Depression Identification

Introduction

Major depressive disorder or depression is a mental illness that negatively impacts an individual's mood and activities leads to a persistent feeling of sadness and loss of interest. It is estimated that depression accounts for 4.3% of the global burden of disease and 1 million suicides every year. Despite these detrimental effects, the symptoms of depression can only be diagnosed after two weeks. Therefore, a method to detect early depression is a vital improvement in addressing this mental disease.

In recent years, social media platforms such as Facebook, Twitter and Reddit have become an important part in mental health support. It is claimed that social media platforms have been increasingly used by individuals to connect with others, share experiences and feelings. Analyzing the sentiment in comments and posts on such platforms would provide insights about an individual's feeling and health status, which helps to identify mental illness including depression.

There have been many studies that attempt to identify the depression using social media platforms such as "Utilizing Neural Networks and Linguistic Metadata for Early Detection of Depression Indications in Text Sequences" (Marcel, Sven & Christoph, 2018) and "Identifying Depression on Social Media" (Kali, 2019). While the paper of Marcel, Sven & Christoph discusses the efficiency of different word embedding techniques such as word2vec, GloVe and fastText, Kali focuses on the comparison of different models. Both studies show the outstanding performance of convolutional neural network (CNN) in classifying depressed subjects.

Objective and Scope

This project aims to identify depressed subjects utilizing comments scrapped from Reddit. With respect to the excellence of CNN in previous research, we will build a CNN architecture to predict comments labeled as depression. We also expect to see how a more popular neural network model in text classification will perform comparing to the CNN. There will be two neural network architectures: the base-line model is CNN, and the second model is the recurrent neural network (RNN) which is well-known for text classification and sequence classification tasks. Moreover, word embeddings will be implemented in our models. We will vectorize the comments and create an embedding layer using the Tokenizer class in the Keras API, and pre-train a Word2Vec model.

Dataset

The data was collected following the reasoning of JT Wolohan (2). The data was scraped from two subreddits: /r/depression and /r/AskReddit using Python Reddit API Wrapper (PRAW). Comments in /r/depression are labeled as depression and comments in /r/AskReddit are labeled as non-depression. For future work, more data would be collected to make the dataset more diverse and representative.

Research Methodology

1. Data Preprocessing

The first step is to preprocess the data. The necessary features such as titles, contents and the labels are extracted from the raw data. Any missing data due to deleted contents are eliminated. The remaining comments are converted into lowercases letters. Irrelevant texts such as subreddits, warnings, html tags, numbers and extra punctuations are removed. Depression comments are labeled as 1 and non-depression comments are labeled as 0.

The preprocessed data contains 5,474 comments in total: 2,719 comments labeled as depression and 2,755 comments labeled as non-depression, which makes the dataset extremely balance for analyzing and modeling. The dataset is divided into training set and testing set with an 70% - 30% ratio. From the training data, 30% of the comments is split for the validation.

2. Word Embedding

In this project, the keras embedding layer and the Word2Vec model are used for word embeddings. At first, we fit the Tokenizer function from the Keras API on the comments to convert the them into sequences of word indices. These sequences are padded to have the same length. Secondly, the Word2Vec model from genism package is utilized to train the comments and produce a pre-trained word embedding matrix.

3. Neural network architecture

In total, there will be four models: CNN with embedding layer, CNN with pre-trained Word2Vec, RNN with embedding layer, and RNN with pre-trained Word2Vec.

The CNN architecture contains an embedding layer as the input, a Conv1D layer, a GlobalAveragePooling1D layer (an alternative for the Flatten – Fully Connected – Dropout paradigm), a hidden fully connected layer (which is used to address the non-linearity from high level features), and a Dense layer as the output. The dimensionality of the output space is 1 and the activation function is 'sigmoid' since the expected output is binary (0 or 1).

The RNN architecture contains an embedding layer, a LSTM layer, a Dropout layer (which is used to reduce overfitting), and the output Dense layer. To implement the pre-trained word embedding model, the Word2Vec embedding matrix is set as the weights of the embedding layer.

4. Model evaluation

To evaluate the models, the popular metrics in classification such as accuracy, precision, recall and F-1 score are applied. Since we concern about the accuracy of classification, accuracy is selected as the primary evaluation metric. The ROC curve and AUC score are used to estimate the discrimination capacity of the models.

Result and Analysis

Table 1

Evaluation Metrics

Model	Accuracy	Precision	Recall	F-1 Score	Loss
CNN – Tokenizer	0.9361	0.9330	0.9387	0.9359	0.1907
RNN – Tokenizer	0.9014	0.8617	0.9547	0.9058	0.2694
CNN – Word2Vec	0.8801	0.8724	0.8885	0.8804	0.3127
RNN – Word2Vec	0.8935	0.8697	0.9240	0.8960	0.3144

Overall, all the models have very good performance in classifying detection. The accuracies when applying the models to the testing data range from 88% to 93% which indicates their excellence predictive powers. The AUC scores are all above 0.9, which suggests that our models have an outstanding class separation capacity (see Appendix A). The architecture with the best performance is the CNN with keras embedding layer. The accuracy is 93.61% and the test loss is 0.19. The model that has worst performance is the CNN with pre-trained Word2Vec. Its accuracy is 88.01% and the test loss is 0.31. The two RNN models show small difference in their performances.

According to Table 1, using a pre-trained Word2Vec seems not be as good as using the keras embedding layer. The reason would be the characteristics of our dataset. The majority of comments and posts in social platforms such as Facebook, Twitter and Reddit are often written in informal ways and do not follow the correct grammar structures. There are many sentences containing short informal texts and symbols to express feelings. Therefore, the Word2Vec seems not to be an optimal option to train our dataset since this model prefers well-structured and formal texts from platforms such as Wikipedia and electronic articles. However, there is an advantage of using Word2Vec is that the neural network models are likely to have no overfitting issue when increasing the number of epochs (see Appendix B).

Another interesting point to discuss is the comparison of CNN and RNN. At first, we will briefly explain how these two models work to detect depression. A CNN model learns to recognize patterns and special expressions from the texts and use them as the filter or feature to classify sentences and comments. In contrast, RNN is designed to recognize patterns across time. The model extracts the sequential information and uses a long-range semantic dependency for classification. In other words, CNN is good at classification using specific, local features, phrases and words, while RNN is better at learning the comprehension of global/long-range semantics to solve the tasks. Since our dataset comes from a social media platform, many comments are informally written and contain no sequential structure. Reddit users prefer to use short texts to describe feelings rather than writing a long story to express their conditions. This results in a better performance of CNN model because the classification is primarily based on the extraction of sentiment terms. RNN would be better when the data contains textual documents and requires the detection of sequential information.

Challenge

The most difficult issue in analyzing data from social media platforms is the use of special characters and emojis. Since our dataset is selected carefully, the majority of comments are written as texts. However, to extend the project to a wider application, we need to address this challenge. This task can be tackled by creating an encoder or an embedding model to convert the special characters, emojis and symbols into numbers or vectors.

Future Work

There are many studies that can be done to improve this project in the future. The first improvement is to address data using special characters, symbols and emojis during the text preprocessing. More data which contains both textual writings and informal comments from a variety of social media sources would be collected to make our dataset more diverse and representative. To deal with such dataset that requires both sequential information and local features extraction, we can ensemble the CNN and RNN architectures to improve the classification accuracy. A neural network model containing convolutional layers and LSTM layers would be constructed to detect depression. Different word embedding models such as Word2Vec, GloVe, fastText can be implemented to the pre-trained embedding layer to figure out the most optimal word embedding technique.

Conclusion

In this project, our group presents a method to address early detection of depression using Reddit comments. Two neural network models including CNN and RNN are built to detect subjects with depression. For more insights, we use the keras layer embedding and the pre-trained Word2Vec model to create the input embedding layer of the networks. The results show that keras embedding layer has better performance than the Word2Vec. The reason may be the inefficiency of Word2Vec in accounting informal and ungrammatical data.

According to the model evaluation, CNN model with keras embedding layer has the best performance. The reason that CNN has better classification accuracy than RNN would be that our dataset does not have many textual comments, so the extraction of sequential information is not necessary. RNN model is expected to have more outstanding performance when dealing with formal and academic writings such as personal experience stories and mental health reports.

References

1. Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: CoRR abs/1810.04805. 2018.

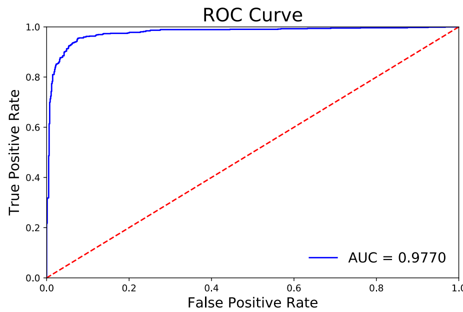
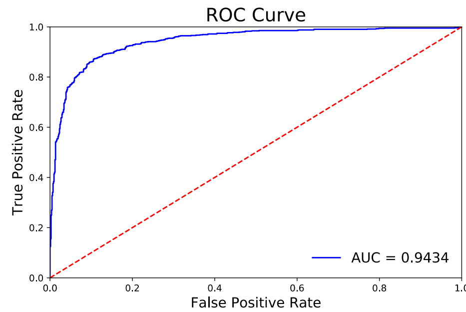
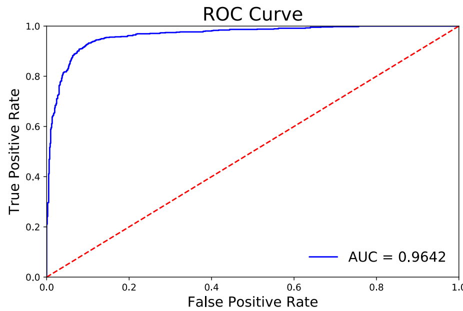
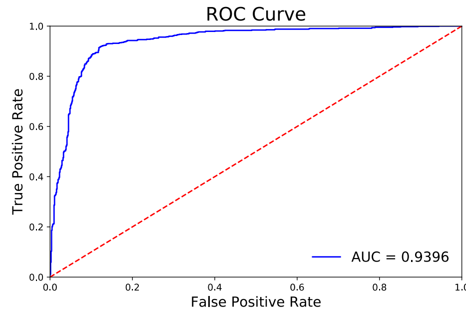
2. JT Wolohan. “Detecting Linguistic Traces of Depression in Topic-Restricted Text: Attending to Self-Stigmatized Depression with NLP”. In: 2018.

3. Kali Cornn. "Identifying Depression on Social Media". Department of Statistics, Stanford University. 2019.

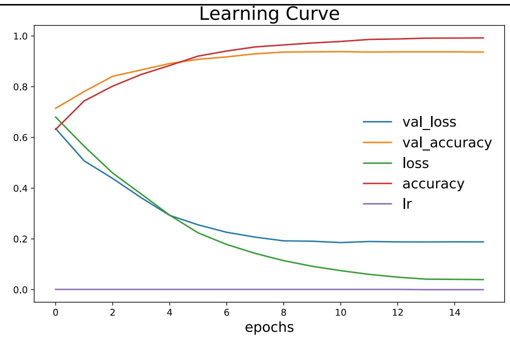
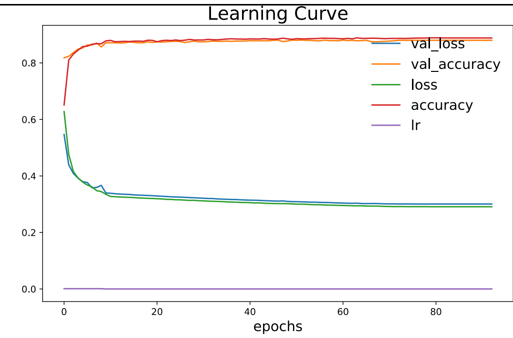
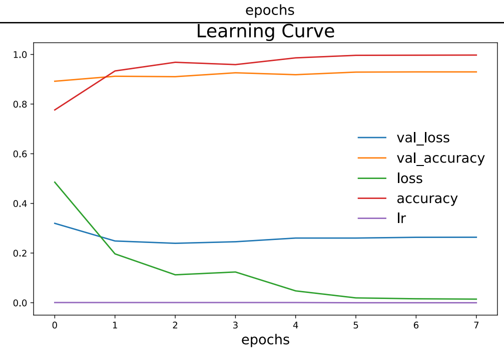
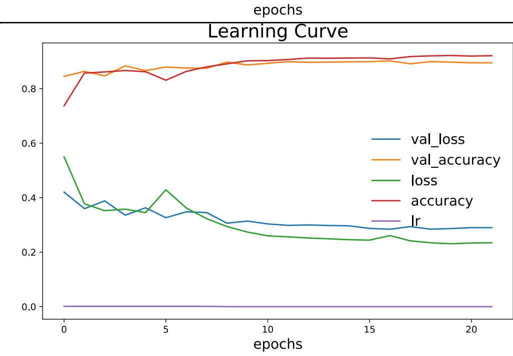
4. Marcel Trotzek, Sven Koitka, and Christoph M. Friedrich. “Utilizing Neural Networks and Linguistic Metadata for Early Detection of Depression Indications in Text Sequences”. In: CoRR abs/1804.07000. 2018.

5. Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. “Tweet2Vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder”. In: SIGIR. 2016.

Appendix A: ROC Curve and AUC Score

<table>
<tr><th></th><th>Tokenizer, Keras API</th><th>Pre-trained Word2Vec</th></tr>
<tr><td>CNN</td><td></td><td></td></tr>
<tr><td>RNN</td><td></td><td></td></tr>
</table>

Appendix B: Learning Curve

<table>
<tr><th></th><th>Tokenizer, Keras API</th><th>Pre-trained Word2Vec</th></tr>
<tr><td>CNN</td><td></td><td></td></tr>
<tr><td>RNN</td><td></td><td></td></tr>
</table>