Depression Detection Using Speech & Text Analysis

Team Members

1. Aksheit Saxena
2. Anish Aralikatti
3. Kavin
4. Anil Kumar
5. Manasa MS
6. Pinnamaraju Raviraj Sitaram

Introduction

- 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.

Data Collection & Strategy

- We are using the DIAC-WOZ dataset 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.
- Dataset is a multi-modal dataset
  - Audio files
  - Video feature files
  - Speech transcripts

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.
- Unzipped the folders for each participant which contained the files of the interview transcript, audio feature files and video feature files.
- 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_Depre-ssion_AVEC2017.csv.

Models Used

- Following are the models tried on DIAC-WOZ dataset to detect depression.
  - CNN for Text
  - CNN for Audio
  - LSTM on Text (Word level)
  - LSTM on Text + Audio (Sentence level)
  - Transformers (Exploration)

CNN for Text Processing

- Convolution Neural Network is a type of deep neural network
- Commonly used for text analysis for tasks such as sentiment analysis, depression detection, etc and image classification
- CNNs are translation invariant and help in identifying patterns of words irrespective of their position in the sentence.
- Text Analysis is done by using word embeddings as input.
- Used Google News Vectors as a pre trained word embedding model.

CNN for Text Processing

Preprocessing

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

CNN for Text Processing

Model

The CNN model used for text analysis consists of 2 Conv2D layers, 2 MaxPooling layers and 2 Dense Layers.

- Has a Conv2D layer with 150 nodes, kernel of (1,5) and ReLU as activation function
- Has a MaxPooling Layer of (1,3)
- Conv2D layer with 75 nodes, kernel of (1,3) and ReLU activation function
- MaxPooling Layer of (1,2)
- Dense layer of 128 nodes with ReLU activation function
- Output layer with 1 node and Sigmoid activation function
- Used Adam optimizer with binary crossentropy loss.

CNN for Text Processing

Results

- Ran the model with various combinations of hyperparameters
- 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**

CNN for Audio Processing

- Data Pre-processing :
  - 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.

CNN for Audio Processing

- Model Details
  - 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

CNN for Audio Processing

- Challenges faced -
  - Computing resource limitations
  - Huge Dataset
  - Preprocessing audio data
  - Unbalanced dataset
- Resolutions -
  - Batchwise training
  - Optimised code to free up resources
  - Studied relevance/irrelevance of audio features
  - Upsampling of data

CNN for Audio Processing

- Results -
  - Fixed threshold = 0.4 varying epochs.
  - Fixed epochs = 6 varying threshold.

	F1		Precision		Accuracy
Threshold	Depressed	Not Depressed	Depressed	Not Depressed	
0.4	0.53	0.74	0.50	0.77	0.67

CNN for Audio Processing

- Results -

```
              precision    recall  f1-score   support

           0       0.63      0.73      0.68        30
           1       0.00      0.00      0.00        13

    accuracy                           0.51        43
   macro avg       0.31      0.37      0.34        43
weighted avg       0.44      0.51      0.47        43

2/2 [==============================] - 2s 618ms/step
Y_pred:  (43, 1)
              precision    recall  f1-score   support

           0       0.66      0.83      0.74        30
           1       0.00      0.00      0.00        13

    accuracy                           0.58        43
   macro avg       0.33      0.42      0.37        43
weighted avg       0.46      0.58      0.51        43

2/2 [==============================] - 2s 569ms/step
Y_pred:  (43, 1)
              precision    recall  f1-score   support

           0       0.68      0.90      0.77        30
           1       0.00      0.00      0.00        13

    accuracy                           0.63        43
   macro avg       0.34      0.45      0.39        43
weighted avg       0.47      0.63      0.54        43
```

Transformer (BERT)

- A BERT based transformer model is used for classifying the text data (transcript)
- 2 Encoders (pre-trained, locally trained) are tried for input.
- Takes huge time and resources for the training part and the final result is worse than the results observed with simpler models.

	F1		Precision		Accuracy
Threshold	Depressed	Not Depressed	Depressed	Not Depressed	
0.4	0.4	0.67			

LSTM (Audio - Text)

LSTM – Long Short term Memory

LSTM is good for contextual data (time series data) Because it have gates which can store context of previous data points. It is a type of RNN

Highway layer:

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)

Implementation:

LSTM based model is used for processing both the audio and text at word level and Sentence level with dimension set to 300 vector for words and 1700 sentences is used for text. We got best results with 2 highway layers, 3 dense layers with 2 as batch size.

	F1		Precision		Accuracy
Threshold	Depressed	Not Depressed	Depressed	Not Depressed	
0.6	0.45	0.85	0.45	0.80	0.77

LSTM Sentence Level

LSTM – Long Short term Memory

LSTM is good for contextual data (time series data) Because it have gates which can store context of previous data points. It is a type of RNN

Highway layer:

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)

Implementation:

LSTM based model is used for processing both the audio and text at word level and Sentence level with dimension set to 300 vector for words and 1700 sentences is used for text. We got best results with 2 highway layers, 3 dense layers with 2 as batch size.

	F1		Precision		Accuracy
Threshold	Depressed	Not Depressed	Depressed	Not Depressed	
0.6	0.45	0.85	0.45	0.80	0.77

Experiments and Possible Optimization Challenges

- Tried to pass the probabilities of CNN and LSTM models predictions, to logistic regression to find the patten and get the better results. Motivation for this is that CNN is predicts based on the amplitude of the entire conversation. LSTM is predicting based on the text (words and sentences). Logistic regression is finding a patten on both models and giving predictions.
- Issue: the train data size and prediction data size is not big enough (we have to take the test data set from CNN and LSTM, we got only 40 data points)
- Using Lexical analysis in addition to speech analysis.
- Focusing on features like audio frequencies.
- Investigating more features to add such as amplitude, pause durations etc.
- Using regression to score depression (say 0-10) onset chances rather than just a binary classification.

Conclusions

- 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.