IDENTIFYING DEPRESSIVE TWEETS USING NATURAL LANGUAGE PROCESSING (NLP) FRAMEWORKS

Damilola Oladimeji, Laura Garland and Qingzhong Liu

Department of Computer Science, Sam Houston State University,
Texas. U.S.A.

ABSTRACT

The number of patients diagnosed with depression yearly is a growing concern among mental health advocates. Consequently, the effect of this ailment is detrimental to not only the patient but also family members, as well as their jobs or school. Many factors, ranging from hereditary conditions to life-altering experiences, can trigger depression, and symptoms vary between individuals. Hence, the disparity of symptoms in diagnosing depression makes it difficult to identify early on. Fortunately, the prevalence of social media platforms has led to individuals posting updates on various aspects of their lives, particularly their mental health. These platforms now provide valuable data sources for mental health researchers, aiding in the timely diagnosis of depression. In this research, we use sentiment analysis to identify depressed tweets from random tweets. We used six natural language processing frameworks for our classification. They are BERT, XLNet, ALBERT, DeBERTa, RoBERTa, and ELECTRA. Our results show that BERT performs best with an accuracy of 99%, while ALBERT is the model with the lowest accuracy rate at 87%. This research shows that by leveraging NLP frameworks, we can successfully utilize machine learning for the early detection of depression and help diagnose individuals struggling with this ailment.

KEYWORDS

sentiment analysis, depressed tweets identification, BERT, NLP

1. INTRODUCTION

Depression is a form of mental illness that affects individuals differently. The symptoms of this condition are highly personal and tend to be more behavioral than physical [1]. This disease affects more than a person’s feelings; it can impair the patient’s work, school, and familial relationships. Some symptoms of depression include sadness or anxiety, restlessness, insomnia, fatigue, and suicidal thoughts; however, these symptoms vary amongst patients.

According to the Centers for Disease Control and Prevention (CDC), life-altering events like giving birth, losing a loved one, undergoing a financial crisis, and medication or alcohol use can trigger depression [2]. Furthermore, this illness can affect anyone, regardless of their age. The CDC also reports that depression impacts roughly 16 million adult Americans annually, and one in every six people will experience depression at some point in their lives [3]. Since patients experience varying symptoms of depression, diagnosing this illness early on is challenging.

With the rise of social media, mental health researchers have access to valuable data sources that can assist in the early diagnosis of depression. Platforms like Twitter allow users to post updates

David C. Wyld et al. (Eds): SIPM, ITCA, CMIT, FCST, CoNeCo, SAIM, ICITE, ACSIT, SNLP -2023
pp. 191-199, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130920

and express their emotions, which can indicate the early stages of depression. Consequently, we can analyze the texts on this platform to detect various events or illnesses, like depression, using machine learning techniques. One example of this is natural language processing (NLP).

A growing field within NLP is sentiment analysis, which analyzes text to determine the meanings conveyed within the text. For example, this can determine user intentions and emotions. This is particularly important for social media, where users can upload similar posts with slight differences that change the overall meaning of the post. Therefore, sentiment analysis can identify users who are angry or depressed before they commit an irreversible act.

In this research, we use sentiment analysis to analyze over 14,000 tweets retrieved from Twitter to determine if the user is depressed. We compare six NLP algorithms, including XLNet, BERT, RoBERTa, ALBERT, DeBERTa, and ELECTRA, to determine which can accurately identify depression from user tweets.

The rest of this paper comprises the following: Section II discusses existing literature related to NLP and sentiment analysis. Then, sections III and IV provide our methodology and results, respectively. Finally, Section V provides our conclusion and future works.

2. Relevant Works

The following section discusses existing literature related to sentiment analysis, with a particular focus on social media.

Nair et al. [5] investigate tweets associated with the COVID-19 pandemic. Researchers divide tweets into positive, negative, and neutral classifiers. It maintains consistent data preprocessing for all three-sentiment analysis algorithms and uses logistic regression for discriminative classification and BERT to simultaneously provide long short-term memory (LSTM) in both directions. BERT had the highest performance at 92% accuracy.

This paper [6] classifies depression-related words into low, medium, and high categories using the term frequency-inverse document frequency (TF-IDF) and linguistic inquiry and word count (LIWC). Mustafa et al. [7] also use the SemEval tweet collection dataset to identify user opinions.

This study uses pre-trained word embeddings from Word2Vec and continuous-bag-of-words and also uses Delta TF-IDF to provide weighted word embeddings. This approach produced the highest accuracy in this study at 65.3%, higher than Support Vector Machine (SVM), and TF-IDF.

In this paper [8], a novel approach called topic-enriched depression detection model (TDDM) is introduced, which extracts user posts using CRFTM and encodes them using XLNet to predict depression from social media posts. Researchers then compared this method to BERT, ALBERT, XLNet, convolutional neural network (CNN), and BiLSTM algorithms and found that it provided the best performance at 83% accuracy. Similarly, Gao et al. [9] propose a sentiment information-based network model (SINM), which uses an LSTM transformer encoder to increase the stability of sentiment analysis on Chinese texts. In this study, researchers use two company-generated datasets, ChnSentiCorp and ChnFoodReviews.

Panikar et al. [10] propose a modified NLP pipeline for performing granular sentiment analysis on texts written in Hindi. This includes domain-specific lexicons, parsing based on expressions, and identifying and implementing a set of rules for grammar. Araslanov et al. [11] also used

Naïve Bayes classifier and logistic regression on tweets in the Russian language to determine user emotions. It found that logistic regression with feature selection produces the most accurate results of these two algorithms at 80%.

Furthermore, Khan et al. [12] use several algorithms, including Naïve Bayes, Random Forest, SVM, Decision Tree, K-Nearest Neighbor, and XG Boost to determine whether data retrieved from Bengali paragraphs are sad or happy. Of these algorithms, Naïve Bayes performed best at 88% accuracy for both categories, while the rest saw varying results for both emotions.

Our research differs from existing literature as we implement a diverse range of NLP models for sentiment analysis. Additionally, each model architecture was built using varying techniques due to our utilization of pre-trained models, which helped to enhance the accuracy of the results.

3. METHODOLOGY

In this section, we will describe the data sets, pre-processing steps, and feature selection used in this research. Furthermore, we will detail each NLP framework we use to carry out a sentiment analysis on the tweet data. Figure 1 shows the methodology used in this research. We describe each phase in detail below.

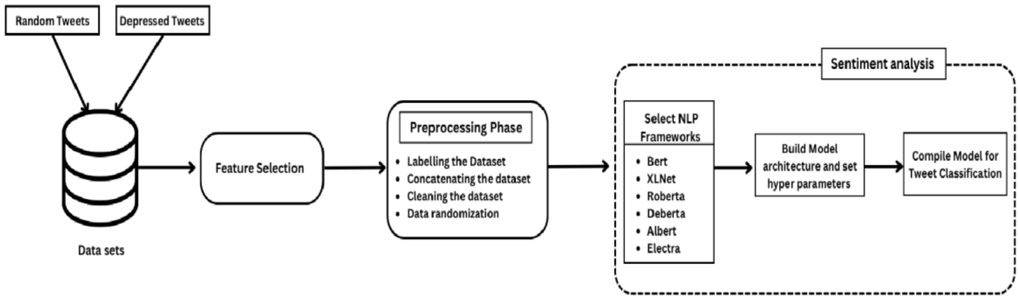

Figure 1. The proposed methodology adopted in this research.

3.1. Data Sets

We used two datasets retrieved from Kaggle to conduct our analysis. The first dataset consisted of depressed tweets [13], while the second consisted of random tweets [14]. We selected 3,200 rows from the depressed CSV file and 12,000 rows from the random tweets CSV file.

3.2. Feature Selection

To reduce the number of inputs in our dataset, we performed feature selection, ensuring that we chose data points useful in identifying depressed tweets. Initially, the depressed CSV file contained eight columns, with column five containing depressed tweets. In contrast, the random tweet CSV file had four columns, with the random tweets located in the " SentimentText " column. After analyzing both CSV files, we realized we only required the text column to improve performance.

The features selected in this research are listed as follows:

- **SentimentText:** This column consists of text used as input in this research. It represented the tweets as composed of depressed and random textual information. The aim is that the selected NLP models will be able to understand, classify, and predict these tweets into their respective labels after training.

- **Label:** This column represents the output used in this research. These data points predict whether a given text in the “SentimentText” column is depressed or random.

3.3. Preprocessing Phase

In the preprocessing phase, we carried out various steps, which are discussed in the subsections below.

3.3.1. Data Set Labeling and Concatenation

Recall that we used two CSV files, one with entirely depressed tweets and the other with random tweets. We classified both datasets by including an additional column called “label” in each of their respective files. In the depressed CSV file, we assigned the value "1," whereas, in the random file, we assigned the value “0” to the label column.

After assigning the labels, we concatenated both files to create the dataset used in this research. After concatenation, we removed null values and had a total of 14,313 rows, compromising both depressed and random tweets, as shown in Table 1.

Table 1. Tweet Classification Counts in the Dataset.

Labels	**Tweets**	**Counts**
0	Random	12000
1	Depressed	2313

3.3.2. Cleaning the Dataset

We cleaned our dataset in order to ensure the text column we used to train our models was accurate and relevant. We removed duplicate columns, usernames, emojis, special characters, and website links. Additionally, we punctuated contractions to make them complete sentences and changed all text to lowercase.

3.3.3. Data Randomization

In the dataset, we conducted cluster sampling to randomize it. Implementing this reshuffled the dataset so that each batch of tweets, depressed and random, would be represented when training our model. With our dataset being so large, the randomization technique significantly reduced bias.

Following this step, we visualized the most frequently used words in depressed tweet rows using word clouds. Word clouds ensure we are targeting the most fitting words associated with depression in our depressed tweets.

3.4. Sentiment Analysis

Here, we discuss the various NLP frameworks (models) used in our research and detail the steps taken to build and train each one.

3.4.1. NLP Models

This study applies six NLP models to our dataset to determine which can provide accurate results.
Details on each of these models are as follows:

- BERT: Bidirectional Encoder Representations from Transformers, also known as BERT [15], is a popular machine learning model in NLP. It consists of many layers of transformer encoders, with the number of layers ranging from 12 for the BERT base model to 24 for the BERT large model.
- RoBERTa: This is the acronym for robustly optimized BERT pretraining approach [16]. Researchers from Facebook created RoBERTa after discovering that the original BERT model experienced issues with undertraining data. Therefore, this model provides the highest possible number of training iterations, increases text data substantially, and introduces a dynamic masking approach.
- DeBERTa: Also known as decoding-enhanced BERT with disentangled attention, DeBERTa [17] introduces disentangled attention and enhances the mask decoder to create a better model than both BERT and RoBERTa. Using disentangled attention introduces a new self-attention process named position-to-content which the prior two models lack.
- ALBERT: This is the acronym for a lite BERT [18]. This model addresses issues that can arise from the original BERT model, such as long training times and memory constraints, by introducing two techniques to reduce parameters. The first of these techniques is factorized embedding parameterization, and the second is cross-layer sharing.
- XLNet: XLNet is a powerful machine learning model that adopts the bidirectional learning ability of BERT with Transformer-XL's autoregressive model [19]. Introducing an autoregressive aspect to this hybrid model removes the issue of data corruption found in previous models like BERT while retaining the benefits of bidirectional learning.
- ELECTRA: Unlike BERT and its associated models, ELECTRA [20] removes masking and introduces a concept called replaced token detection. This provides a more efficient model for sample data, as it requires less computation than BERT models. It does this by using a small language model to replace some of the tokens and asking the pre-trained discriminator to decide whether the tokens are originals or replacements. This approach requires the model to learn from all inputs rather than the masked amount.

3.4.2. Building and Training the Models

Here, we highlight the hyperparameters set to train the models and fine-tune them for better results. Additionally, we emphasize the tokenizers and pre-trained models used to build each model's architecture. We used pre-trained models from TensorFlow to build each model's architecture in this research. Due to the low training and effort required to build each model's architecture, these pre-trained models proved effective in our research.

Table 2 shows the tokenizers and imported pre-trained models used to build the architecture for our models. We used 140 as our input length, trained the model using four layers/epochs, and used the Adam optimizer to compile our model.

Table 2. Tokenizers, pre-trained models, and learning rate (lr) were used for the NLP models.

NLP	Tokenizer	Pre-trained model	lr
BERT	bert-base-uncased	bert-base-uncased	3e-5
XLNet	XLNetTokenizer	xlnet-base-cased	3e-5
RoBERTa	AutoTokenizer	roberta-base	5e-5
ALBERT	AutoTokenizer	albert-base-v2	5e-5
ELECTRA	AutoTokenizer	google/electrasmall-discriminator	5e-5
DeBERTa	AutoTokenizer	microsoft/debertabase	3e-5

4. RESULTS

In this section, we show the loss function adopted, the accuracy results, and the confusion matrix of each model used to classify the tweets in this research.

4.1. Loss Function

We used a Binary Cross-Entropy (BCE) loss function for this research since we have only two classes, “1” (depressed tweets / negative case) and “0” (random tweets / positive case), in our dataset. The cross-entropy loss function is one of the most widely used classification losses, as it displays how accurately a machine learning model classifies a dataset with respect to the ground truth labels. Equation 1 shows the formula for the BCE loss function used in this research.

$$Loss_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(Y_i.\log \hat{Y_i} + (1 - Y_i).\log(1 - \hat{Y_i})) \quad (1)$$

where:

n = the total number of samples
Y_i = the real values
$\hat{Y}_i$ = the predicted values
$Y_i.\log\hat{Y}_i$ = positive case error
$(1 - Y_i).\log(1 - \hat{Y}_i)$ = negative case error $-$ = Loss from total error inversion

4.2. Accuracy of the NLP Frameworks

After building and compiling our models, we obtained the accuracy of each model, which is displayed in Table 3. Almost all the NLP models had accuracy in the high 90 percentile, with only ALBERT having an accuracy of 87%. The highest-performing model in our tweet classification research was the BERT model, with an accuracy of 99.95%.

Table 3. The accuracy in percent of the NLP frameworks used for the tweet classification.

NLP Model	Accuracy (%)
BERT	99.95
XLNet	99.89
ROBERTA	99.91
ALBERT	87.48
ELECTRA	99.91
DeBERTa	99.92

4.3. Confusion Matrix

Furthermore, we used a performance measurement tool known as the confusion matrix to measure the NLP model's performance. Figure 2 shows the confusion matrix results for each model used in this research.

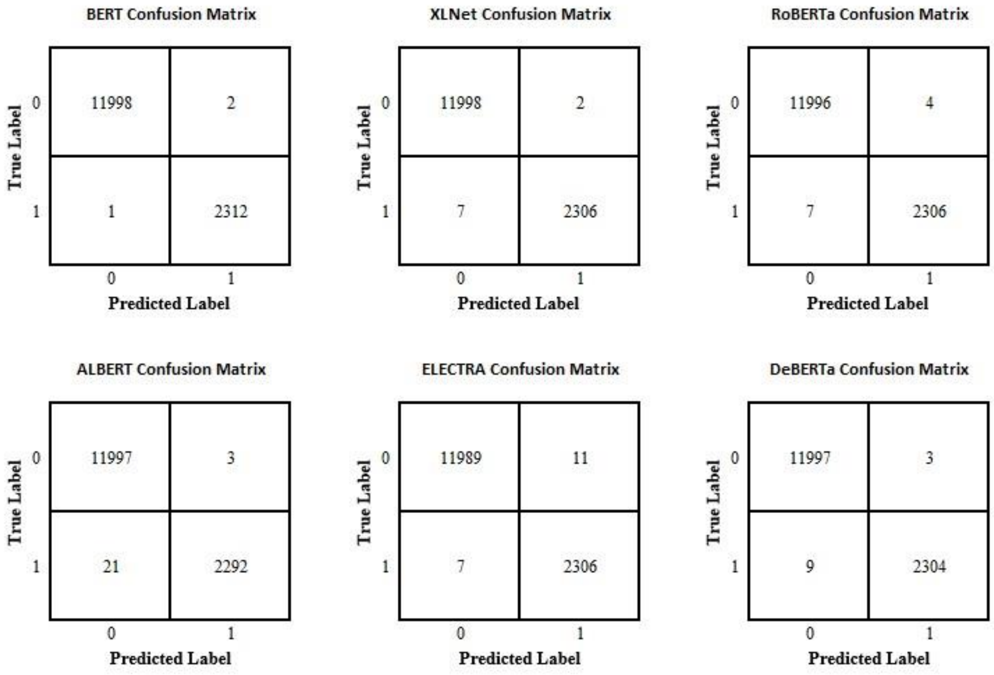

Figure 2: (a) BERT Confusion Matrix (b) XLnet Confusion Matrix (c) RoBERTa Confusion

Matrix (d) ALBERT Confusion Matrix (e) ELECTRA Confusion Matrix (f) DeBERTa Confusion Matrix

Table 4 further illustrates the confusion matrix in this research. We describe the features used in the confusion matrix below:

- True Positive (TP): The model accurately predicted the text as random, and the actual label is 0. This is located at the top left segment of the figure.

- True Negative (TN): The model accurately predicted the text as depressed, and the actual label is 1. This is located at the bottom right segment of the figure.
- False Positive (FP): The total number of predicted random texts is depressed. This is located at the top right segment of the figure.
- False Negative (FN): The total number of depressed texts predicted as random. This is located at the bottom left portion of the figure.

Table 4 shows the total number of tweets predicted correctly as depressed and the tweets that were depressed and falsely pressed as random by each of the models used in this research on our dataset with 14313 data points. Recall that from Table 1, the depressed tweets in our dataset are 2,313, and the random tweets are 12,000. Table 4 shows BERT produced the highest accuracy, correctly predicting 2312 depressed tweets and 11998 random tweets.

Table 4. The Confusion Matrix of the NLP models

NLP Model	TP	TN	FP	FN
BERT	11998	2312	2	1
XLNET	11998	2306	2	7
RoBERTA	11996	2306	4	7
ALBERT	11997	2292	3	21
ELECTRA	11989	2306	11	7
DeBERTa	11997	2304	3	9

Alternatively, ALBERT produced the least accurate model by inaccurately predicting the most depressed tweets as random tweets in this research.

5. Conclusion

Using NLP frameworks to identify depressed tweets can be an effective tool for the timely detection of depression in patients, thereby assisting mental health professionals in adequately providing timely support to these patients. In our research, we used six NLP models, with five of them accurately classifying depressed tweets from random tweets with 99% accuracy. While this approach has some limitations, such as inaccurately classifying some tweets as random when they were, in fact, depressed, it shows the great potential of technology in addressing mental health issues. Future direction can be targeted towards optimizing the pre-trained models to increase accuracy rates amongst these frameworks further.

Acknowledgement

The support for this study from the SHSU Institute of Homeland Security is highly appreciated.

References

[1] A. Stringaris, “What is depression?” pp. 1287–1289, 2017.

[2] ThePhoenixRC, “How to explain depression to someone: The phoenix,” Mar 2023. [Online].Available: https://www.thephoenixrc.com/how-toexplain-depression-to-a-loved-one-who-doesnt-understand/

[3] "Mental health conditions: Depression and anxiety," Sep 2022. [Online]. Available:https://www.cdc.gov/tobacco/campaign/tips/diseases/depressionanxiety.html

[4] IBM, "What is natural language processing?"

[5] A. J. Nair, V. G, and A. Vinayak, "Comparative study of twitter sentiment on covid - 19 tweets," in 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 1773–1778.

[6] R. U. Mustafa, N. Ashraf, F. S. Ahmed, J. Ferzund, B. Shahzad, and A. Gelbukh, "A multiclass depression detection in social media based on sentiment analysis," in 17th International Conference on Information Technology–New Generations (ITNG 2020). Springer International Publishing, 2020.

[7] R. Othman, Y. Abdelsadek, K. Chelghoum, I. Kacem, and R. Faiz, "Improving sentiment analysis in twitter using sentiment specific word embeddings," in 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), vol. 2, 2019, pp. 854–858.

[8] W. Gao, B. Yang, Y. Wang, and Y. Fang, "Depression detection in social media using xlnet with topic distributions," Journal of Computers, vol. 33, no. 4, pp. 95–106, 2022.

[9] G. Li, Q. Zheng, L. Zhang, S. Guo, and L. Niu, "Sentiment infomation based model for chinese text sentiment analysis," in 2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), 2020, pp. 366–371.

[10] R. Panikar, R. Bhavsar, and B. V. Pawar, "Sentiment analysis: a cognitive perspective," in 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, 2022, pp. 1258–1262.

[11] E. Araslanov, E. Komotskiy, and E. Agbozo, "Assessing the impact of text preprocessing in sentiment analysis of short social network messages in the russian language," in 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020, pp. 1–4.

[12] M. R. H. Khan, U. S. Afroz, A. K. M. Masum, S. Abujar, and S. A. Hossain, "Sentiment analysis from bengali depression dataset using machine learning," in 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1–5.

[13] Sharanharsoor, "Twitter sentiment analysis," Mar 2023. [Online]. Available: https://www.kaggle.com/code/sharanharsoor/twittersentiment-analysis/data

[14] David, "Twitter sentiment," Nov 2017. [Online]. Available: https://www.kaggle.com/datasets/ywang311/twitter-sentiment

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692, 2019.

[17] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disentangled attention," arXiv preprint arXiv:2006.03654, 2020.

[18] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," arXiv preprint arXiv:1909.11942, 2019.

[19] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," Advances in neural information processing systems, vol. 32, 2019.

[20] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pretraining text encoders as discriminators rather than generators," arXiv preprint arXiv:2003.10555, 2020.