Depression detection from social media comments using deep learning

SYED MD. AHNAF HASAN, Dept. of Computer Science and Engineering, Brac Univeristy, Bangladesh
MD. TANVIR HOSSAIN, Dept. of Computer Science and Engineering, Brac Univeristy, Bangladesh

CCS Concepts: • **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Additional Key Words and Phrases: Do, Not, Us, This, Code, Put, the, Correct, Terms, for, Your, Paper

1 INTRODUCTION

2 RELATED WORKS

There have been several advancements in detecting depression from text data. Shah et al. in 2020 suggested a hybrid model of detecting depression based on BiLSTM, word embeddings, and metadata features of social media posts [14]. The most effective combination of Word2Vec embeddings and BiLSTM demonstrates an F1 score of 0.81. While the approach provided accurate classifications, it discloses the issue of timely depression detection and calls more studies for closing this gap in depression detection time. Samanvitha et al. in 2021 used the CLEF 2020 dataset and the following models: Naive Bayes, Logistic Regression, Random Forest, and SVM to detect depression from posts in social media [12]. The model with the highest accuracy is the Naïve Bayes classifier had the highest accuracy prerequisites, 83%, which is 2% - 4% higher compared to other models. Zahoor et al. in 2022 have reviewed sentiment analysis and category classification of restaurant reviews based on the different algorithms for Naïve Bayes, Logistic Regression, SVM, and Random Forest [15]. Thus, the results of the dataset obtained from "SWOT's Guide to KARACHI's Restaurants Cafes Dhabas HBFE & Takeouts" showed that Random Forest has the highest accuracy of 95%. Lestandy et al. in 2023 have studied the effect of word embedding dimensions on depression data classification with BiLSTM [8]. They used the Kaggle Reddit Depression Dataset and GloVe and Word2Vec embeddings with BiLSTM. Word2Vec achieved the best performance with 96.22% accuracy when trained with 500 dimensions with BiLSTM, better than GLoVe. Rizwan et al. in 2022 examined the possibility of detecting depression from social media texts using small transformer-based language models [11]. Through the application of the Twitter dataset labelled with VADER and TextBlob, depression intensity classification for models such as Electra Small Generator and Albert Base V2 was conducted. The former outperformed all models by obtaining the highest F1 score of 89%. AlSagri et al. in 2020 utilized machine learning tools for depression detection among Twitter users, with specific features produced by the network behavior and tweets extraction [3]. Accuracy of SVM is higher compared with other models, although all the metrics are lower than 80%. Data collection was assisted by the Twitter Search API and annotated manually. Borba de Souza et al. in 2022 used DAC Stacking in their study to classify depression, anxiety and their comorbidity in the SMHD Dataset [4] . They used LSTM, CNN and a Hybrid LSTM-CNN architecture and achieved impressive results yielding f-measures near 0.79 in both depression and anxiety cases. Sanga et al. in 2023 determined depression detection in a digital framework with textual analysis [13]. Attention-enabled deep learning models were trained, including an LSTM, GRU, AlBERT, DistilBERT, CNN, and Attention, on a wide range of Reddit and Kaggle datasets. The deployed

Authors' Contact Information: Syed Md. Ahnaf Hasan, syed.md.ahnafhasan@g.bracu.ac.bd, Dept. of Computer Science and Engineering, Brac Univeristy, Dhaka, Bangladesh; Md. Tanvir Hossain, webmaster@marysville-ohio.com, Dept. of Computer Science and Engineering, Brac Univeristy, Dhaka, Bangladesh.

models demonstrated 1.52% accuracy in the mean, with DistilBERT-BiLSTM achieving up to 94.93% accuracy and indicating AUC score of 0.9501. Akter et al. in 2023 presented the development of a novel cyberbullying detection model from a dataset obtained during the TRAC-2 Workshop [2]. The dataset consists of 25,000 comments gathered in three languages: English, Bengali, and Hindi. Several methods were used to implement the model: LSTM, BiLSTM, LSTM-Autoencoder, Word2vec, BERT, and GPT-2. Achieved results differ significantly, with proposed models' raw English reaching up to 99% accuracy, semi-noisy Bangla obtaining 95% accuracy, and noisy English 92% accuracy. Flores et al. in 2023 presented DeepScreen through the use of temporal facial landmark features from the DAIC-WOZ dataset, incorporating GRU-D and BRITS. they achieved a significant leap of 13.4% in depression screening F1 score, which was 0.85 [6]. Akter et al. in 2022 discussed the issue of a growing amount of violent content on social media. Author used Hindi, Bangla, and English datasets and dealt with the problem of data scarcity in several ways, including most importantly the use of machine translation [1]. Different deep learning methods, such as LSTM, BiLSTM, BERT, and GPT-2, are then used, and the results indicate that BERT is the best performing model; BERT records an accuracy of 0.79 when used for English data, while BERT Multilingual records.72 and .69 on Bangla and Hindi datasets respectively.

3 METHODOLOGY

The workflow of this research work is shown in Fig. 1. At first the data is split into train, validation and test sets. The splits for train, validation and test are 70%, 15% and 15% respectively. Later the tokenization is performed on the splits. Two main types of models are used in this work: RNN models and BERT Base Uncased which is a pretrained transformer model. For the RNN based models, a tokanizer object created from the tokenizer class from keras is fitted on the train split. Later the train, test and validation splits are transformed to integer sequences via the tokenzier object. Later padding and truncation operations applied on the tokenized sequences. For the BERT Base Uncased model, we use the BERT tokenizer available in the Hugging face transfomers library. The BERT tokenizer object in this case, handles tokenization, padding and truncation. The maximum length chosen to be kept after padding and truncation is 256. The newly generated train sequences are used to train the deep learning models, while the validation set is evaluated simultaneously to stop the model from training if it starts overfitting via early stopping. Consequently, the test split is evaluated using the trained models. Finally, we use LIME to interpret the local predictions of the models.

3.1 Dataset

The "Depression: Reddit Dataset (Cleaned)" dataset [7], [8] is used for training and evaluating the deep learning models. The dataset is available in the link: https://www.kaggle.com/datasets/infamouscoder/depression-reddit-cleaned/code. It contains 7731 labeled user comments curated from various comments in Reddit. The label '1' is used to denote depressed text while the label '0' indicates non depressed text. From Fig. 2 we can see that the dataset is nearly balanced. The dataset consists of 3900 non depressive texts and 3831 depressive texts. Two instances of each class of texts are shown in Table. 1. The dataset contains text data which are cleaned, as there are no punctuation marks, symbols, etc., furthermore the text is in lower case.

3.2 Tokenization

Deep learning models require the input which is fed to them to be numerical in nature. Therefore, the text that will be fed to the deep learning models, need to be converted to numerical format. A way to do this is tokenization. In this process, an integer value is assigned to a token most often words in the text. For the RNN models, the a tokenizer object created using Keras library is

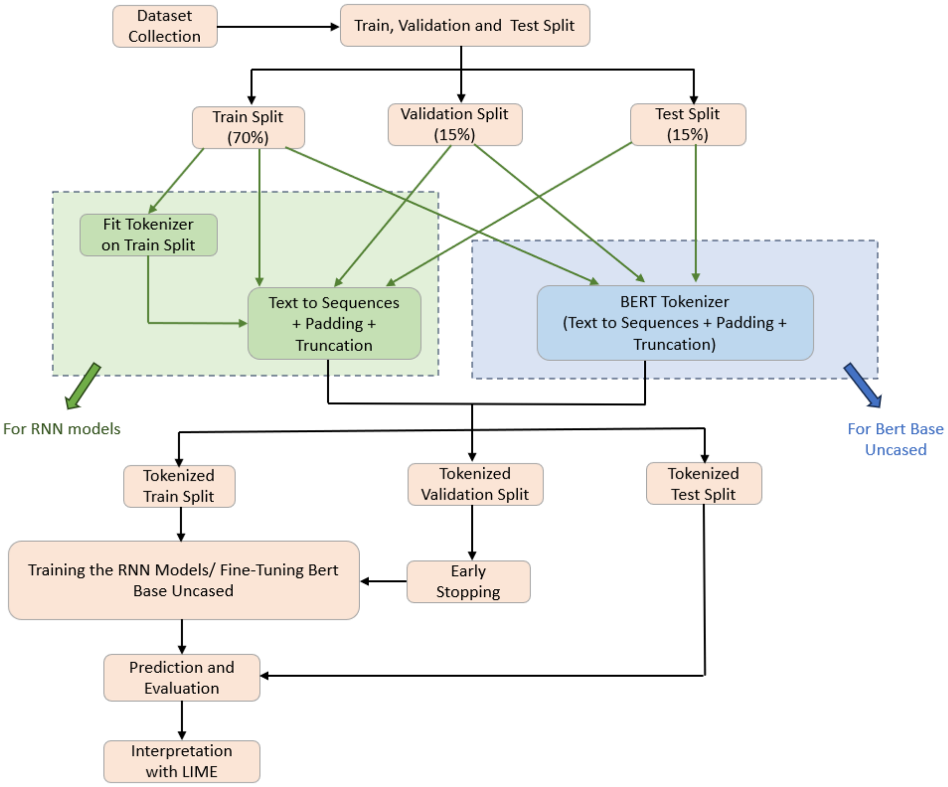
Dataset Collection
Train, Validation and Test Split
Train Split (70%)
Validation Split (15%)
Test Split (15%)
Fit Tokenizer on Train Split
Text to Sequences + Padding + Truncation
BERT Tokenizer (Text to Sequences + Padding + Truncation)
For RNN models
For Bert Base Uncased
Tokenized Train Split
Tokenized Validation Split
Tokenized Test Split
Training the RNN Models/ Fine-Tuning Bert Base Uncased
Early Stopping
Prediction and Evaluation
Interpretation with LIME

Fig. 1. Methodology

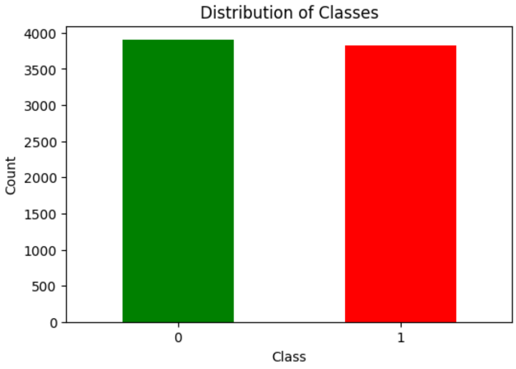

Fig. 2. Dataset Distribution

fitted on the train split. Later the train, test and validations splits are tokenized using the tokenizer object. In Fig. 3. we can see the distribution of lengths of each sequence instance. We find that the maximum sequence length among all the instances is 4239. However, most of the instances have a sequence which fall in the range of 0-250. So, we choose 256 has the maximum length of the sequences. Any shorter sequence will be zero-padded and longer sequences will be truncated. For

Table 1. Dataset Instances

clean_text	is_depression
i dont even deserve to live	1
sooo sick of the snow ughh	0

BERT Base Uncased, we use the BERT Tokenizer from Hugging face transformers library which handles tokenization, padding and truncation.

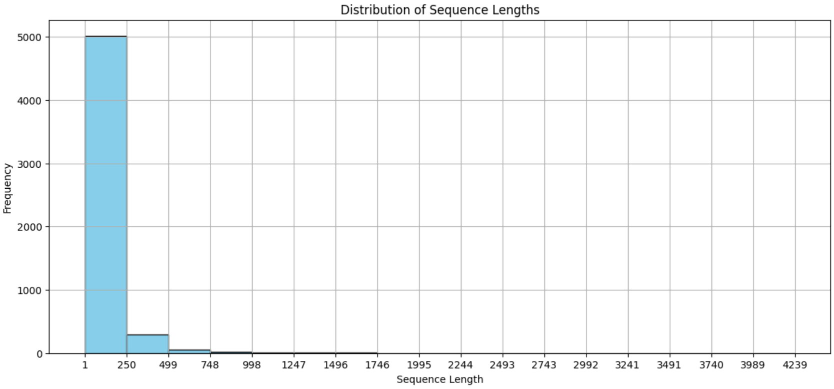

Fig. 3. Distribution of Sequence Length

3.3 GloVe

GloVe: Global Vectors for Word Representation is a technique to generate word embeddings [9]. Deep learning models work with numerical values, i.e. data they are trained on and the data they evaluate all have to be converted to numerical format. However, only converting words in a text to a single numerical value or to a one-hot encoding vector is not enough, as the meaning of a word is not effectively captured in this way. Word embeddings are vectors assigned to a word with a very high dimension. Each of these dimensions carry an aspect of a word's meaning. GloVe is an efficient technique to generate these embeddings. Word embeddings generated by GloVe can be found in various dimensions such as 25, 50, 100, 200, etc. In this project we have used the 100 dimensional word embeddings generated by GloVe.

3.4 Deep Learning Models

3.4.1 RNN Models. In this work, performance of six RNN models are evaluated. Each of the models use either unidirectional or bidirectional variations of the RNN layers. Three RNN layers are used: Simple RNN, LSTM and GRU, which are available in Keras library. Each of these layers will have 32 units. The architecture of these models in general is shown in Fig. 4. The first hidden layer will be the embedding layer. The GloVe pretrained embeddings will be used here to transformer the incoming tokens to a higher dimensional embeddding. The chosen dimesion of embeddings is 100. The next hidden layer contains either of the mentioned RNN layers, which will have 32 units. The Simple RNN layer has an incoming input and a feedback path carrying its hidden state value. It uses the hidden state value from previous time step and current time step input to generate output. It works consecutively as sequences of data like in the case of text is passed to it. The LSTM is a variation of RNN that uses separate paths for propagating long term memory and hidden state. It changes the values of the memory state based on the importance of the current input and the previous time step hidden state. It uses 3 gates: input gate, forget gate and output gates to accomplish its task. It solves the vanishing gradient problem of basic RNN. GRU is another variation of RNN that is similar to LSTM, but has one less gate than LSTM, making it faster. It does not use a separate path to propagate long term memory state, rather it uses the hidden state for that task. The final layer in the RNN models is the output layer that will have a single dense unit with sigmoid activation function to generate the probabilities of the classes.

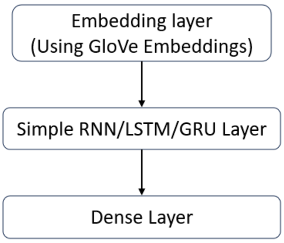

Fig. 4. Architecture of RNN models

3.4.2 Bert Base Uncased model. Transformers are attention based deep learning models with a large number of parameters which can effectively capture information from sequences by using parallel computations. Moreover, the attention mechanism can be applied multiple times on the same input to capture more context and information; this mechanism is called multi-head attention mechanism. Due to the large number of parameters of popular transformer models, it is computationally very expensive to train them from scratch. Therefore, in most cases for downstream task such ours, a pretrained transformer model is used to be fine-tuned on a dataset. BERT: Bidirectional Encoder Representations from Transformers, is a transformer based language representation model [5] which was trained on a huge corpus via self-supervised learning.The Bert Base (Uncased) model used in this work is a pretrained BERT model available in hugging face transformers library. This model has got 110 million parameters, which helps it to effectively capture information from sequence data.

3.5 LIME

LIME: Local Interpretable Model-agnostic Explanations is an explainable AI (XAI) model which is used to explain predictions made by deep learning models [10]. Deep learning models are inherently black box in nature due to their complex structure. Therefore, it is not clear how the model made a prediction or came to a conclusion for a particular instance of data. XAI models are used to explain why the deep learning model drew a particular conclusion on an instance based on the feature of the instance. LIME is used to generate local explanations for a particular instance of data by generating perturbed data points in the local neighborhood of the data point and uses a linearly interpretable model which best approximates the prediction of the black box model on that particular data.

Table 2. Summary of Hyperparameters

Parameter Name	Value
Simple RNN/LSTM/GRU units	32
Word Embedding dimension (For RNN models)	100
Vocabulary Size of Tokenizer (For RNN models)	10000
Max length of sequences (After padding and truncation)	256
Early Stopping Patience	3
Number of epochs	50

3.6 Hyperparameters

The hyperparameters used in this research, have been summarized in Table 2. The number of the units of RNN layers will be 32. For, RNN models, the size of the word embeddings will be 100, the vocabulary size will be 10,000. The maximum length of tokenized sequences after padding and truncating will be 256. Early stopping patience will be 3; if the accuracy of validation set while training does not improve for next consecutive 3 epochs, the training will stop. The number of epochs to train each model is chosen to be 50.

3.7 Performance Metrics

Performance of all the models were evaluated using four metrics: accuracy, precision, recall and F1 score. Accuracy is the ratio of the number of correctly predicted instances to the total number of instances. It works very good for balanced data but is misleading for imbalanced data. Precision is the ratio of true positives to the summation of true positives and false positives, i.e. the percent of predictions which are correct out of all the predictions made by the model for a particular class. Recall on the other hand is the ratio of true positives to the summation of true positives and false negatives which indicates the fraction of the actual labels that the model predicted correctly for a class. F1 score is a metric that is a balance between precision and recall; it is the harmonic mean of the precision and recall. Precision, recall and F1 score provide important insights to a model's performance especially for imbalanced data. Macro average, micro average or weighted average can be taken for precision, recall and F1 to combine the scores of all the classes.

4 EXPERIMENT RESULTS AND ANALYSIS

5 CONCLUSION

ACKNOWLEDGMENTS

We would like to express our gratitude to Annajiat Alim Rasel sir for his supervision and guidance on this project.

REFERENCES

[1] Mst Shapna Akter, Hossain Shahriar, Nova Ahmed, and Alfredo Cuzzocrea. 2022. Deep learning approach for classifying the aggressive comments on social media: Machine translated data vs real life data. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 5646–5655.

[2] Mst Shapna Akter, Hossain Shahriar, Alfredo Cuzzocrea, Fan Wu, and Juanjose Rodriguez-Cardenas. 2023. A Trustable LSTM-Autoencoder Network for Cyberbullying Detection on Social Media Using Synthetic Data. In *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 5418–5427.

[3] Hatoon S AlSagri and Mourad Ykhlef. 2020. Machine learning-based approach for depression detection in twitter using content and activity features. *IEICE Transactions on Information and Systems* 103, 8 (2020), 1825–1832.

[4] Vanessa Borba de Souza, Jeferson Campos Nobre, and Karin Becker. 2022. DAC stacking: A deep learning ensemble to classify anxiety, depression, and their comorbidity from Reddit texts. *IEEE Journal of Biomedical and Health Informatics*

26, 7 (2022), 3303–3311.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Ricardo Flores, Avantika Shrestha, and Elke A Rundensteiner. 2023. DeepScreen: Boosting Depression Screening Performance with an Auxiliary Task. In *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 5213–5222.

[7] Manuel Kanahuati-Ceballos and Leonardo J Valdivia. 2024. Detection of depressive comments on social media using RNN, LSTM, and random forest: comparison and optimization. *Social Network Analysis and Mining* 14, 1 (2024), 1–16.

[8] Merinda Lestandy et al. 2023. Exploring the Impact of Word Embedding Dimensions on Depression Data Classification Using BiLSTM Model. *Procedia Computer Science* 227 (2023), 298–306.

[9] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[10] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[11] Muhammad Rizwan, Muhammad Faheem Mushtaq, Urooj Akram, Arif Mehmood, Imran Ashraf, and Benjamín Sahelices. 2022. Depression classification from tweets using small deep transfer learning language models. *IEEE Access* 10 (2022), 129176–129189.

[12] S Samanvitha, AR Bindiya, Shreya Sudhanva, and BS Mahanand. 2021. Naïve Bayes Classifier for depression detection using text data. In *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)*. IEEE, 418–421.

[13] Prabhav Sanga, Jaskaran Singh, and Prakhar Priyadarshi. 2023. Unmasking Depression via Attention-modulated Text Analysis using Deep Learning. In *2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI)*. IEEE, 335–340.

[14] Faisal Muhammad Shah, Farzad Ahmed, Sajib Kumar Saha Joy, Sifat Ahmed, Samir Sadek, Rimon Shil, and Md Hasanul Kabir. 2020. Early depression detection from social network using deep learning techniques. In *2020 IEEE region 10 symposium (TENSYMP)*. IEEE, 823–826.

[15] Kanwal Zahoor, Narmeen Zakaria Bawany, and Tehreem Qamar. [n. d.]. Evaluating text classification with explainable artificial intelligence. *Int J Artif Intell ISSN* 2252, 8938 ([n. d.]), 8938.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009