Depression Detection on Twitter Dataset

Student Name: Rifat Hossain Rafi
Student ID: a1836604

July 10, 2023

THE UNIVERSITY
of ADELAIDE

July 10, 2023

Table of Context

1 Introduction

According to the WHO in 2019, mental illness is one of the main causes of disability globally. This is a serious issue that needs to be addressed. Depression, sleeplessness, bipolar disorder, schizophrenia, anxiety disorders, and drug or alcohol use disorders are examples of mood or personality problems that are classified as mental illnesses. Only a small percentage of those who need treatment for mental illness out of millions of people receive it [1].

It is customary to rely on clinical data, which are typically gathered after the sickness has manifested and been reported. Furthermore, these clinical data are insufficient because the majority of those who suffer from mental illness do not seek help [2]. Surveys can also be collected via mail, phone, or in-person interviews, although these methods are more expensive and time-consuming. Utilizing population data to better understand mental health issues has advanced thanks to social media analysis. In order to discover mental problems among the general population, social media post analysis can be a valuable alternative.

Twitter's explosive expansion as a venue for individuals to share their ideas and attitudes towards a wide range of topics has drawn a lot of attention to Twitter sentiment analysis. Approaches to Twitter sentiment analysis typically concentrate on identifying the sentiment of certain tweets. The majority of the current research on tweet-level sentiment detection uses either supervised learning or lexicon-based techniques. For sentiment classifier learning, supervised learning methods need training data. Typically, training data for Twitter is obtained by either presuming that the polarities of tweets (positive, negative, or neutral) can be determined using emoticons or by accepting the consensus of the sentiment detection websites' results.Furthermore, supervised techniques need to be retrained as new data is added because they depend on the domain [3].

They then used a variety of techniques, such as vocal entertainment, prosody, and speech rate, to examine therapist empathy. In order to detect depression from tweets, we would like to propose a hybrid approach named **D**etection Using **L**STM & **C**NN **(DLC)** for our project.

2 Related Work

The majority of currently used methods to determine the sentiment polarity of tweets involve text pre-processing (e.g., POS, eliminating URLs, extending acronyms, substituting negative stops, stemming, and eliminating mentions) to decrease the the tweets' overall noise level. The theory is that information prior to Processing makes text less noisy, which should aid to increase the efficiency of the more refined and quicken the classicalization procedure [10][11].

A certain geographic area and a specified time period were taken into consideration by Carchiolo et al. [12]. They have made use of SNOMED-CT, a computer programme that is methodically arranged and capable of accurately detecting medical terminology. Liu et altopic-adaptive .'s sentiment classification model [13] uses mixed labelled data derived from several domains and common text features to function as a classifier. Utilizing point-wise mutual information and information retrieval, text feature data was extracted.

The use of text pre-processing in movie review sentiment analysis was examined by Haddi et al. [14]. The experimental findings indicate that employing suitable features and representations after pre-processing, the accuracy of sentiment classification may be significantly enhanced. The performance of Twitter sentiment classifiers was

examined by Saif et al. [15] to see how different stop word removal techniques affected the polarity classication of tweets. They examined how eliminating stop words affected two supervised sentiment analysis approaches using six different stop word identification methods on six distinct Twitter datasets. The effectiveness of Twitter sentiment analysis methods was negatively impacted by the use of pre-compiled lists of stop words.

Khan et al. 16. used naive bayes for subjective text analysis by computing cumulative aspect probabilities in combination with class labels. The document is assigned to the class with the highest likelihood. To reduce the bias towards majority classes and increase the effectiveness of k-NN, variations were applied. To prevent ambiguity in sentiment polarity, aspect-based sentiment analysis was conducted. By doing tokenization, eliminating stop words, and repeated characters, Mehra et al16 and their team was able to recover raw data from Twitter. A mix of naive bayes and a fuzzy classifier, the SentiStrength tool was used to detect the tweets with negative sentiments, and it produced results with 100% accuracy, recall, and precision. Utilizing linguistic variables to determine the polarity of words in the tweets, fuzzy classifier uses fuzzy functions to calculate sentiment score of tweets with negative sentiments classified as less negative, negative, and more negative before combining scores of each word to obtain final score of a sentence. Here, the polarity was determined in relation to unfavourable words from a built-in vocabulary. The naive bayes classifier's features included POS and N-grams.

Based on the knowledge above we were heaily inspired to work on pre-processing and afterwards our proposed model DLC which is a combination of CNN and LSTM.

3 Featured Models

3.1 LSTM

LSTMs, or Long Short-Term Memory networks, are variations. is a kind of Recurrent Neural Networks (RNNs) that addresses the gradient RNNs' disappearing/exploding bug [4]. In essence, LSTMs are created to identify distant dependencies in texts. Three gates are present in each LSTM unit to regulate which parts are active. of data to remember, disregard, and go on to the next phase. The contextual semantics of each word are stored in LSTMs by the surrounding data and store dependencies over time words. But they only pay attention to one direction of the input. which was previously. Contrarily, Bi-LSTMs emphasise the past and the present. the input's future directions Using this technique, the network can gather more data than before, whereby each token concealed representations from each direction are in alignment, concatenated[4].

3.2 CNN

A deep learning architecture called Convolutional Neural Network (CNN) has one input and a output layer, several neural hidden layers, and a layer. Usually Token sequences are fed into the CNN in NLP tasks. Following that, CNN filters operate as n-grams over continuous representations. After that, these n-grams filters are combined by subsequent network layers, dense layers.

CNN is able to learn the characteristics and tell them apart. CNN operates automatically, requiring no hand-engineered features, saving time and human labour and doing away with the requirement for prior knowledge. Additionally, unlike a multilayer perceptron (MLP), CNNs allow for the reduction of free parameters as well as

the prevention of vanished or exploded gradients during training. In order to increase efficiency and save memory, all the weights in the convolutional layers are shared, which implies that the same filter is applied to all the fields inside a layer [5].

3.3 BERT

BERT: Transformers' Bidirectional Encoder Representations. By utilising a "masked language model" (MLM) pre-training objective that was influenced by the Cloze task, BERT reduces the previously noted unidirectionally constraint.

Pre-training and fine-tuning are the framework for BERT. Pre-training involves training the model using unlabeled data. data spanning various pre-training tasks. The BERT model is initially setup with for adjusting using Using labelled data from the training set, the pre-trained parameters and all other parameters are adjusted. downstream activities Despite being started with the same pre-trained parameters, each downstream task has its own fine-tuned models [6].

3.4 Word Embedding

Mikolov's Word2Vec model has the benefit of being able to capture the syntax and semantic meaning of natural language thanks to its vector form. Additionally, the Word2Vec model is a word vector representation approach that has the ability to group words similarly, i.e., related words share the same vector, in order to get the greatest performance in NLP. The vector space and text body are the inputs and outputs of a neural network that calculates the Word2Vec model architecture. A low-dimensional space vector representing the final word vector captures the semantic definition of the term[7]. Word2Vec architectural models come in two different varieties: Skip-Gram models and Continuous Bag of Words (CBOW) models. In order to examine several word vector representations in unstructured text, the Skip-gram model was developed. While CBOW models forecast current words based on context words, Skip-gram model design seeks to make predictions in the range before or after the current word whose input comes from the current word. We are going to use skip gram architecture for semantic similarity purpose.

4 Methodology

4.1 Data Collection

We took the dataset from the following link: https://github.com/AshwanthRamji/Depression-Sentiment-Analysis-with-Twitter-Data/blob/master/ tweetdata.txt.

This complete dataset is a collection of 3754 tweets.

4.2 Data Pre-Processing

For the data pre-processig we need to consider several things. At first, all the words need to converted to lowercase. Removal of url, extra blank spaces and hastags were applied for our project. Stopwords are words that don't provide any valuable information to the dataset (i.e.pronouns, prepositions, conjunctions etc.) Thus, by eliminating them, the items' space in the training and testing set is greatly reduced. Furthermore, removal of emoticons were applied as well.

After the removal takes place we have applied stemming. Grouping words with similar meanings together is a technique known as stemming. In order to reduce

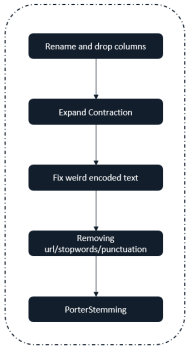

Figure 1: Data Pre-processing

duplication, the suffix must be removed to reveal the word's underlying form. If these various words can be combined into a single term, NLP will be improved in this way.

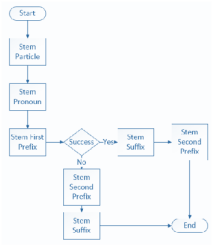

Figure 2: Workflow of Porter Stemming

Application of word embedding to implement Word2Vec was done for this project.The goal of Word2vec is to bring together in vector-space the vectors of words that are related in meaning. In other words, it looks for mathematical similarities. Word characteristics, such as the context of individual words, are dispersed numerically represented as vectors via Word2vec. Figure 1 displays the results of word embedding.

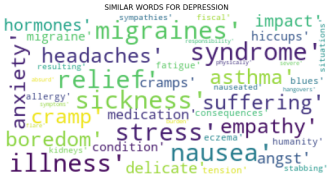

Figure 3: Word2Vec implementation for "Depression"

Pre-training is known to enhance generalisation when the target task's dataset is extremely limited. Previous research on transfer learning has examined the characteristics of this impact, including when finetuning should end and which layers should be tuned [8]. In[9] demonstrate through a series of ablation tests that the advantages of pre-training hold up under large variations in the dataset used for pre-training, including the removal of classes associated with the goal task and its known for the

robustness as well.

In this work, we denote the number of layers as **L**, the hidden size as **H**, and the number of self-attention heads as **A**. We primarily report results on two model sizes:

$$BERT_{BASE} L = 12, H = 768, A = 12 and Total Parameters = 110M$$

$$BERT_{LARGE} L = 24, H = 1024, A = 16 and Total Parameters = 340M$$

.

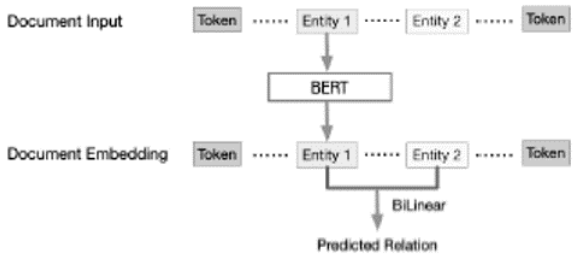

Figure 4: BERT Fine-Tuning

BERT **base** was chosen to have the same model size as OpenAI GPT for comparison purposes. Critically, however, the BERT Transformer uses bidirectional self-attention, while the GPT Transformer uses constrained self-attention where every token can only attend to context to its left[6]. We may simply insert the task-specific inputs and outputs into BERT for fine-tuning, whereupon all the parameters are adjusted from beginning to end. Sentences A and B from the pre-training are comparable to sentence pairs in paraphrasing, hypothesis-premise couples in entailment, question-passage pairings in question answering, and a degenerate text-? pair in text categorisation or sequence tagging at the input. At the output, the [CLS] representation is fed into an output layer for classification tasks like entailment or sentiment analysis, while the token representations are fed into an output layer for tokenlevel tasks like sequence tagging or question answering[6].

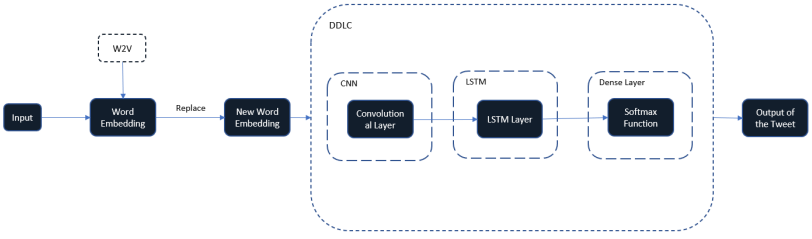

Figure 5: Architecture of DLC

After BERT is pre-trained, we split our embedded dataset into 80/20 ratio for training and testing purpose respectively. Since we dont have enough resources to run the powerful computations, we will use jupyter notebook pro. Fitting and training/testing needs to be done for all the models respectively.

In this experiment, we utilised the LSTM+CNN model, which takes an input and produces a single number that reflects the likelihood that a tweet is depressive. Due to the fact that CNNs are excellent at learning spatial organisation from data by learning some structure from the sequential data and then passing it into a conventional LSTM layer and a common dense model is fed the output of the LSTM layer for prediction. We came up with Hybrid model of DLC. It takes an input and produces a single number that reflects the likelihood that a tweet is depressive. The model receives each sentence as input, replaces it with its embeddings, and then applies a convolutional layer to the newly created embedding vector. By using our hybrid model we will be able to detect if the tweet is actually positive, negative or neutral. Estimated efficiency of 2% can be achieved from our hybrid approach.

5 Results and Discussion

Due to limitation of our machine, we were only able to test out until max length of 256 for our testing purpose. While trying out with the max length of 512 our machine could not handle the load.

As we can see from Figure 6, in terms of BERT, when the sample size if 64 it overfits in every category. While the sample size is 128 and 256 it provides a decent result which does not necessary overfits.

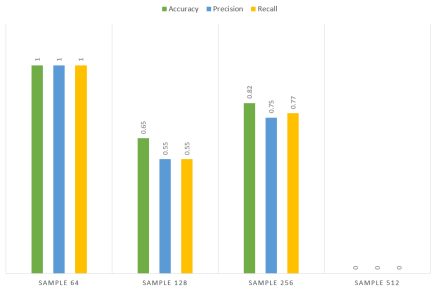

Figure 6: Outcome of BERT

In terms of DLC, when the sample size if 64 it overfits in every category. While the sample size is 128 and 256 it provides a decent result which does not necessary overfits. The comparions and discussion of the scores are given in the tables.

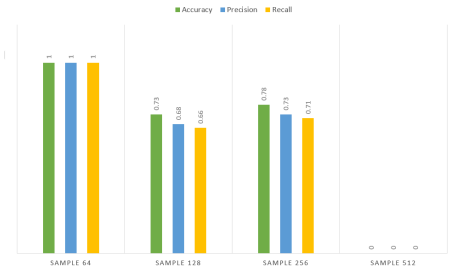

Figure 7: outcome of DLC

From the table 1, we can see that, in terms of sample size 128, our model performed quit well. With 73% accuracy, 68% precision and recall of and recall score of 66% our custom model is preforming quite better.

Table 1: Comparison of Models based on sample size 256

	Sample Size 128		
Model	***Accuracy***	***Precision***	***Recall***
BERT	65%	55%	55%
DLC	73%	68.1%	66%

However if we look at the table 2, the results are quite different. It can be observed that, in terms of sample size 256, our model perform fall a little bit behind from BERT. our model has got 78% accuracy, 73% precision and recall of and recall score of 71%. Whereas, BERT model got 82% accuracy, 75% precision and recall of and recall score of 77%. In terms of accuracy and precision there might not be a huge gap. But in terms of recall, our model has fallen behind.

Table 2: Comparison of Models based on sample size 256

	Sample Size 256		
Model	***Accuracy***	***Precision***	***Recall***
BERT	82%	75%	77%
DLC	78%	73%	71%

Finally, the figure below represents the outcome of our twitter dataset. Sentiment 0 represents neutral and 1 represents positive emotion.

sentiment	reviews
1	gonna take a bath in a while. #SArmy
1	@CodingCreation nothin much sitting outside enojoying the sun you???
1	@bengoldacre Nick Clegg. He's a bit shit; Cameron Junior!
0	@bunsnickers YEAH? :-? EHHH COMPLICATED yung karma karma help! hows school?
1	40 more to go
0	I am off to bed! My poor little guy has been sick all night seems to be feeling better now

Figure 8: outcome of DLC

5.1 Learning Curve

From this project we had quite a lot to learn. At first, initial hurdle was collecting proper dataset which we will be able to use for our experiment. We had to search for labelled data which were time consuming but eventually we found it. Second, we got to learn new methods for implementing pre-processing word2vec. How to customize the model and use it in our own project was good thing to learn. Finally, we were able to grasp the knowledge of using pre-trained model into new model for getting better accuracy. This was quite new and we got to learn a lot.

6 Conclusion

Overall, in this project we demonstrated how twitter dataset can work as an input and from there we can generate if the outcome is positive, negative or neutral. The experiment was conducted based on three classifiers and one hybrid classifier(DLC). This hybrid classifier Later on, results were displayed based on accuracy, precision and recall of each classifiers respectively. According to some papers, sentiment detection can be used later to detect empathy as well which will be helpful for online mental support. For our future work, we will try to research more on how to detect empathy and the usage of it.

7 References

1. Skaik, R., Inkpen, D. (2020). Using social media for mental health surveillance: a review. *ACM Computing Surveys (CSUR)*, 53(6), 1-31.

2. Marcus, M., Yasamy, M. T., van Ommeren, M. V., Chisholm, D., Saxena, S. (2012). Depression: A global public health concern..

3. Saif, H., He, Y., Fernandez, M., Alani, H. (2016). Contextual semantics for sentiment analysis of Twitter. *Information Processing Management*, 52(1), 5-19.

4. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

5. ALRashdi, R., O'Keefe, S. (2019). Deep learning and word embeddings for tweet classification for crisis response. *arXiv preprint arXiv:1903.11024*.

6. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

7. Nawangsari, R. P., Kusumaningrum, R., Wibowo, A. (2019). Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study. *Procedia Computer Science*, 157, 360-366.

8. Hendrycks, D., Lee, K., Mazeika, M. (2019, May). Using pre-training can improve model robustness and uncertainty. *In International Conference on Machine Learning* (pp. 2712-2721). PMLR.

9. Yosinski, J., Clune, J., Bengio, Y., Lipson, H. (2014). How transferable are features in deep neural networks?. *Advances in neural information processing systems*, 27.

10. Kouloumpis, E., Wilson, T., Moore, J. (2011). Twitter sentiment analysis: The good the bad and the omg!. *In Proceedings of the international AAAI conference on web and social media* (Vol. 5, No. 1, pp. 538-541).

11. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R. J. (2011, June). Sentiment analysis of twitter data. *In Proceedings of the workshop on language in social media* (LSM 2011) (pp. 30-38).

12. Carchiolo, V., Longheu, A., Malgeri, M. (2015, September). Using twitter data and sentiment analysis to study diseases dynamics. *In International conference on information technology in bio-and medical informatics* (pp. 16-24). Springer, Cham.

13. Liu, S., Cheng, X., Li, F., Li, F. (2014). TASC: topic-adaptive sentiment classification on dynamic tweets. *IEEE Transactions on Knowledge and Data Engineering*, 27(6), 1696-1709.

14. Haddi, E., Liu, X., Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia computer science*, 17, 26-32.

15. Saif, H., Fernández, M., He, Y., Alani, H. (2014). On stopwords, filtering and data sparsity for sentiment analysis of twitter.

16. Khan, M. T., Khalid, S. (2016). Sentiment analysis for health care. In Big Data: Concepts, Methodologies, Tools, and Applications (pp. 676-689). *IGI Global*.

17. Mehra, R., Bedi, M. K., Singh, G., Arora, R., Bala, T., Saxena, S. (2017, July). Sentimental analysis using fuzzy and naive bayes. *In 2017 International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 945-950). IEEE.