accepting a sequence length of 150 tokens. A pre-trained BERT model (bert-base-cased) with an output shape of (None, 150, 768) for the last hidden state and (None, 768) for the pooled output. A dense layer with three output units (corresponding to the sentiment classes) and a SoftMax activation function. It transforms the output vector into sentiment class probabilities.

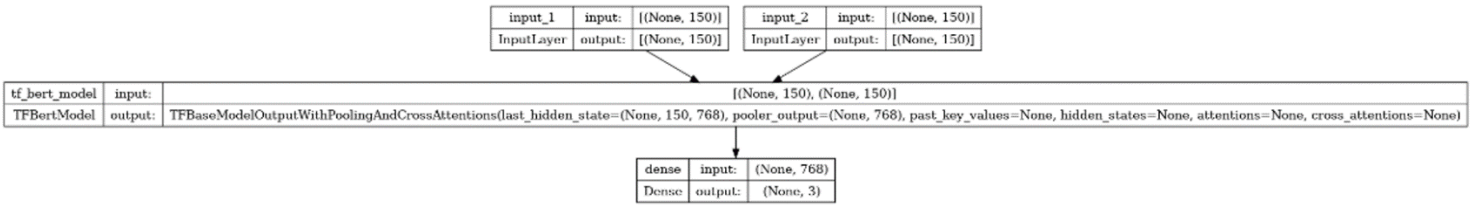

Figure 4. BERT model architecture

The BERT model was fine-tuned on our dataset, starting with the pre-trained weights and adjusting them to the specific task of sentiment classification. The fine-tuning process involved optimizing the categorical cross-entropy loss function:

$$L_{CE} = -\Sigma(\mathrm{y} * \log(y_{hat})) \quad (6)$$

Testing data are used to predict sentiment labels using the trained BERT model. The predicted sentiment labels are then converted into a one-hot encoded format for evaluation.

4. Results and Discussion

In this section, we discuss the results of the proposed methodology. For developing models used in this study, we used Python's scikit-learn package for MNB and the Hugging face transformers module for the BERT model. The previously mentioned methodology's outcomes are shown in this section. We evaluated the models' performance using traditional evaluation metrics like accuracy, precision, recall, F-1 score, and confusion matrix [24].

The confusion matrix summarizes the performance of a model on unseen data, where in the rows represent actual classes and columns represent predicted classes.

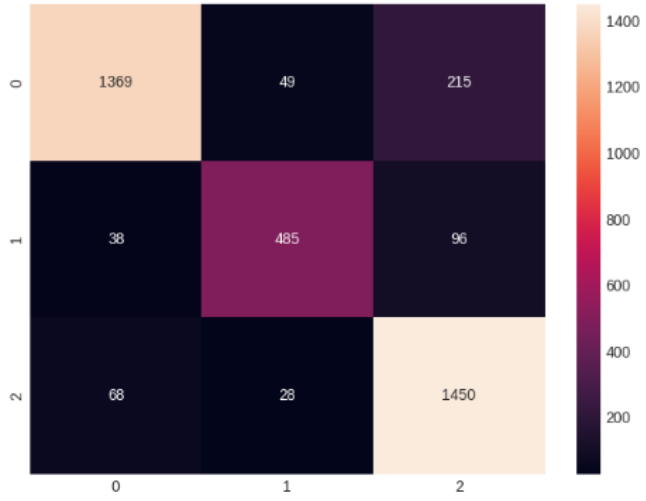

Figure 5. Confusion Matrix of the Proposed BERT Model

The diagonal values of Figure 5 confusion matrix are high hence we can say that model is accurately predicting all the three classes (positive, negative, and neutral).

Accuracy defines the percentage of correct predictions by model It is calculated as:

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

Precision defines the model's capability to predict the target class:

$$\text{Precision of class c} = \frac{TP_c}{TP_c+FP_c} \quad (8)$$

where True Positives TP_c and False Positives FP_c for class c are represented, respectively.

Recall reveals if an ML model can locate every item in the target class:

$$\text{Recall (class c)} = \frac{TP_c}{TP_c+FN_c} \quad (9)$$

Where True Positives TP_c and False Negatives FN_c for class c are represented respectively.

The F1-score is a metric that combines the contributions of precision and recall into a harmonic mean:

$$\text{F1-score (class c)} = 2 * \frac{Pecision_c * Recall_c}{(Precision_c + Recall_c)} \quad (10)$$

These metrics shed light on various facets of the model's functionality. While precision and recall provide class-specific information, accuracy provides an overview of the model's overall performance. The precision versus recall trade-off is balanced by the F1-score.

Table 1. The Evaluation Results for both MNB and BERT

Metric	**Multinomial Naïve Bayes**	**BERT model**
Accuracy	0.707	0.870
Precision (Negative)	0.66	0.93
Precision (Neutral)	0.72	0.86
Precision (Positive)	0.78	0.82
Recall (Negative)	0.42	0.84
Recall (Neutral)	0.78	0.78
Recall (Positive)	0.42	0.94
F1-score (Negative)	0.75	0.88
F1-score (Neutral)	0.74	0.82

F1-score (Positive)	0.51	0.88

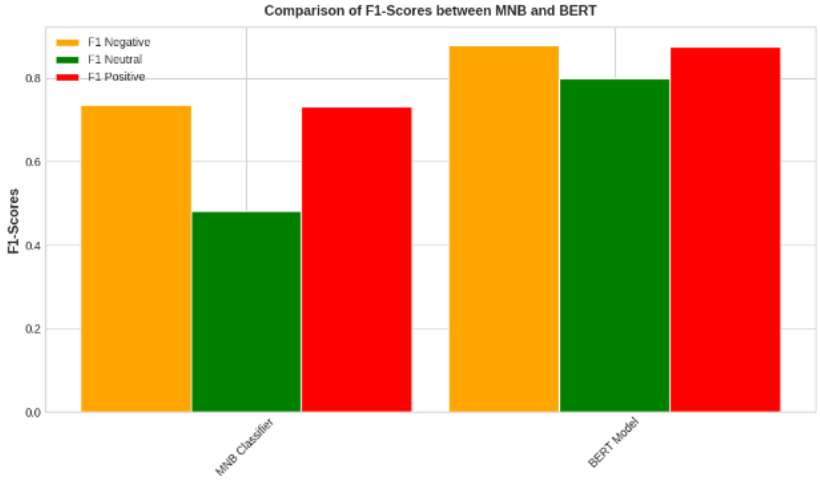

Figure 6. F1 Scores Comparison of MNB and BERT

From Table 1 the evaluation results unequivocally show that the MNB and BERT models perform significantly differently. Despite having an overall accuracy of 0.707, MNB showed imbalances between classes. Precision: 0.70, recall: 0.78, F1-score: 0.74), while it was less successful with neutral tweets (precision: 0.66, recall: 0.42, F1-score: 0.51). This suggests that MNB finds it challenging to convey the subtleties of neutral language. At 0.870, the BERT model demonstrated superior overall accuracy. Additionally, it demonstrated notable gains in performance across all sentiment classes, with precision, recall, and F1-score above 0.80. Even though MNB was effective in handling large datasets with discrete features, simple, and faster in training and prediction, it struggles with understanding context and long-range dependencies. Particularly when neutral sentiment is involved, the assumption of feature independence frequently proves to be false, which results in classification mistakes. As we can see in Figure 6, the BERT model overcomes this weakness of MNB but it requires more training time and computational resources to fine tune on a dataset.

5. Conclusion

In conclusion, this study compared BERT and MNB's performance in sentiment analysis task. The results state that MNB has struggled with predicting neutral class but BERT

performs well in all three classes (positive, negative, and neutral). More accurate sentiment classification resulted from its capacity to identify long-range relationships and contextual links within text, especially for informal language and sarcasm, both of which are prevalent in tweets. These results show that deep learning approaches, such as BERT, outperform more basic models, such as MNB, for sentiment analysis tasks. A future study may look at the integration of domain-specific expertise or other deep learning frameworks to improve sentiment analysis precision.

Ethical Statement

The ethical guidelines for handling internet data were followed in this study. The Twitter data we used was taken from a publicly accessible dataset that complied with the rules set forth by Twitter. No personal user data was gathered. The gathered information was properly anonymized and used only for sentiment analysis research, with no sharing with outside parties.

Acknowledgement

We are appreciative of the people who made the Twitter dataset accessible to the public so that this study could be conducted. We thank the creators of the open-source libraries that were utilised in this research, including scikit-learn, Transformers, and TensorFlow, for their significant contributions to the natural language processing community.

References

[1] Kaplan, Andreas M., and Michael Haenlein. "Users of the world, unite! The challenges and opportunities of Social Media." Business horizons 53, no. 1 (2010): 59-68.

[2] Liu, B. Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1), (2012). 1-167.

[3] Pang, B., & Lee, L. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2), (2008). 1-135.

[4] Tsytsarau, M., & Palpanas, T. Survey on mining subjective data on the web. Data Mining and Knowledge Discovery, 24(3), (2012). 478-514.

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[6] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017) (pp. 5998-6008).

[7] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. Lexicon-Based Methods for Sentiment Analysis. Computational Linguistics, 37(2), (2011). 267-3071.

[8] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. Sentiment Strength Detection in Short Informal Text. Journal of the American Society for Information Science and Technology, 61(12), (2010). 2544-2558.

[9] Mohammad, S.M., Kiritchenko, S., and Zhu, X. (2013). NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 321-327, Atlanta, Georgia, USA.

[10] Singh, Mrityunjay, Amit Kumar Jakhar, and Shivam Pandey. "Sentiment analysis on the impact of coronavirus in social life using the BERT model." Social Network Analysis and Mining 11, no. 1 (2021): 33.

[11] Dubey, Akash Dutt. "Twitter sentiment analysis during COVID-19 outbreak." Available at SSRN 3572023 (2020).

[12] Medford, Richard J., Sameh N. Saleh, Andrew Sumarsono, Trish M. Perl, and Christoph U. Lehmann. "An" Infodemic": Leveraging High-Volume Twitter Data to Understand Public Sentiment for the COVID-19 Outbreak (preprint)." (2020).

[13] Alhajji, Mohammed, Abdullah Al Khalifah, Mohammed Aljubran, and Mohammed Alkhalifah. "Sentiment analysis of tweets in Saudi Arabia regarding governmental preventive measures to contain COVID-19." (2020).

[14] Severyn, Aliaksei, and Alessandro Moschitti. "Twitter sentiment analysis with deep convolutional neural networks." In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pp. 959-962. 2015.

[15] Preethi, G.; Krishna, P.V.; Obaidat, M.S.; Saritha, V.; Yenduri, S. Application of deep learning to sentiment analysis for recommender system on cloud. In Proceedings of the

2017 International Conference on Computer, Information and Telecommunication Systems (CITS), Dalian, China, 21–23 July 2017; pp. 93–97.

[16] Gupta, Umang, Ankush Chatterjee, Radhakrishnan Srikanth, and Puneet Agrawal. "A sentiment-and-semantics-based approach for emotion detection in textual conversations." arXiv preprint arXiv:1707.06996 (2017).

[17] Peters, M., M. Neumann, M. Iyyer, M. Gardner, C. Lee Clark, and K. Lee. "K., Zettlemoyer, L. Deep Contextualized Word Representations." In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1. 2018.

[18] Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.

[19] Rogers, Anna, Olga Kovaleva, and Anna Rumshisky. "A primer in BERTology: What we know about how BERT works." Transactions of the Association for Computational Linguistics 8 (2021): 842-866.

[20] Prottasha, Nusrat Jahan, Abdullah As Sami, Md Kowsher, Saydul Akbar Murad, Anupam Kumar Bairagi, Mehedi Masud, and Mohammed Baz. "Transfer learning for sentiment analysis using BERT based supervised fine-tuning." Sensors 22, no. 11 (2022): 4157.

[21] Zheng, Zhuoyuan, Yunpeng Cai, and Ye Li. "Oversampling method for imbalanced classification." Computing and Informatics 34, no. 5 (2015): 1017-1037.

[22] Abbas, Muhammad, K. Ali Memon, A. Aleem Jamali, Saleemullah Memon, and Anees Ahmed. "Multinomial Naive Bayes classification model for sentiment analysis." IJCSNS Int. J. Comput. Sci. Netw. Secur 19, no. 3 (2019): 62.

[23] Khadhraoui, Mayara, Hatem Bellaaj, Mehdi Ben Ammar, Habib Hamam, and Mohamed Jmaiel. "Survey of BERT-base models for scientific text classification: COVID-19 case study." Applied Sciences 12, no. 6 (2022): 2891.

[24] Hossin, Mohammad, and Md Nasir Sulaiman. "A review on evaluation metrics for data classification evaluations." International journal of data mining & knowledge management process 5, no. 2 (2015): 1.