Depression Detection Using Speech & Text Analysis

Aksheit Saxena
aksheits@iisc.ac.in

Anish Aralikatti
anishar@iisc.ac.in

Kavin
kavina@iisc.ac.in

Anil Kumar
anilkumar123@iisc.ac.in

Manasa MS
manasam@iisc.ac.in

Pinnamaraju Raviraj Sitaram
ravirajp@iisc.ac.in

Abstract: *Early detection and diagnosis of depression can significantly improve treatment outcomes, but it can be challenging to identify depression through traditional methods such as self-reported surveys or clinical interviews. In this project, we propose to use machine learning techniques to detect depression from speech and text analysis. The main objective of the project is to develop a machine learning model that can accurately predict the presence of symptoms of depression in an individual based on their speech patterns and written text.*

Keywords: *CNN, LSTM, Depression*

I. Introduction

Depression is now prevalent among people and depression is a concern because people who are going through depression are unaware of their condition. Due to this there is a delay in diagnosis which leads to a delay in treatment. So, it would be feasible if people can periodically examine themselves from their homes. There are some initiatives where a bot will ask questions and record the answers. Idea is to create models to evaluate the person and caution him/her about their state. This is to explore state of the art models and understand how it behaves on different hyper parameters.

Machine learning and natural language processing (NLP) techniques have shown promise in detecting depression from text data, such as social media posts, online forums, and clinical notes. In this project, we aim to develop a deep learning model to detect depression from multi modal data, using a dataset of clinical interviews from the DIAC-WOZ dataset. Specifically, we will explore the use of a neural network (CNN, LSTM) to classify the interviews as either positive for depression or negative for depression. The goal of this project is to demonstrate the potential of machine learning techniques for detecting depression from data, and to provide insights into the most important features and factors that contribute to accurate classification. The results of this project could have significant implications for the early detection and treatment of depression and could help to improve the quality of life for individuals affected by this disorder.

II. Literature Review

LSTM is a type of RNN with higher memory power to remember the outputs of each node for a more extended period to produce the outcome for the next node efficiently. LSTM networks combat the RNN's vanishing gradients or long-term dependence issue. Gradient vanishing refers to the loss of information in a neural network as connections recur over a longer period. In simple words, LSTM tackles gradient vanishing by ignoring useless data/information in the network.

III. Data set

A. DIAC-WOZ Dataset

We are using the DIAC-WOZ dataset [1][2] which was compiled by the University of Southern California. This is a part of the larger Distress Analysis Interview Corpus (DIAC), which contains clinical interviews that are designed to support the diagnosis of conditions such as depression, anxiety, PTSD, etc.

B. Modalities

The DIAC-WOZ dataset contains audio and video recordings of clinical interviews conducted. These WoZ interviews were conducted by a virtual assistant called Ellie. This virtual interviewer is controlled by a human interviewer sitting in another room. These interviews have been transcribed for capturing a variety of verbal and non-verbal features. Each participant's session is presented as a zip file containing a transcription of interaction, audio file and files containing facial features extracted from the recorded interview video.

C. Dataset Size

The dataset contains 189 sessions of interactions of participants out of which 130 belong to non-depressed class and 59 belong to depressed class.

It also contains separate files called:

- train_split_Depression_AVEC2017.csv: Contains participant ID and details of each interview participant for official train split of the data.
- dev_split_Depression_AVEC2017.csv: Contains participant ID and details of each interview participant for official development split of the data.
- test_split_Depression_AVEC2017.csv: Contains participant ID and details of each interview participant for official test split of the data.

IV. Data Preprocessing

For the DIAC-WOZ dataset, we first accessed and downloaded the zipped data files of each interviewed participant from the DIAC-WOZ database. We unzipped the folders for each participant which contained the files of the interview transcript, audio feature files and video feature files. The transcript file and audio feature files are in csv format. Once the folders were unzipped, we created separate folders for train, development and test sets according to the contents of the files train_split_Depression_AVEC2017.csv, dev_split_Depression_AVEC2017.csv and test_split_Depression_AVEC2017.csv.

V. Implementation

A. CNN for Text Processing

Convolution Neural Network is a type of deep neural network that is commonly used for image classification and text analysis tasks such as sentiment analysis, depression detection, etc. This is done by using word embeddings as input. The reason to use CNN is because they are translation invariant and help in identifying patterns of words irrespective of their position in the sentence.

Word embeddings are vector representations of words that capture the semantic meaning of words based on their context. For this task, we used the Google News Vectors is a pre-trained word embedding model that represents words as vectors of 300 dimensions. These vectors can be used as input to a CNN for text analysis. We are also using nltk.corpus for getting the stop words.

1) Preprocessing:

Before passing the text as input to the CNN model, we need to preprocess the dataset. The following tasks were performed:

- First the transcript for each participant was extracted and stored along with the participant ID. This was done for train, development and test sets.
- Next the interview transcript for all 3 sets was processed to remove stop words using the nltk.corpus.
- Since there is an imbalance in the dataset, we use upsampling.
- The transcript for each participant is converted into a sequence of word embeddings that represent the text data of the transcript. Each word is then converted into its corresponding vector representation using the GoogleNews-vectors-negative300.
- This data is now fed into the Convolutional Neural Network.

2) CNN Model:

The CNN model used for text analysis consists of 4 layers. It has 2 Conv2D layers with ReLU activation function, 2 dense layers and max pooling layers. One Dense layer has ReLU activation function with 128 nodes and the other dense layer is the output layer with Sigmoid activation function. We are using Adam optimizer with binary crossentropy loss.

3) Results:

We ran the CNN model on the text data using various combinations of values of hyperparameters that included different thresholds, class weights, activation functions, filter sizes and number of filters.

CNN model for text performed with the highest F1 of **Depression Class: 0.47** and **Not Depression Class: 0.67** when we used the **threshold: 0.45** with ReLU activation function.

B. CNN for Audio Processing

CNN model is used to analyze both the acoustic features of the speech recordings of individuals. The model achieved an overall accuracy of 67% and an accuracy of 53% for detecting the depression.

1) Preprocessing:

The covarep.csv file contains pre-extracted acoustic features for each audio recording, including fundamental frequency (F0), energy, and spectral features. These features are used as input to CNN model for depression prediction. The covarep.csv file can be loaded into a pandas DataFrame. The file provides a flag voiced/unvoiced which indicates whether the particular audio segment is voiced or unvoiced. The irrelevant features in case of unvoiced segment is removed.

Since the number data points for depressed data is less up-sampling is done to balance the data distribution

2) CNN Model:

The input layer of the CNN model takes preprocessed acoustic features as input, with a shape of (40000, 74). The model has three convolutional layers with 60, 30, and 15 filters respectively, each with a different filter size. The activation function used in the convolutional layers is ReLU. Additionally, the model has three max pooling layers with a pool size of 3, followed by a flatten layer that converts the output into a 1D array. The dropout layer is used to prevent overfitting and has a rate of 0.8. The CNN model also has two dense layers with 128 and 1 units respectively, using ReLU and sigmoid activation functions.

3) Results:

CNN model for Audio performed with the highest F1 of **Depression Class: 0.53** and **Not Depression Class: 0.74** when we used the **threshold: 0.40** with ReLU activation function.

C. LSTM on Text (Word level)

LSTM is a great model for our case because conversations with Bot will be interrelated within questions. All the questions will be interrelated with each other. Thus the answers of different questions needs to evaluated to find the individual to be depressed or not. LSTM uses time series and it have capabilities to give weights to previous sentences and it will keep few sentences in memory and forget few. LSTM gave good predictions for this case. LSTM uses gates to create a logic on whether to save a part of the sentence or forget it. We are using gates to check how model works with this data.

1) Preprocessing:

We will have time of sentence start and end then the actuals words spoken. For each participant, the output features vector will have, total duration of speech, word duration for all words, each word's index from word vectorizer. Thus the text data is converted into array of numbers. Total duration of speech is given in the input file. Word duration is calculated by using the time stamp of each sentence start and end time stamp divided by total number of words. Text index is got from word vectorizer. Label is available in the data file which is stored in the Ytrain and Ytest. This data is normalized using sklearn. preprocessing. normalize function. Then the shape of each data point is regularized at the shape of 1700. If there is less words, 0 is appended.

2) LSTM Model:

Highway layer has 2 non-linear transforms: a carry and a transform gate which transforms the data which is passed to the dense layers. (Motivation of highway layer is to add nonlinearity to the data even before dense layers as we are using test data for prediction)

We are using tensor Flow. We have used different configurations to find the best fit. As a base line, input layers shape is 1700 by 300. Adam Optimizer, Loss function used is “binary_crossentropy”, Output Activation is “Sigmoid”.

3) Results:

LSTM model for test by Words performed with the highest F1 of **Depression Class: 0.45** and **Not Depression Class: 0.85** when we used the **threshold: 0.6** with ReLU activation function.

D. LSTM for Text & Audio (Sentense Level)

This LSTM model is processing the text as sentence level. There are 250 sentences are fed into the model and each contains up to 17 words. Words are converted into vector of 300 dimensions.

1) LSTM Model:

LSTM model uses 1 audio and text input layer each, 3 highway layers and 4 dense layers for Sentence level predictions.

2) Results:

LSTM model for Text and Audio performed with the highest F1 of **Depression Class: 0.40** and **Not Depression Class: 0.78**, When we used the **threshold >= 0.5** with sigmoid activation function.

Conclusion

The LSTM Word level model performs the best with an **F1-score of 0.77**, which is higher than the other models. The LSTM Sentence Level model also performs relatively well, with an **F1-score of 0.7**. On the other hand, both CNN models perform relatively poorly, with F1-scores below 0.7 for most classes.

Looking at the precision metric, the LSTM models have a higher precision for the 0 class, while the CNN models have a higher precision for the 1 class. However, since we are considering F1-score as the metric of highest performance, we should prioritize models with a higher overall F1-score.

In conclusion, based on the given data, the LSTM Word level model performs the best overall, with the highest F1-score.

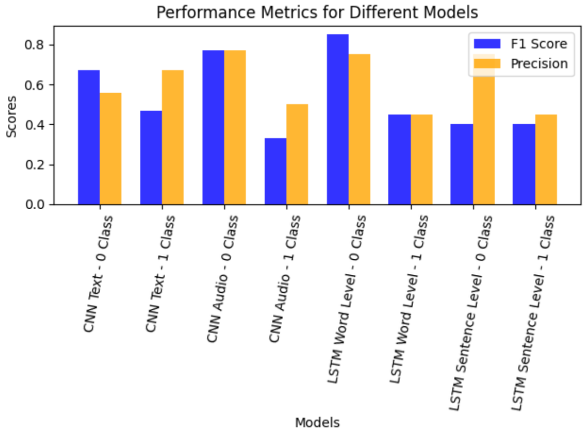

Figure 1. Performance Metrics of Different Models

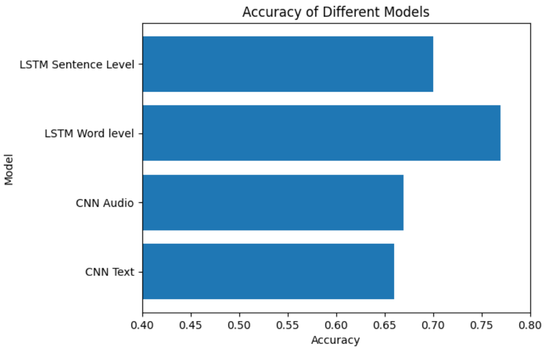

Figure 2. Accuracy of Different Models

Experiments

1) Logistic regression on probablity:

Prediction probabilities from CNN and LSTM had some pattens. So we had a intuition to find the patten and improve the accuracy through logistic regression model. We took 40 predictions from test data set from CNN and LSTM. Created a new logistic regression model to train on 3 data points and validate on other 10 data points. The input X vector will have CNN predicted probability score and LSTM predicted probability score. After training and prediction, the results were not promising. F1 Score is 0, because it has not predicted any one as depressed. Due to lack of data points, the training was not proper

2) Transformers:

Transformers have risen in popularity due to the strong expressive power of transformers in NLP tasks. A BERT based transformer model has been used for the classification of the dataset. For the embedding and attention masks two approaches are followed a custom class for multi head attention block is created for training on data set and an auto tokenizer (BERT) has been used. Two experiments with both the embeddings have been done using the BERT classifier (bert-base-cased) and (beert-large-cased). The resulting classification did not perform better than LSTM (accuracy of 65%) and requires large processing time.

References

[1] Gratch J, Artstein R, Lucas GM, Stratou G, Scherer S, Nazarian A, Wood R, Boberg J, DeVault D, Marsella S, Traum DR. The Distress Analysis Interview Corpus of human and computer interviews. InLREC 2014 May (pp. 3123-3128).

[2] DeVault, D., Artstein, R., Benn, G., Dey, T., Fast, E., Gainer, A., Georgila, K., Gratch, J., Hartholt, A., Lhommet, M., Lucas, G., Marsella, S., Morbini, F., Nazarian, A., Scherer, S., Stratou, G., Suri, A., Traum, D., Wood, R., Xu, Y., Rizzo, A., and Morency, L.-P. (2014). “SimSensei kiosk: A virtual human interviewer for healthcare decision support”. In Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS’14), Paris.

[3] "Kaggle", kaggle.com https://www.kaggle.com/datasets/arashnic/the-depression-datasetK. Elissa, “Title of paper if known,” unpublished.

[4] Manan Gupta, "Depression Detection Through Multi-Modal Data", https://github.com/notmanan/Depression-Detection-Through-Multi-Modal-Data

[5] Sukesh Shenoy, "Depression Detection in Speech", https://github.com/sukesh167/Depression-Detection-in-speech