Detecting Depression Tendencies in Twitter Users

Abhiraj Smit 112687696

Anmol Shukla 112551470

Gargi Sawhney 112684130

Abstract

Depression is a serious mental illness that requires understanding and care, otherwise it may be lead to life-threatening consequences such as self-harm and suicide. Depression cures are widely available, and with professional help, depression can be treated. If depression tendencies in a person could be identified at an early stage, necessary help could be given so that the situation does not become worse. Language is a major component of mental health assessment and treatment. Today, many people express their thoughts and emotions on social media, depression being one of them. Thus, social media language can serve as a useful lens for mental health analysis.

In this work, we seek to combine the power of big data and deep NLP models to predict whether the tweets express a depression tendency or not. We consider a dataset scrapped from twitter and try to explore if the Tweets can be classified as depressed or not. This works present a comprehensive study of different state of the art models to classify a tweet and to evaluate the shortcomings of different model in different situations.

1 Introduction

Depression is a mental illness that many people conceal and are unaware that it is affecting them. A lot of times, you can analyze a depressed person by their social activity. Twitter contains a lot of information which can be extracted to analyze and identify the depression tendencies for a person. We chose Twitter data for this analysis because Twitter is a very popular platform for sharing information in real-time. The brevity of the Tweets allow users to quickly share events as well as their emotions with their followers. Also, a large number of user profiles are public unlike Facebook or Snapchat. Depression being a subjective topic and with a very thin line segregating it from sadness, human expertise in this domain will yield the best results. Having said that, it will take an inhumane effort and resources to manually go through all the tweets and to find out the depression tendency of a user. In this project, we aim at building models for predicting depression using tweets of a user and then extending this analysis to identify the depression tendencies for any particular user.

In the study "CLPsych 2015 Shared Task: Depression and PTSD on Twitter" by Coppersmith et al., several approaches were proposed to identify whether a Twitter user suffered from depression or not. Resnik et al., 2015 used topic modelling for this classification task, while Preotiuc-Pietro et al., 2015 examined a wide variety of methods for inferring topics automatically, combined with binary unigram vectors (i.e., "did this user ever use this word?"), and scored using straightforward regression methods. Pedersen, 2015 took a rule-based approach to these tasks, and as such provides a point to examine how powerful simple, raw language features are in this context. In another research paper, they used keywords like depression, anxiety, sad etc. to get the tweets and positively labelled those tweets.
While the approaches used in CLPsych task had a good dataset that included Tweets of users that were clinically classified as depressed, the approaches used did not give a very good accuracy and they used hand-engineered features to identify depressive Tweets. Also, due to privacy concerns, they did not release the dataset. They also did not evaluate the contemporary state-of-the-art NLP techniques and relied on raw language features. The data extraction approach mentioned in the research paper "A content analysis of depression-

related Tweets" by Patricia A. Cavazos-Rehg et al. used specific depression-related keywords for building the dataset. This approach can lead to bias in the data and can mark depression prevention or cure tweets with positive labels.

In this project, we aim to tackle the classical NLP task of classification. Specifically, we want to classify Tweets as either "suggestive of depression" or "not suggestive of depression". A natural extension of our project is that given a Twitter user handle, determine the depression tendencies of the user. A summary of the work is as explained below:

1. We scraped tweets from particular twitter which shows depression tendencies and label the tweets accordingly. We also manual annotated around 4500 tweets to reduce the number of false positives in our database.

2. We ran logistic regression as our baseline model with both TF-IDF and count vectorizer with ngrams of size 4.

3. We analyzed the performance for different deep learning architectures: LSTM + CNN, BiLISTM + CNN, BiLSTM with Attention + CNN. We also found a relevant set of examples corresponding to which one model fails to predict correctly while other model does.

4. We evaluated our model performance with pretrained embeddings against the learned embeddings. Also, compared model performance for different embedding dimensions as well as when embeddings are in trainable state and when they are not.

5. We ran the state of the art model RoBERTa and compared its performance with the multiple neural network architectures. We will discuss in detail about these models in section (4) of this report.

6. We devised a depression tendency score given several tweets of a user using our model.

For evaluation, we found a set of examples for which one model fails to classify correctly when compared to other model.The differences between the performance of the models for our dataset has been discussed in section (6) of the report. The main outcomes or takeaways of this project can be summarized as follows:

- Implementation of an end-to-end system to predict the depression tendency of twitter users.
- It is important to reduce the bias in a dataset so that the classifier doesn't classify based on a fixed set of depression related keywords
- Stop Word removal during preprocessing leads to improved performance.
- Stemming words reduces the model performance.
- Trainable embeddings didn't prove to improve the accuracy of our models.
- Performance wise analysis helped us to understand the behaviour of different neural network architectures
- Impact of using state-of-the-art model RoBERTa.

2 Methodology

An essential part of this project is the systematic acquisition and processing of data. We tried different ways to acquire data that will be most suitable for the task. Since the data is sensitive, we tried to get the clinically approved data but the procedure to obtain is time taking and involves a bunch of steps not feasible in the time frame.

2.1 Dataset Acquistion Process

We needed to have a balanced dataset with good proportion of positive and negatively labelled data. We need Tweets that are suggestive of depression along with Tweets that are not. But, it is difficult to find Tweets that are linked with depression mainly due to the fact that any Tweet containing keywords such as depression, sad might not be directly linked to depression. A number of Tweets containing such keywords are actually aimed at helping people suffering from depression. Therefore, to find Tweets indicative of depression, we tried to find public accounts that showed signs of depression and generally Tweets content indicative of depression. We were able to find around 20k such Tweets using Twint an Selenium. We would also need to add positive/neutral

Tweets that are not suggestive of Depression. This is an easier task as we get many such Tweets from different public accounts on Twitter such as News channels, celebrity accounts, accounts posting motivating/positive quotes etc. We also got positive/neutral tweets from Sentiment 140 dataset from Kaggle.

2.2 Dataset Cleaning and Preprocessing

We took the following steps to clean our data:

- Removal of URLs, Images and Videos present in the tweets.
- The emoticons,smileys and hashtags from tweets are deleted so that we only have words whose embeddings are present.
- Tweets which are really small in length i.e having smaller than a threshold number of words are removed.
- Performance measure after performing stemming(converting winning to win) and stop (words like the, a, who) words. Our dataset was more accurate after removal of stop words and by avoiding stemming.

We removed the rows which contains nan or null fields, after performing above pre-processing steps. We also manually annotated around 4500 tweets with positive and negatively labelled to reduce the number of false positive and false negatives in our dataset. Now we have a fresh, better, larger, and a more comprehensive data-set, we have worked on it more meticulously to unearth interesting aspects and analysis from our models.

2.3 Bias in Data

In the research paper [7], they tend to focus on the keywords which are usually associated with depression like anxiety, sadness etc. The dataset for such cases will be biased towards these words for positive labelled data.
In order to check how much our dataset is biased towards these words, we made a word cloud on the data which we have marked as positive. We built a naive classifier to identify depressed tweets based on the dataset built over these keywords. The naive classifier marks the sentence as positive if it contains the top 10 words from the word cloud. We got an accuracy of *0.756* for this case.
To rectify such bias, we added non-depressed tweets containing these keywords which are labelled negative e.g. tweets from mental health awareness pages. We ran our naive classifier on this set, to get an accuracy of *0.649*.
We have reduce the bias in our data and now can run our baseline model to fetch results.

3 Proposed Approach

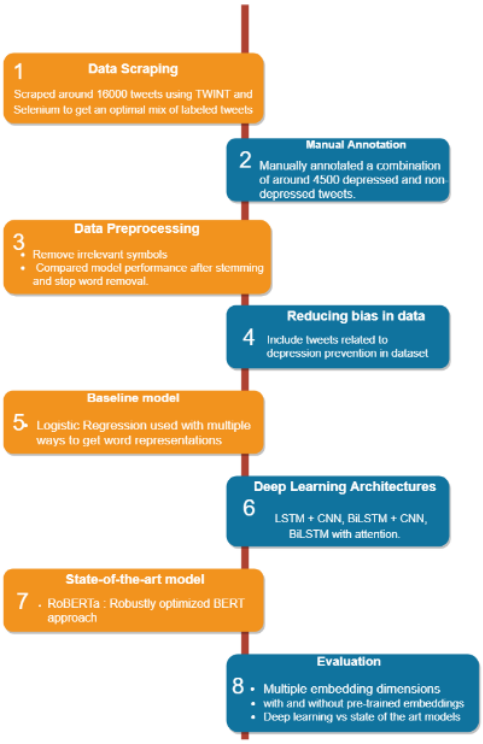

Figure 1: Proposed Approach

3.1 Baseline

For our baseline model, we will be using Logistic Regression classifier. Before running the model, we need to transform our sentences i.e. Tweets into a representation that can be used as an input for our classifier. We will use 2 approaches to find our word representations:

1. **TF-IDF** (term frequency–inverse document frequency): It vectorized word by taking into account the frequency of the word in the document and the frequency of that words in other documents.
2. **Count Vectorizer**: This approach is called the bag of words model, which means we are representing each sentence or document as a collection of discrete words and ignore grammar or the order in which words appear in a

sentence. This will result in word vectors that will be as long as the size of vocabulary.

TF-IDF gave the better performance so we chose it as our baseline model. To improve our baseline further, we choose the ngrams with size 4.

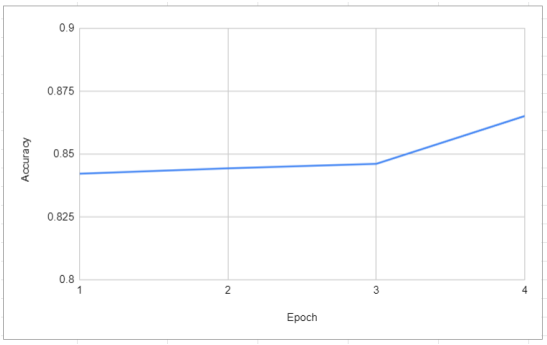

Figure 2: Accuracies of (1)Cvec (2)TF-IDF (3)Cvec-ngrams4 (4)TF-IDF-ngrams4

3.2 Issues

Our baseline model accuracy can be improved and so we will now run our model on various neural networks models.

4 Models

4.1 LSTM + CNN

Current research has compared the performance of RNN and CNN based models[2]. The idea is to explore a combination of the two: LSTM-CNN.

Implementation Details: For this model, we use a dropout probability of 0.2 and apply a one-dimensional convolutional operation with 64 filters, kernel size of 5, and activation function relu. In order to extract global abstract information, we apply a global max pooling layer on the feature map through which we obtain an abstract feature representation of length 64. These features as obtained from the convolutional layer are passed sequentially through the LSTM layer of 100 units. We then use a dense layer to reduce the output to 1 and use the sigmoid activation function.

4.2 BiLSTM

Unidirectional LSTM preserves information of the past because it only has knowledge about previous inputs while forming a representation. We, therefore, extended the idea presented above by using a bidirectional LSTM instead of a unidirectional LSTM. Now, at any point in time, we will have information from both past and future.

Implementation Details: We obtain the embeddings corresponding to the tweets which are passed to bidirectional LSTM layers with 100 units. BiLSTM captures the information of the input sequences in the forward and backward direction. We then concatenate these two representations to use the last output in the sequence.

4.3 BiLSTM with Attention

Different words have different level of importance in a sentence and this relative importance of words contributes towards making a decision on the overall meaning of the sequence. This was imperative to understand for our problem statement of classifying the sentence as depressed or non-depressed. We, therefore, decided to extend our BiLSTM model by incorporating an attention based mechanism to focus on important words.

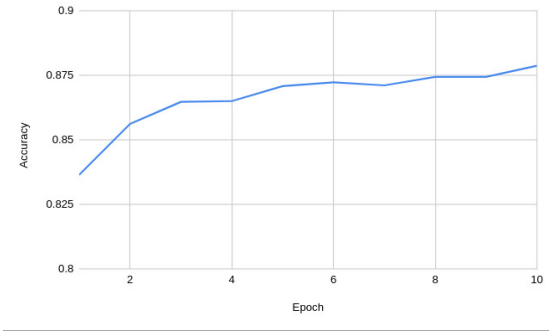

Figure 3: Accuracy curve for BiLSTM with attention

Architecture:

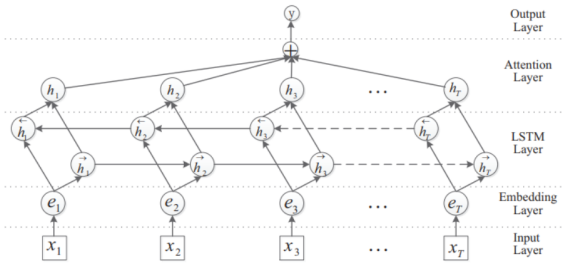

Figure 4: BiLSTM with attention reference Zhou et al

Implementation Details: The implementation done above for Bi-LSTM is extended to include a context-aware attention mechanism. It is the

weighted summation of the hidden state output from the forward and backward layer. Let H be a matrix consisting of output vectors $[h_1, h_2, ..., h_T]$ that the LSTM layer produced, where T is the sentence length. The representation r of the sentence is formed by a weighted sum of these output vectors:

$$M = tanh(H) \quad (1)$$

$$\alpha = softmax(w^T M) \quad (2)$$

$$r = H\alpha^T \quad (3)$$

$$h^* = tanh(r) \quad (4)$$

We use this representaion as our feature vector for classification.

5 State of the art: RoBERTa

RoBERTa is a Robustly optimized BERT approach, an improved recipe for training BERT models that matches and exceeds the performance of all of the post-BERT methods. We have used SimpleTransformers library which is a wrapper on the Transformers library by HuggingFace. RoBERTa uses *5* English Language corpora of different sizes as data for pretraining - BookCorpus, Wikipedia, CC-News, OpenWebText and Stories. For this model, we have used a learning rate of 4e-5, batch size of 8, warmup ratio of 0.06, and 5 epochs.

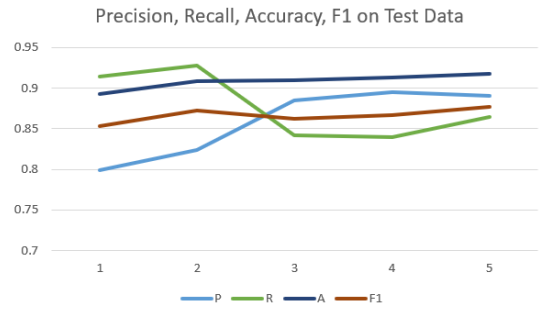

Figure 5: Metrics for RoBERTa

6 Evaluation

6.1 Evaluation Measures

For each of the models we have calculated True Positive(TP), False Positive(FP), False Negative(FN) and True Negative(TN).
We have used the following parameters for evaluation,

$$Precision(P) = \frac{TP}{TP + FP}$$

$$Recall(R) = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * P * R}{P + R}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

To calculate the depression tendency for a user in a certain window of time, we will test the user tweets by our best model **RoBERTa** and check the number of tweets depicting depression tendencies. To get the depression tendency score, we will divide the number of tweets depicting depression tendency by his total number of tweets in that time frame.
Let number of tweets done by a user in a fixed time frame be d.
Number of tweets depicting depression tendency in that time frame be t.

$$DepressionTendency = \frac{d}{t}$$

6.2 Results

Model	P	R	F1	Acc
LR + TF-IDF	0.714	0.885	0.791	0.846
LR + GLoVe	0.752	0.734	0.743	0.833
LSTM + CNN + GLoVe	0.773	0.854	0.812	0.870
BiLSTM + GLoVe	0.797	0.835	0.815	0.876
BiLSTM + Attention + GLoVe	0.820	0.807	0.862	0.878
Roberta	0.882	0.864	0.873	0.925

Table 1: Metrics for different models and configurations.

As shown in the results above, RoBERTA gave us a better accuracy than the other models. We also found the performance of BiLSTM with attention layer to be in line with our expectations - it performed better than vanilla BiLSTM but as the Tweets are relatively short, it did not give a significant improvement in accuracy. The performance of LSTM and BiLSTM model was close but they did significantly better than Logistic Regression. One novel approach we tried with Logistic Regression was to use GLoVe embeddings for

words and summing up and normalizing the word embeddings to get sentence level representation. This approach was inspired from the DAN architecture taught in the class and it did not perform too bad.

6.3 Analysis

We have shown a table on which we ran for different dimensions of Glove embeddings which are trained on Twitter Corpora. Our best result turns out to be for the 100 dimension Glove embeddings. For the 100 dimension embedding, we also ran with trainable set to True and observed slightly reduced accuracy.

Embed -dings	Dim	Train -able	F1	Accuracy
GLoVe	25	False	0.833	0.855
GLoVe	50	False	0.834	0.857
GLoVe	100	False	0.862	0.879
GLoVe	100	True	0.845	0.864

Table 2: A comparison of different GLoVe embeddings.

Impact of using pre-trained word embeddings
We analyzed the performance of our models using pre-trained word embeddings. Initially, we used Stanford NLP's GLoVe embeddings trained on Wikipedia corpora, but, we observed that the embeddings pretrained on Twitter corpora gave us better accuracy and results when compared to the Wikipedia based word embeddings. The input tweet is embedded into a Twitter data trained embedding matrix which is then fed into the neural network. The accuracy as shown in Table 1 increased when using pre-trained word embeddings instead of learning these embeddings.

In the table[3] (Comparison of prediction made by different models), we have depicted our best 3 models performance against different sentences. We have seen the sentences which have failed by LSTM but that were predicted correctly with BiLSTM + Attention. Similarly, we have seen the example sentences which are failed in BiLSTM + Attention but gave correct result with the RoBERTa.

The reasons why these sentences have failed:

1. Sentence 1 (*Actual Label: Depression*) is really long and so BiLSTM with attention was able to capture the long range dependencies in the sentences.

2. Sentence 2 (*Actual Label: Non Depression*) has the key word depression and first part of sentence before punctuation is a depressed sentence but the whole meaning of sentence doesn't represents depression. BiLstm is able to capture the long range dependencies between the first part and the second part.

3. Sentence 3 (*Actual Label: Depression*) as a whole only represents depression and so only BiLSTM is able to capture this but LSTM fails to do so.

4. Sentence 4 (*Actual Label: Non Depression*) is similar to Sentence 2, and so it is predicted as depression by LSTM but it is actually non depression.

5. Sentence 5 (*Actual Label: Non Depression*). Though the sentence has word Depression and it also has the word non existent, our BiLSTM model is not able to detect double negative and so predicted it wrong.

6. Sentence 6 (*Actual Label: Non Depression*) is a definition of a word that describes the state of depression. Here the user himself is not feeling the depression and the model might not be able to pick up the reference that user is not talking about his own emotions.

6.4 Code

Our code can be found here - Google Drive link

7 Conclusions and Takeaways

After comparative performance evaluation, we inferred that the stop word removal helps in increasing the accuracy of our model. Moreover, stemming of words during preprocessing reduces model performance.
It was important to ensure that model doesn't learn to classify because of the bias in the dataset. The dataset collection methodology used by us reduces bias in our dataset and ensures that the model doesn't learn to classify just based on a fixed set of most frequent depression related words.
Our baseline model gave us an accuracy of 0.846

Original Tweet (Unprocessed)	LSTM	BiLSTM +Attn	RoBERTa
I've disappointed a lot of people this year but believe me when i say there's no one I've disappointed more than myself			
Depression affects many older adults receiving home care, but often is not properly treated			
Even if I say it will be alright still I'm here you say you want to end your life Now and again we try to just stay alive			
You're Already Whole: Some Thoughts on How We Talk About Anxiety and Depression			
my depression is basically nonexistent these days so am i in the clear to finally say i beat it and i f**k made it??? cause i feel like i finally reached that point			
Mulligrubs: A state of depression or low spirits #UnusualWords			

Table 3: A comparison of predictions made by different models.

%. This accuracy was low when compared to the same baseline implementations of other researches because of the explicit noise introduced by us to remove the bias from our dataset. We conducted several experiments and compared performance across different models. The state-of-the-art model RoBERTa gave us the best accuracy of 92.5% followed by BiLSTM-CNN with attention. Performance wise analysis of the different neural network architectures and finding a set of examples at which it fails gave us a better understanding of the behaviour of the different models analyzed.

8 Further Improvements

1. The performance of models can further be improved by using a larger manually annotated dataset which is labelled by people having knowledge relevant to this domain. Having a dataset with all examples correctly labelled would give more confidence in the predictions done by model.

2. We can experiment further on our model development and see the impact on our model performance. One interesting model to explore would be the use of stacked LSTMs coupled with CNN.

References

1. Deep Learning for Depression Detection of Twitter Users
2. Sentiment analysis on reviews: Feature Extraction and Logistic Regression
3. Deep Learning for Depression Detection of Twitter Users
4. CLPsych 2015 Shared Task: Depression and PTSD on Twitter
5. Sentiment 140 Kaggle Dataset
6. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification
7. A content analysis of depression-related Tweets
8. Detecting Depressing Through Tweets
9. Identifying Depression on Social Media