Conversational Emotion Recognition using Text and Audio Modalities

MSc Research Project
MSc. Data Analytics

Raj Ravindra Kupekar
Student ID: X18186432

School of Computing
National College of Ireland

Supervisor: Prof. Christian Horn

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Raj Ravindra Kupekar

Student ID: X18186432

Programme: MSc. Data Analytics **Year:** 2019-20

Module: Research Project

Supervisor: Prof. Christian Horn

Submission Due Date: 28/09/2020

Project Title: Conversational Emotion Recognition using Text and Audio Modalities

Word Count: 6890 **Page Count** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Raj Ravindra Kupekar

Date: 28/09/2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Conversational Emotion Recognition using Text and Audio Modalities

Raj Ravindra Kupekar
X18186432

Abstract

In this recent advancement, extraction and identification of human emotions plays a crucial role in developing interpersonal relationship between human and machine. Emotion recognition system are now been adopted in TV industries for training purposes of the performers to improve their acting skills for connecting the audiences. Accordingly, research is been carried out to study the effect of the emotional behaviour of human using various modalities independently. In this research work, a MELD database is been used which is a conversational-based repository originated from a 'Friends' TV series. Here, two independent unimodal networks are implemented using the text and audio modalities and their accuracy and performances are relatively compared for any differences. Accordingly, for text unimodal a Bi-LSTM model is observed to be the efficient model with an accuracy of 75% while a LSTM model is seen to be the superior model for audio modality with its highest accuracy of 47%.

1 Introduction

Human emotions govern our daily lives as they are a big part of human understanding and certainly, they affect our decision making as people vary widely in recognizing the emotions of others. Due to the recent advancements in data science, it has enabled us to recognize the human emotion more accurately. Thus, emotion recognition has become a nascent research area in the field of Human-Computer Interaction (HCI), primarily in areas like neuroscience, psychology, marketing, etc. Also, improvements in hardware and wide-scale annotation infrastructure have enabled us to design such a system. Emotion recognition goes beyond the subjective challenges of languages and phenomena such as sarcasm and provides an in-depth understanding of a person's behavior. It also enables us to draw statistics and make decisions to highlight the strengths and improve the weaknesses thus, making it possible to highlight the problem case.

Human deliver their speech using both linguistic and paralinguistic information which are sometimes associated with implicit messages such as emotional states and sentiments. Generally, they are the expressed mental and physiological states linked with the feelings, thoughts, and human behavior. This conveyed emotion reflects both the mood as well as the personality of the human subject. It plays a vital role by demonstrating the speaker's response in person-to-person verbal communication. Although, the expressed words can take different meanings in different emotions. Thus, the identification of emotional states is therefore critical for achieving an optimal and effective communications between humans.

1.1 Background and Motivation

Significantly, less research work is been carried out in emotion recognition in conversation as compared to the recent researches for emotion recognition using audio, video, and text modalities. Accordingly, certain challenges such as context modeling, interlocutor's emotional shifts, irony, or sarcasm tone, and many more are associated with emotion recognition in conversation which makes it difficult to address. Previous researches are usually limited to dyadic conversations and therefore this research work focuses to build a multi-party conversation-based emotion recognition system. In such systems, the uttered words are usually dependent on their conversational context as they act as a set of parameters that drive a participant's associated emotions.

Also, modeling such a system is quite difficult as the emotional dynamics may certainly be interrelated with the previous utterances, thus, making it hard to determine without the conversational history of the speakers. This phenomenon can also be termed as inter-speaker dependencies because of the multimodal nature of the conversation. In conversation emotion detection, sequential dialogues are suffered because of challenges like short utterances like "yeah", "okay", "no" as such words can vary in emotions depending upon the context and discourse of the dialogue. In such cases, depending only on the single modality to perceive the expressed emotion is not enough. Thus, in this research work, a comparative study of conversational emotion recognition is carried out including transcripted text as well as its audio counterparts.

1.2 Research Question

To what extent a unimodal Conversational Emotion Recognition (CER) system build using textual utterances and audio files on the same data, make any difference in accurately recognizing the human emotion using deep neural networks.

Thus, in this research, a conversational dataset extracted from a 'Friends' TV series is been adopted. Accordingly, two unimodal networks are been implemented using textual and audio data files and the outputs of these networks are compared and analyzed to check for any difference in the performance and accuracy.

The structure of this report is described as follows. A related literature review in this research area is been studied in section 2, followed by undertaking an appropriate methodology in section 3. Further, section 3 is divided into five modules in which the business understanding, data preparation, data modeling, and model evaluation methods are discussed, respectively. Section 4 covers the discussion part of the implemented model and its results. Finally, the conclusion of this report is outlined in section 5 and references in section 6.

2 Related Work

In this section, a brief review of recent advances in studies on emotion recognition analysis using text and audio data are been presented. This section is broadly categorized into the following six different subsections.

2.1 Machine learning methods for emotion classification using text

Natural language processing along with supervised machine learning algorithms are effectively used in analyzing and classifying the emotional content of texts. In these recent advancements, the field of emotion recognition using text is been widely explored and studied for the English language. (Azmin & Dhar, 2019) built a multiclass emotion recognition system from a Bangla language text corpus using a traditional multinomial Naïve Bayes (NB) classifier. For this study, a large corpus of comments from Facebook and other bloggers where been collected which were categorized into three different emotion labels- happy, sad, and anger. Hidden Markov Model (HMM) based POS tagger, Bi-gram word model, and TFIDF vectorizer are been used as a feature selection and extraction technique. A multinomial NB classifier is implemented for classifying the Bangla emotions and effects of various feature selection combinations are been analyzed.

Text analysis of various languages other than English are been explored intensively as every other language comes with different levels of challenges. (Gürcan, 2018) aims to study and classify Turkish news text using different machine learning algorithms as the linguistic structure of the Turkish language is agglutinative and is different from others in terms of space. The dataset is built by gathering the texts from news sites and manually feeding the emotion labels to it. A document-term weighted matrix is used to transform the pre-processed textual data into a numerical matrix, such that different supervised machine learning algorithms can be implemented. In this study, the model is evaluated using a different trainable set extending from 500 texts to 2000 texts where; the experimental results showed that the multinomial NB classifier outperformed other implemented machine learning algorithms.

2.2 Word embedding technique for text feature extraction

In text analysis, often the semantic and syntactic information are overlooked. Thus, in many pieces of research, a word vectorization technique is adopted such that the semantic similarities of the vocabulary are maintained. (Jin & Xu, 2020) presents a study of the performance of three different word vectorizers namely- Word2vec, Doc2vec, and TFIDF models; to convert the word vocab into a high dimensional vector. These vectors are then subjected to a dimensional reduction technique using a Principal Component Analysis (PCA) where 10 PCA component is selected as an optimal dimension. The sentiment classification is carried out using traditional machine learning approaches were the results using the TFID model outperformed other embedding techniques. For this study, precision and F1-score are used as a performance evaluation metrics.

In word embedding technique, the vocabulary list is been built by training the model on the given textual data. Recent advances in this field have enabled us to use a pre-trained word embedding model which reduces the time complexity for training. In this context, a SEEN model is presented by (BATBAATAR, et al., 2019), which makes use of a pre-trained word embedding model for capturing the semantic relationship. The model is trained using two sub-networks where the Bi-LSTM network is used for capturing the semantic relationship between the words while the CNN network is used for capturing the emotional relationship. The Bi-LSTM network is built using a Word2vec as a feature extraction embedding technique, whereas EWE word embedding is adopted for building the CNN network.

2.3 Deep neural networks for emotion classification using text

Deep learning for text analysis has enabled to make use of an inbuilt embedding layer using a one-hot encoding technique (Zheng, 2019). Embedding layer is used for obtaining the fixed-length vector representation of a given text corpus. The encoded one-hot vector is then subjected to a cosine similarity matrix to reduce the high dimensionality. Content words and emotional function words are integrated to estimate the final emotional output. The extracted features are then fed as an input to the LSTM network such that the long-term dependencies of the words are maintained.

Text analysis can be carried out for detecting the emotion from psychiatric social texts and effectively highlighting the stress to diagnose the level of depression in online communities (WU, et al., 2020). A deep learning framework with a combination of Bi-LSTM and CNN network is used for the identification of emotion labels. The feature extraction from raw data is done using a pre-trained GloVe word embedding technique with a 300-vector dimension. The learned word vectors are then subjected to a Bi-LSTM network for maintaining the semantic information. The output of this network is then fed to a CNN network for capturing the local important features of word vectors.

Over the course of time, text analysis is now been implemented in almost every sector. Traditionally, emotion recognition is carried out on informal textual contents like news, comments, and many more. (AHMAD, et al., 2020) presents a system that detects and classifies the emotional states from the formal poetry text using a deep learning approach. As poetry texts are mostly comprised of stop words, the removal of stop words is not considered in text pre-processing. A word embedding layer is then built using a Keras and further selecting the best n-grams features from it. An attention layer is built on the top of the Bi-LSTM layer to obtain the contextual information, followed by a CNN architecture as a classification layer. The model is evaluated using precision and F1-score as performance metrics and the contribution of attention layer is observed to be crucial in attaining the highest model performance.

In many researches, a combination of two or more neural networks are built for decision making. Using the same principle, researches are been conducted to study the effect of a combination of two different word embedding techniques. In (Al-Omari, et al., 2020), an EmoDet2 system is designed which is a combination of Word2vec and BERT embedding techniques for classifying the emotion having four classes. Emoticon handling is implemented for converting emojis into text, along with Ekphrasis package for handling the spelling mistakes. A 300-dimensional vector is extracted using a pre-trained Word2vec embedding which is also trained using a GloVe embedding layer. The BERT embedding is trained on the raw data using a transformer package to obtain a 173-dimensional vector, followed by a 145-dimensional vector using an Affective Tweets Weka package for capturing the semantic features. These extracted features are then combined, and a neural network is implemented for decision making.

2.4 Machine learning methods for emotion recognition using speech/audio

In the field of human-computer interaction (HCI), emotion recognition using speech is gaining a lot of attention from researchers. In (Manamela, et al., 2018), a SER system is designed using machine learning methods by means of support vector machine (SVM) and K-nearest neighbor (KNN) for solving speech emotion recognition tasks. For this study, audio features are extracted using 34 short-term features including time domain, frequency domain, and cepstral domain with the help of a pyAudio analysis. Furthermore, the best-trained algorithm is selected using an Auto -WEKA data mining tool with its best-suited hyperparameters.

Generally, human speech delivers information and context through speech, tone, pitch, and many such traits of the human vocal system. In this context, a machine learning algorithm is trained on a North American English audio database (Deshmukh, et al., 2019). Pre-emphasis and de-silencing of the input audio signals are performed to get a noise-free audio sample followed by framing and windowing of the signals to get a clean processed audio segment. Here, feature extraction of pre-processed audio signals is carried out in three steps where initially energy spectrum and MFCC features are extracted followed by obtaining the pitch values of these audio signals. The extracted features are combined and trained using an SVM algorithm and evaluated using the mean and mode values for the extracted features.

2.5 Feature extraction techniques for audio/speech

In the field of speech analysis, different audio/speech database are been designed such that various researches can be carried out. Speech-based emotion recognition system is been designed and trained using MATLAB software to study the emotion classification on a Berlin Emo database having audio recording from 12 different actors in the German language (D, et al., 2019). Here, feature extraction is done by adopting various combinations of extraction procedures where initially the energy level of each audio file is extracted followed by declaring a speech rate for sampling the audio file. Moreover, the pitch information of the audio file is obtained such that the MFCC features are extracted with a frame length of 512 samples.

Feature normalization is yet another method that can be combined with the feature extraction technique to get better results from raw audio files (Iqbal & Barua, 2019). As unstructured data may vary in range, the extracted features are normalized and weighed on the same scale by subtracting the feature values from its mean and dividing it by its standard deviation. Subsequently, normalization aided to gain the highest model performance using a Gradient Boosting algorithm when compared to previous works.

Although, in audio analysis features extraction procedure can be blended in synchronization with the different characteristics of the nature of audio files. (Yoon, et al., 2019) shows an emotion recognition study where the MFCC feature extraction is carried out as a delta and acceleration coefficients. These coefficients are added with a prosodic feature which is made up of the voicing probability and loudness contours, to add the low-level speech signals for better performance. For modeling of these features, and Audio Recurrent Encoder (ARE) is implemented which gave a comparatively better performance with less trainable weights. In [14], a multi-hop attention layer is built on the top of the Bidirectional Recurrent Encoder (BRE) such that only the relevant extracted features are used for decision making. The attention layer is built and trained using three stages where the output of each attention layer is fed as an input to the other successive layer.

2.6 Deep neural networks for emotion recognition using speech/audio

In audio analysis, both verbal and non-verbal sounds contribute to recognizing the human emotion. (Huang, et al., 2019) presents a study that uses a deep learning network for learning the audio patterns from an audio segment. PRAAT tool and Prosodic Phrase auto tagger is used for extracting the verbal and non-verbal features. Here, a Bi-LSTM network is built on the top for capturing the insights from these features. The neural network is tuned using a hyperparameter tuning where two different optimizers namely- Adam and AdaDelta are used. The model is evaluated using accuracy as a performance metric and the results show that the model produced an accuracy of 63% which is 8% higher than the studied existing researches.

To deal with unwanted information in non-speech voice segments a combination of Bi-LSTM and attention layer is presented in (Suganya & Charles, 2019). Here, the non-verbal sound is subjected to silence removal filter using a threshold value, and audio features are then extracted using 13 MFCCs and 13 chroma embedding technique. The long-term dependencies in the extracted features are preserved using a Bi-LSTM network with a batch size of 32 and 0.001 as a default learning rate. With an objective to reduce the time complexity and improve the model performance only the relevant speech segments that contribute in decision making are processed using an attention layer. The system is also evaluated by modifying the values for minimum silence duration and threshold. Finally, an ideal value of 60ms is used as a minimum silence duration while 0.1% as the optimal threshold value.

Several studies have been developed for an end-to-end system for speech recognition using neural networks. (Atmaja & Akagi, 2019) a SER system is designed using a two-layer LSTM network stacked upon a 2D CNN layer which is used to model the extracted features. A zero mean and unit variance with a 20s long sequence is fed as an input to the CNN layer. A 64-time impulse filter with a kernel size of 8 is used as a temporal convolution layer with a max-pooling size of 10. Drop out layers are included in the deep neural networks as the number of parameters is added with a drop rate of 0.2. Importantly, in this study instead of MSE as object function, a Concordance Correlation Coefficient (CCC) is used as a loss function where the model performance is boosted by 8% as compared to previous studies.

3 Research Methodology

In this section, the research methodology is discussed which is been used to develop the proposed system. For this research work, a Cross Industry Process for Data Mining (CRISP-DM) methodology is been implemented. It provides a well-structured approach in executing a data mining project with a broad iterative process through which a problem case can be solved. It consists of a six-step process from understanding the business case to generating insights from the given data which is illustrated in Fig. 3. In this project, two unimodal networks using text and audio data are built using a deep learning technique.

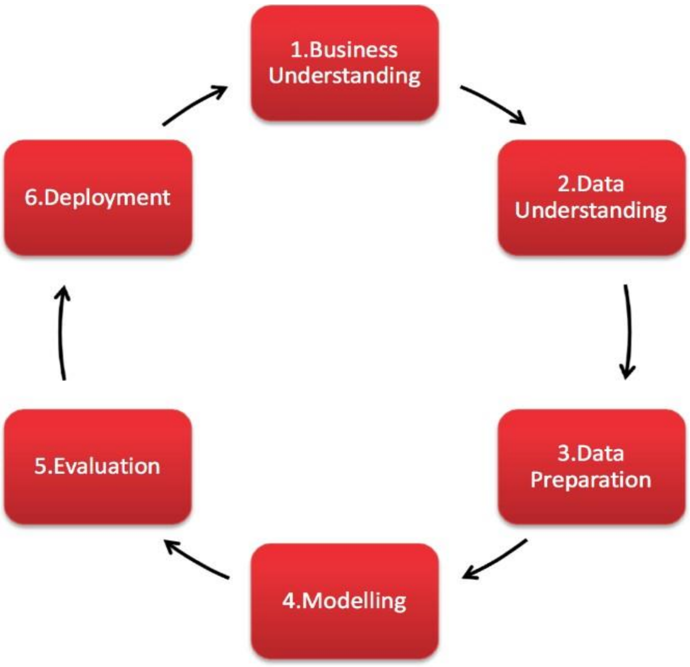


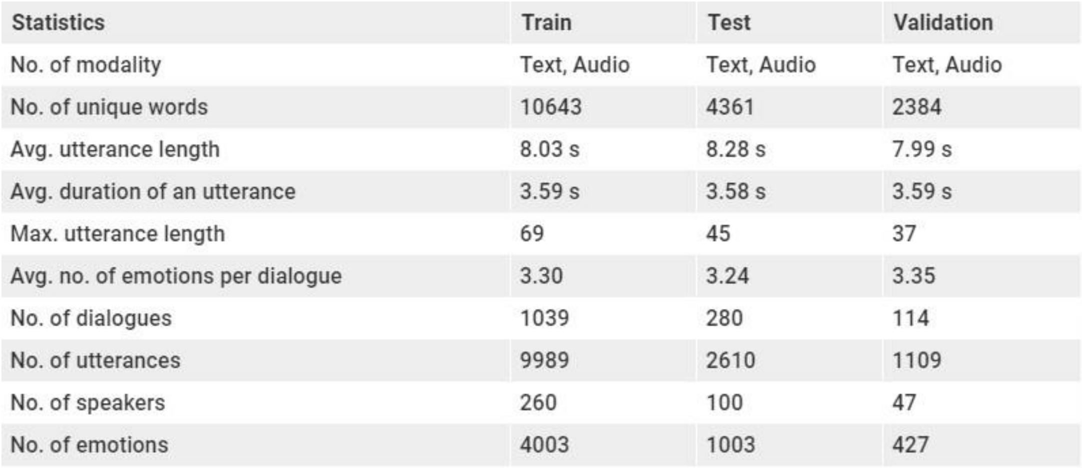
Fig. 3 Cross Industry Process for Data Mining (CRISP-DM)

3.1 Business Understanding

In this stage, the primary objective of the business is determined by defining the associated clause of the problem case. In many fields, the emotion detection system is incorporated in deciding the contextual emotions such that appropriate measures can be undertaken for a profitable business. For this application, transcripts and audio clips from a popular TV series is been analyzed to draw emotions of performers for quality purposes. This will help the movie production companies to guide their artists to present it in a more authentic way to deliver the correct contextual emotions to their audiences. Also, such systems are also adopted in marketing industries to understand the emotions of the existing and potential customers to improve their customer experiences and business. Thus, an emotion detection system will subsequently, aid in determining the consumers' expectations and requirements.

3.2 Data Understanding

This stage requires an initial understanding of raw data as well as the class distribution of the target variable. For this research work, Multimodal Emotion Lines Dataset (MELD) is been used which is having the same dialogue instances with both textual and audio modalities for emotion analysis. It is consisting of about 1400 dialogues and more than 13000 utterances from a popular TV series 'Friends'. Each utterance is been labeled using seven different emotion labels namely- anger, disgust, surprise, neutral, fear, sadness, and joy. Accordingly, for both modalities training, testing, and validation data are separately provided. Textual data is available in a comma-separated file (.csv) consisting of different attributes related to each individual scene. Every record is associated with the dialogue and utterance number with its corresponding speaker and emotion. In the case of audio data, .wav extension files are available. These audio files are named using the dialogue and utterance numbers such as to map them with their corresponding emotions given in .csv files. The figure below shows the data description with its associated attributes.



Statistics	Train	Test	Validation
No. of modality	Text, Audio	Text, Audio	Text, Audio
No. of unique words	10643	4361	2384
Avg. utterance length	8.03 s	8.28 s	7.99 s
Avg. duration of an utterance	3.59 s	3.58 s	3.59 s
Max. utterance length	69	45	37
Avg. no. of emotions per dialogue	3.30	3.24	3.35
No. of dialogues	1039	280	114
No. of utterances	9989	2610	1109
No. of speakers	260	100	47
No. of emotions	4003	1003	427

Table 3.2 Data description

The class distribution of the target variable is unbalanced in nature with a higher number of records for neutral emotion covering for approximately 50% of the total count and on other hand fewer records for disgust and fear emotions. The bar plot for the value counts of the target variable is shown below.

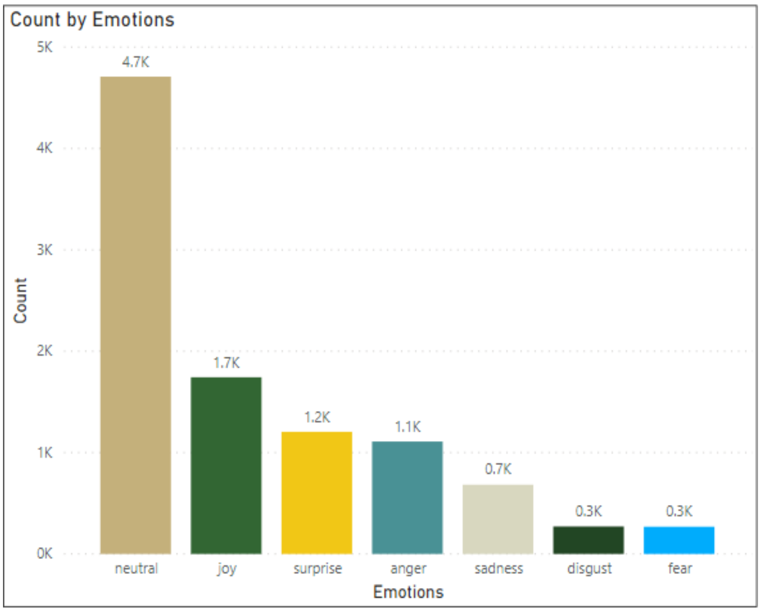


Fig 3.2.1 Class distribution emotions

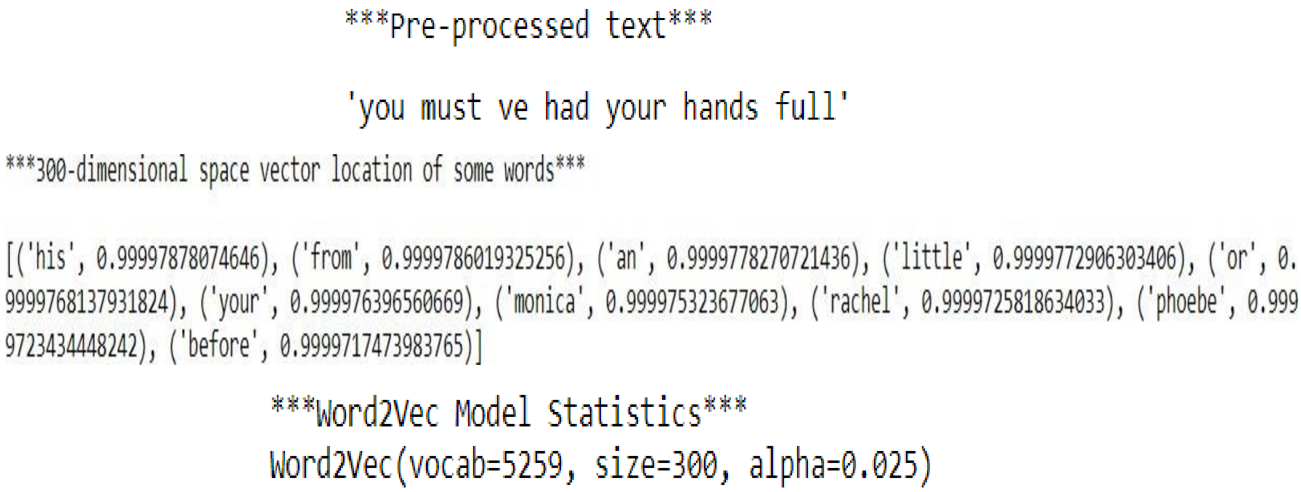
Thus, for text modality, the data is available in CSV format while the audio data is available in wav format.

3.3 Data Preparation

To train a deep learning model, it is mandatory to transform the data in the appropriate format to draw insights from it. In this stage, the relevant data is selected for analysis followed by cleaning the data and extracting the features from it. This section is subdivided into preprocessing and feature extraction for text and audio.

3.3.1 Pre-processing and feature extraction for text

In-text analysis, the strings values are transformed into a vector representation to train a deep learning or machine model. Initially, the data is checked for any missing values in the given text corpus. This is followed by the cleaning of raw data using a nltk library. The non-alphabetical words along with punctuation marks are removed from the given text utterances as they do not contribute in decision making. As, text utterances are short dialogues that are mostly built up using stopwords, critically, these stopwords are not eliminated from the text corpus. Also, performing stopwords elimination would probably lead to no textual words in utterances. A Word2vec embedding matrix is trained on the cleaned processed text to obtain its vector representation. It is used to maintain the semantic and syntactic relationship between the words in the text utterances. The pre-processed training data is used to build the word embedding vocab where a Word2vec matrix is build using a skip-gram model with a 300-dimensional vector. Additionally, a down sampling parameter with a value of 0.001 is used to down sample the most frequent words. The below figure shows the pre-processed text along with word2vec statistics.



```
***Pre-processed text***

'you must ve had your hands full'

***300-dimensional space vector location of some words***

[('his', 0.99997878074646), ('from', 0.9999786019325256), ('an', 0.9999778270721436), ('little', 0.9999772906303406), ('or', 0.
9999768137931824), ('your', 0.999976396560669), ('monica', 0.999975323677063), ('rachel', 0.9999725818634033), ('phoebe', 0.999
9723434448242), ('before', 0.9999717473983765)]

***Word2Vec Model Statistics***
Word2Vec(vocab=5259, size=300, alpha=0.025)
```

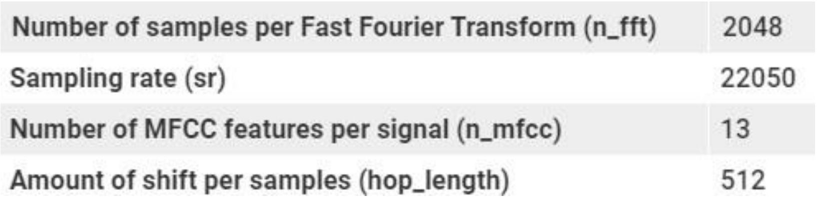
Fig 3.3.1 Text pre-processing and feature extraction outputs

Thus, feature extraction of textual data is carried out successfully using a word2vec embedding matrix with a 300-dimensional vector on a vocab size of 5259 words.

3.3.2 Pre-processing and feature extraction for audio

In this section, the feature extraction of audio data is carried out using a Mel-frequency Cepstral Coefficient (MFCC). Before extracting the coefficients, a single audio file is loaded, and certain parameters are observed from plotting its waveform to generating its power spectrum and MFCC graph. To generate the MFCC features, a Librosa library is used which is having an MFCC feature function. As, frequency signals in an audio change over time, it is necessary to define a frame length to generate samples per audio to perform Fourier Transformation. Here, the audio files are sampled at the sample rate of 22050. Further, windowing of samples is required to tackle the assumption of the infinite nature of the audio signal and to reduce spectral leakage. Accordingly, this is done by setting a value for the number of samples per

Fast Fourier Transform (n_fft) and then generating a periodogram for that signal. A hop length parameter is set which determines the amount of shift the window shift towards the right. The following table shows the parameter values selected to extract the MFCC coefficients.



Number of samples per Fast Fourier Transform (n_fft)	2048
Sampling rate (sr)	22050
Number of MFCC features per signal (n_mfcc)	13
Amount of shift per samples (hop_length)	512

Table 3.3.2 Mel-frequency cepstral coefficient parameters

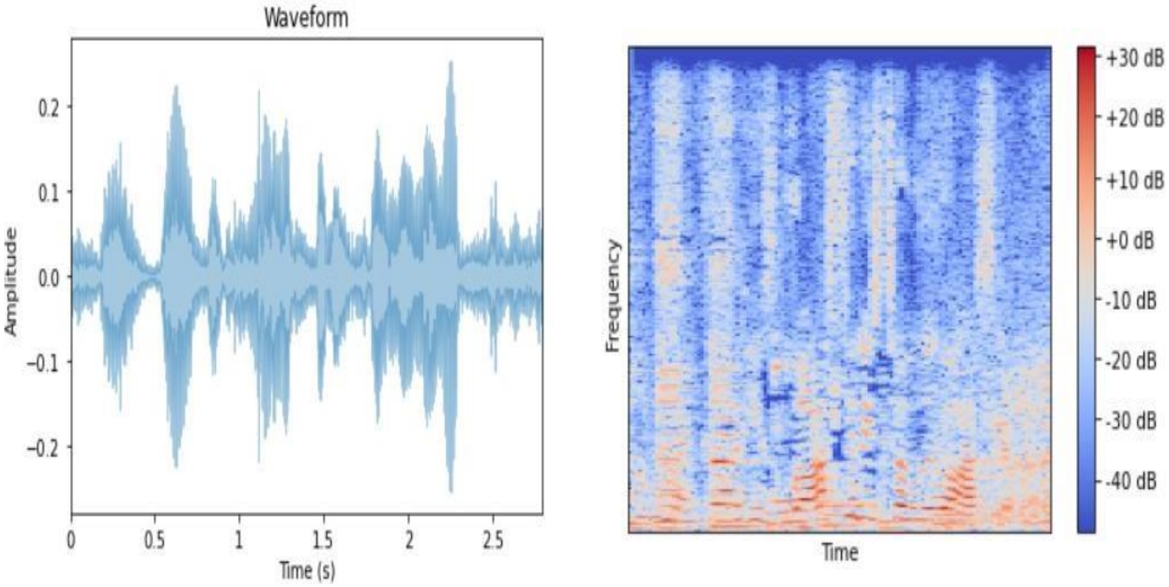
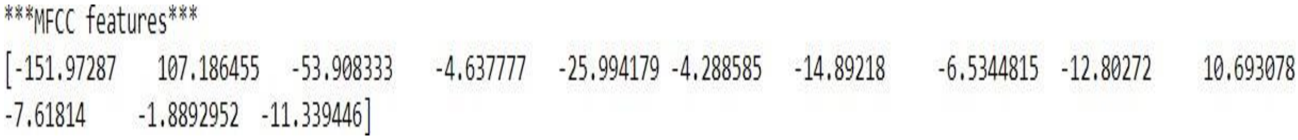


Fig 3.3.2.1 Waveform and Mel-frequency cepstral coefficient of a sample audio signal



```
***MFCC features***
[-151.97287    107.186455   -53.908333    -4.637777    -25.994179 -4.288585    -14.89218      -6.5344815  -12.80272     10.693078
-7.61814      -1.8892952  -11.339446]
```

Fig 3.3.2.2 Mel-frequency cepstral coefficient of a sample audio signal

The above figure shows the pre-processing of a sample audio signal. Fig.3.3.2.1. shows the waveform plotted using amplitude and time for a sample signal. The subsequent figure shows the image representation of MFCC features for than signal and finally Fig.3.3.2.3. outlines the vector representation of the MFCC features. Thus, the MFCC is used for extracting the features from the audio signals having 13 MFCC features.

4 Design Specification

In this section, the design architecture of our CER system is been illustrated. The text unimodal and audio unimodal are implemented by following the mentioned stages in the diagram. At initial stages, the pre-processing of the raw is carried out followed by feature extraction using Word2Vec and MFCC for text and audio, respectively. Accordingly, Bi-LSTM and LSTM models are built for decision making. The flow diagram for this research work is shown in figure below.

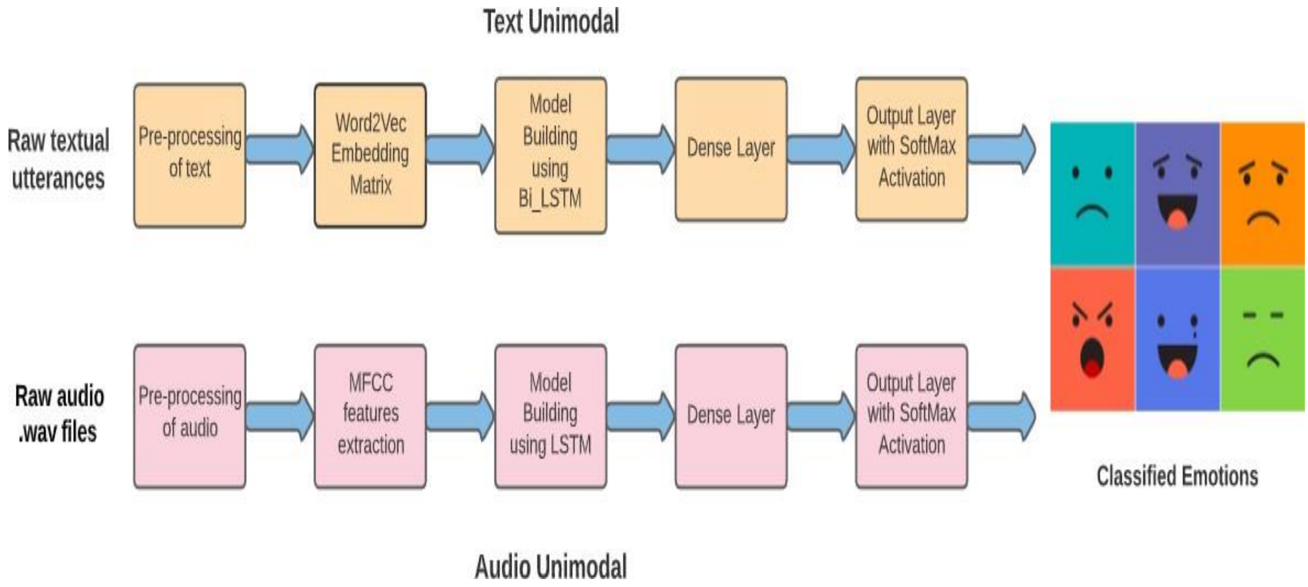


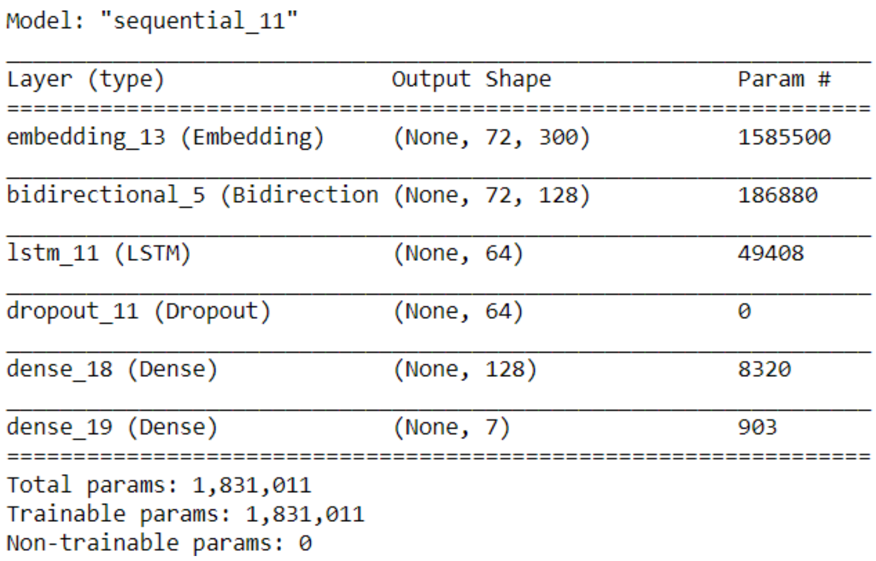
Fig 4 Conversational Emotion Recognition system architecture

5 Implementation

In this section, model building for emotion recognition using text and audio are discussed. Accordingly, each unimodal is built using two deep learning networks.

5.1 Model building for text modality

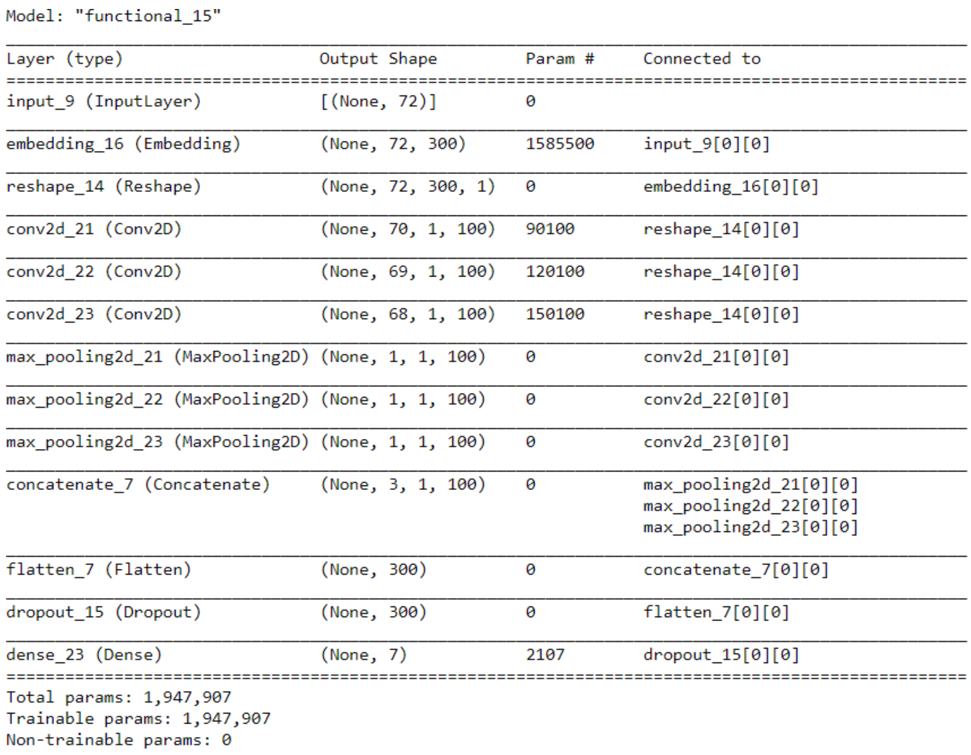
For text analysis, two unimodal neural networks are built using a Bidirectional Long Short-Term Memory (Bi-LSTM) and a 2D Convolution Neural Network (2D-CNN). The Bi-LSTM model is built to capture the semantic and syntactic relationship between the text utterances. Thus, in this manner, the long-term dependencies in the text are maintained by the model. A sequential model is build using a Bi-LSTM layer of 64 neurons and with a uniform kernel initializer. An LSTM layer with 64 activation neurons is further added in the network followed by a dense layer of 128 neurons. Activation function ReLu is used for the activation of these layers except the SoftMax activation function for the output layer with 7 output neurons. A recurrent dropout rate of 0.2 is used in both the Bi-LSTM an LSTM layer. The below figure shows the summary of the implemented Bi-LSTM model.



```
Model: "sequential_11"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_13 (Embedding)     (None, 72, 300)           1585500   
_________________________________________________________________
bidirectional_5 (Bidirection (None, 72, 128)           186880    
_________________________________________________________________
lstm_11 (LSTM)               (None, 64)                49408     
_________________________________________________________________
dropout_11 (Dropout)         (None, 64)                0         
_________________________________________________________________
dense_18 (Dense)             (None, 128)               8320      
_________________________________________________________________
dense_19 (Dense)             (None, 7)                 903       
=================================================================
Total params: 1,831,011
Trainable params: 1,831,011
Non-trainable params: 0
```

Fig 5.1.1 Bi-LSTM network for text

Furthermore, to compare the results of the Bi-LSTM model, another neural network is trained using a 2D-CNN model. Here, the model is trained using three 2D CNN layers with three max-pooling layers. ReLu activation function is used for the activation of the CNN layers while the SoftMax activation function for the output layer. All three CNN layers are trained using 100 filters with a filter size of 3, 4, and 5 for the three layers, respectively. Also, a stride of 1x1 matrix is used in max-pooling layers where these three max-pooling layers are concatenated. The following figure shows a summary of the implemented CNN model.

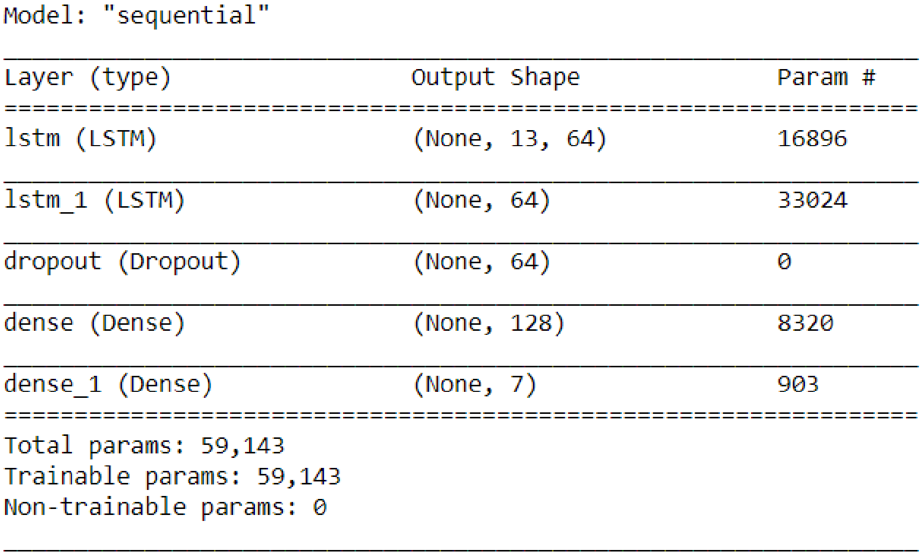


```
Model: "functional_15"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to                     
==================================================================================================
input_9 (InputLayer)            [(None, 72)]         0                                            
__________________________________________________________________________________________________
embedding_16 (Embedding)        (None, 72, 300)      1585500     input_9[0][0]                    
__________________________________________________________________________________________________
reshape_14 (Reshape)            (None, 72, 300, 1)   0           embedding_16[0][0]               
__________________________________________________________________________________________________
conv2d_21 (Conv2D)              (None, 70, 1, 100)   90100       reshape_14[0][0]                 
__________________________________________________________________________________________________
conv2d_22 (Conv2D)              (None, 69, 1, 100)   120100      reshape_14[0][0]                 
__________________________________________________________________________________________________
conv2d_23 (Conv2D)              (None, 68, 1, 100)   150100      reshape_14[0][0]                 
__________________________________________________________________________________________________
max_pooling2d_21 (MaxPooling2D) (None, 1, 1, 100)    0           conv2d_21[0][0]                  
__________________________________________________________________________________________________
max_pooling2d_22 (MaxPooling2D) (None, 1, 1, 100)    0           conv2d_22[0][0]                  
__________________________________________________________________________________________________
max_pooling2d_23 (MaxPooling2D) (None, 1, 1, 100)    0           conv2d_23[0][0]                  
__________________________________________________________________________________________________
concatenate_7 (Concatenate)     (None, 3, 1, 100)    0           max_pooling2d_21[0][0]           
                                                                 max_pooling2d_22[0][0]           
                                                                 max_pooling2d_23[0][0]           
__________________________________________________________________________________________________
flatten_7 (Flatten)             (None, 300)          0           concatenate_7[0][0]              
__________________________________________________________________________________________________
dropout_15 (Dropout)            (None, 300)          0           flatten_7[0][0]                  
__________________________________________________________________________________________________
dense_23 (Dense)                (None, 7)            2107        dropout_15[0][0]                 
==================================================================================================
Total params: 1,947,907
Trainable params: 1,947,907
Non-trainable params: 0
```

Fig 5.1.2 CNN model for text

5.2 Model building for audio modality

In audio emotion recognition, a Long Short-Term Memory (LSTM) model is implemented. It is similar to the Bi-LSTM text unimodal, but here the Bidirectional layer is not included in the network. The LSTM layer is trained using an input shape of (13,1), where the first argument is the number of MFCC features while the second argument represents the time stamp.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm (LSTM)                  (None, 13, 64)            16896     
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                33024     
_________________________________________________________________
dropout (Dropout)            (None, 64)                0         
_________________________________________________________________
dense (Dense)                (None, 128)               8320      
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 903       
=================================================================
Total params: 59,143
Trainable params: 59,143
Non-trainable params: 0
_________________________________________________________________
```

Fig 5.2. Model summary of LSTM network for audio

The above figure shows the model summary. Here, recurrent dropout and dropout layers are been adopted for dropping the layer with a dropout rate of 0.2, respectively. Accordingly, the output layer is the dense layer with 7 output neurons and SoftMax as an activation function. In audio analysis, other baseline models are build using a machine learning approach. Thus, machine learning models like Support Vector Machine (SVM), Random Forest, Decision Tree, and AdaBoost Ensemble model are implemented. The models are then compared with the deep learning model for evaluation purposes.

6 Evaluation

In this section, the implemented models are evaluated to get an optimal solution for emotion recognition. Similarly, the evaluation section is subdivided into the evaluation of text unimodal and evaluation of audio unimodal.

6.1 Hyperparameter tuning of text unimodal

Initially, the Bi-LSTM model is build using the default parameters. For activation of a Bi-LSTM layer, a ReLu activation function is used to avoid the problems related to vanishing gradients. Also, a recurrent dropout layer is added in Bi-LSTM and LSTM layer to avoid the overfitting of the model. The model is further evaluated by tuning the hyperparameters like optimizers, loss function, and applying early stopping criteria. Firstly, an Adam optimizer is used for model building followed by attempting for Adagrad and RMSprop as an optimizer. Secondly, the loss function is tuned by taking sparse categorical cross-entropy and categorical cross-entropy. Thirdly, the model is subjected to early stopping criteria, where accuracy and validation accuracy are selected for monitoring the early stopping of the model. Moreover, the confusion matrix is used for evaluating the predicted outcomes by the model. Similarly, the CNN model is also evaluated using the similar hyperparameters.

6.2 Results of text unimodal

In this research work, as two different neural networks were implemented in classifying the textual emotions, the Bi-LSTM model outperformed the CNN model with an accuracy of 74%. The best-chosen hyperparameters for text unimodal are RMSprop for optimizer, categorical cross-entropy as loss function, and early stopping using accuracy as a monitoring parameter. As, early stopping criteria were adopted the number of epoch size was limited to approximately 19 epochs for both the implemented models. The figure below depicts the model's accuracy on training data and validation data.

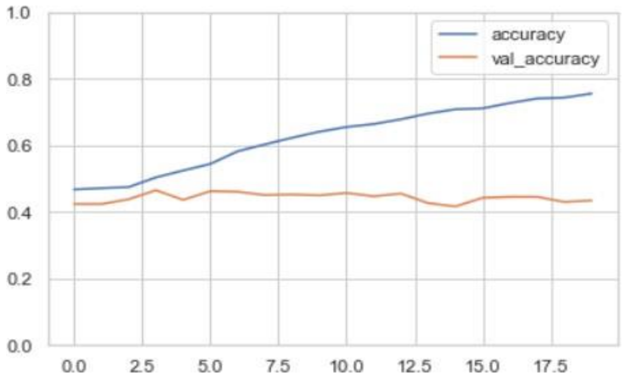


Fig 6.2.1 Accuracy of the model on training and validation data

It is observed that the validation accuracy remains steady at approximately 44% while the model accuracy increases and reaches approximately 75%. Although, while comparing the accuracy graph of training and testing data, it is observed that the accuracy of testing data drops below the validation accuracy. This phenomenon is outlined in the following graph. Accordingly, the loss function graph is also illustrated in Fig 6.2.2. It is observed that loss function for training data remained constant while the loss function of testing data reached its peak at some point but eventually remained equal to the training loss at the end.

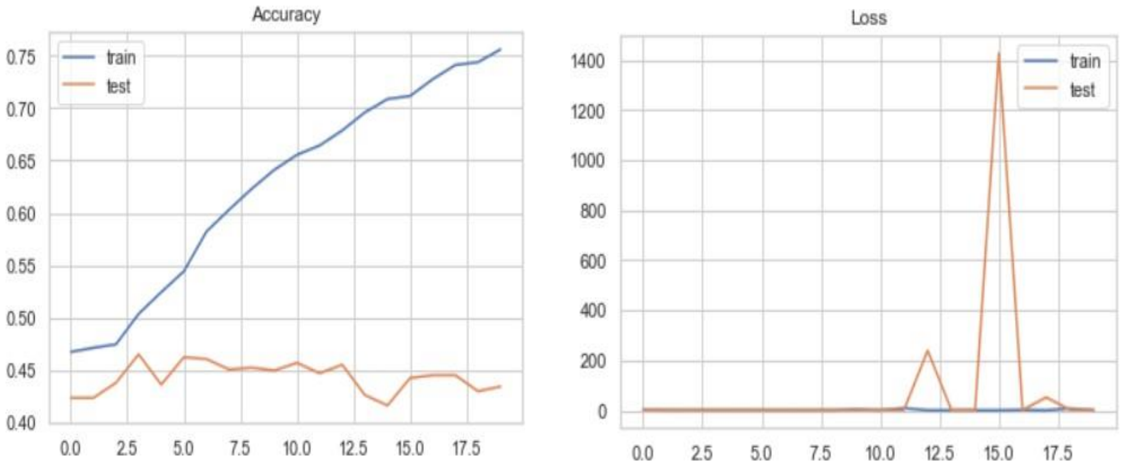


Fig 6.2.2 Accuracy graph and loss function graph for training and testing data

Finally, the performance of our final Bi-LSTM model is evaluated using a confusion matrix. As there are seven emotion categories, it is observed that only the ‘neutral’ emotion label is precisely classified with a 71.3% of prediction percentage followed by ‘joy’ with approximately 32% of correct predictions. This is because of the imbalanced nature of the data with a greater number of records for ‘neutral’ emotion labels followed by ‘joy’.

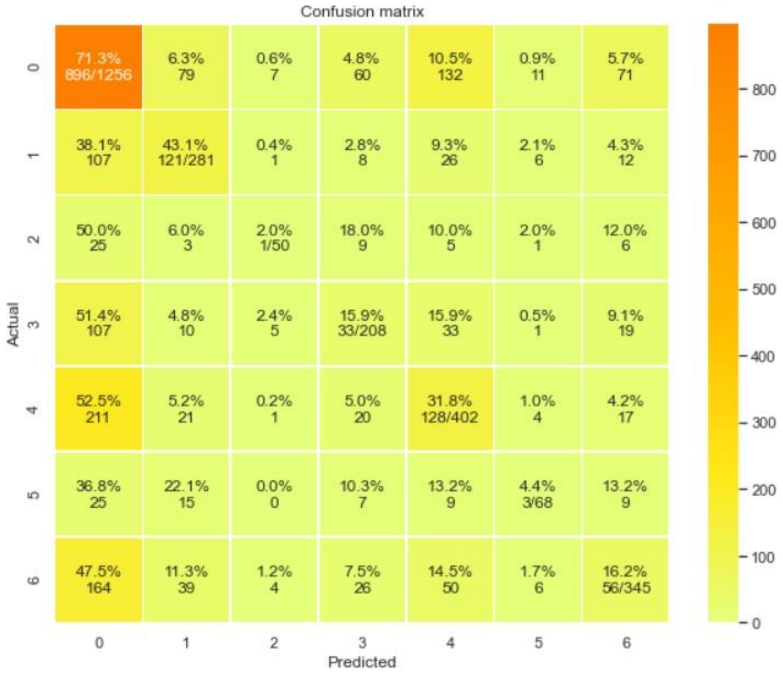


Fig 6.2.3 Confusion matrix of Bi-LSTM model

Thus, the in-text unimodal Bi-LSTM model is selected as the best optimal model as compared to a CNN model. The following are the results for the CNN text unimodal. These figures consist of the confusion matrix for the predicted test values, accuracy graphs for the validation, training, and testing data and loss function graph for train and test data.

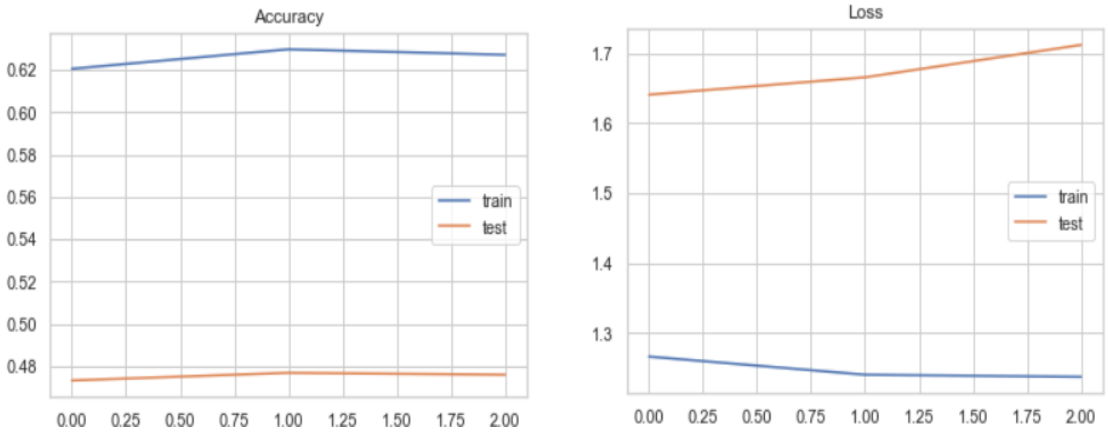


Fig 6.2.4 Graph plots of accuracy and loss function for train and test data

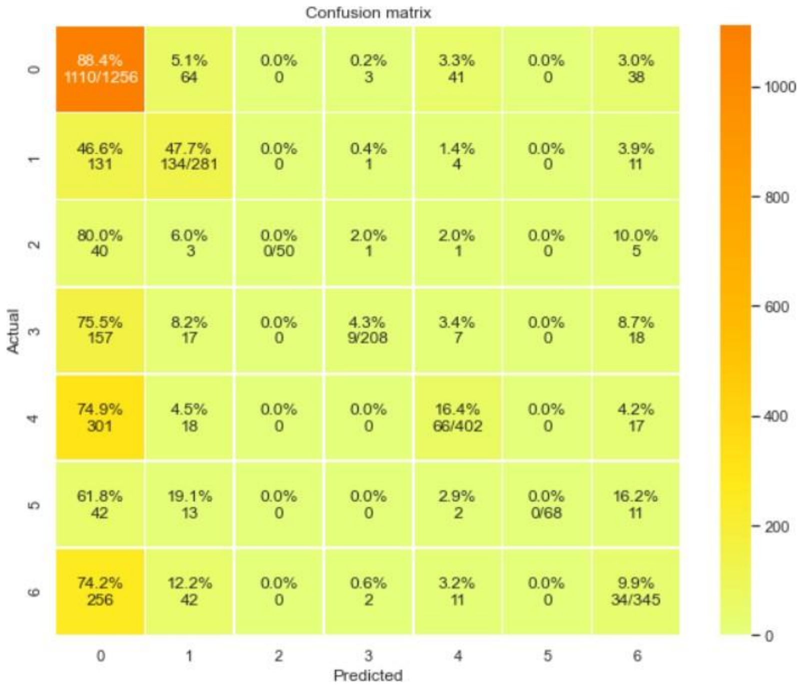


Fig 6.2.5 Confusion Matrix for CNN model

6.3 Hyperparameter tuning for audio unimodal

For audio analysis, several machine learning models are implemented as a baseline model and an LSTM model is built for emotion recognition. In the case of machine learning models, Support Vector Machine (SVM), Random Forest, Decision Tree, and AdaBoost Ensemble model are implemented. The SVM model is tuned by taking different parameters for kernel such as RBF and linear. Accordingly, the random forest and decision tree algorithms are tuned by altering the parameters for n_estimators, minimum and maximum dept, and criteria. In the case of the LSTM model, a ReLu activation function is used to avoid the problems related to vanishing gradients. Also, a recurrent dropout layer is added in Bi-LSTM and LSTM layer to avoid the overfitting of the model. The model is further evaluated by tuning the hyperparameters like optimizers, loss function, and applying early stopping criteria. Firstly, an Adam optimizer is used for model building followed by attempting for Adagrad, SGD, and RMSprop as an optimizer. Secondly, the loss function is tuned by taking sparse categorical cross-entropy and categorical cross-entropy. Thirdly, the model is subjected to early stopping criteria, where accuracy and validation accuracy are selected for monitoring the early stopping of the model. Moreover, the confusion matrix is used for evaluating the predicted outcomes by the model.

6.4 Results of audio unimodal

In audio recognition, an LSTM model is observed to be a superior model as compared to the other baseline models. The model accuracy of the LSTM model is observed to be around 0.47 which is slightly higher than the model accuracy of other baseline models. The best-chosen hyperparameters for audio unimodal are SGD for optimizer, categorical cross-entropy as loss function, and early stopping using accuracy as a monitoring parameter. The figure below depicts the model's accuracy on training data and validation data.

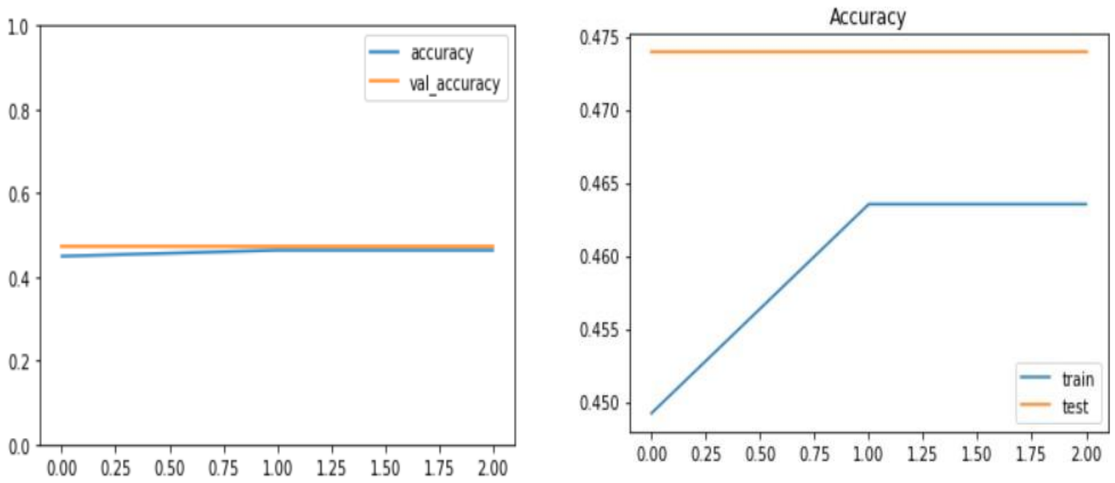


Fig 6.4.1 Accuracy of the model on training, testing and validation data

From the above figure, it is clear that the model performed poorly in emotion recognition, and also the model accuracy on training and testing data is observed to be roughly equivalent at about 45% throughout the training period. Accordingly, the training and testing accuracy is compared in Fig 6.4.1. Here, it is seen that testing accuracy remains constant at around 47% which is comparatively higher than the training accuracy which is having the highest accuracy of 46%. Further, for evaluating the model confusion matrix is generated for the classified observations.

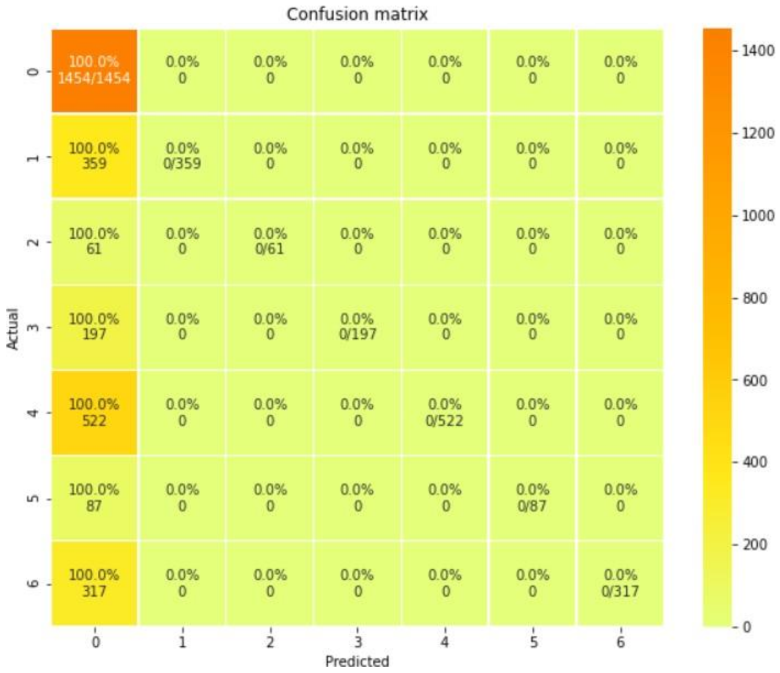


Fig 6.4.2 Confusion Matrix for LSTM model

Similar to the textual unimodal, the audio emotion recognition is also having 7 emotions labels. The confusion matrix for the audio model is displayed above. From the figure, it observed that the most predicted emotion label is neutral with a prediction rate of 100%. Accordingly, the model performed poorly for the other emotion labels. This is possibly because of the unbalanced nature of the emotion labels.

In the following section, the results of the baseline machine learning models are been discussed. Here, the models are evaluated using precision, F1-score, confusion matrix, and recall.

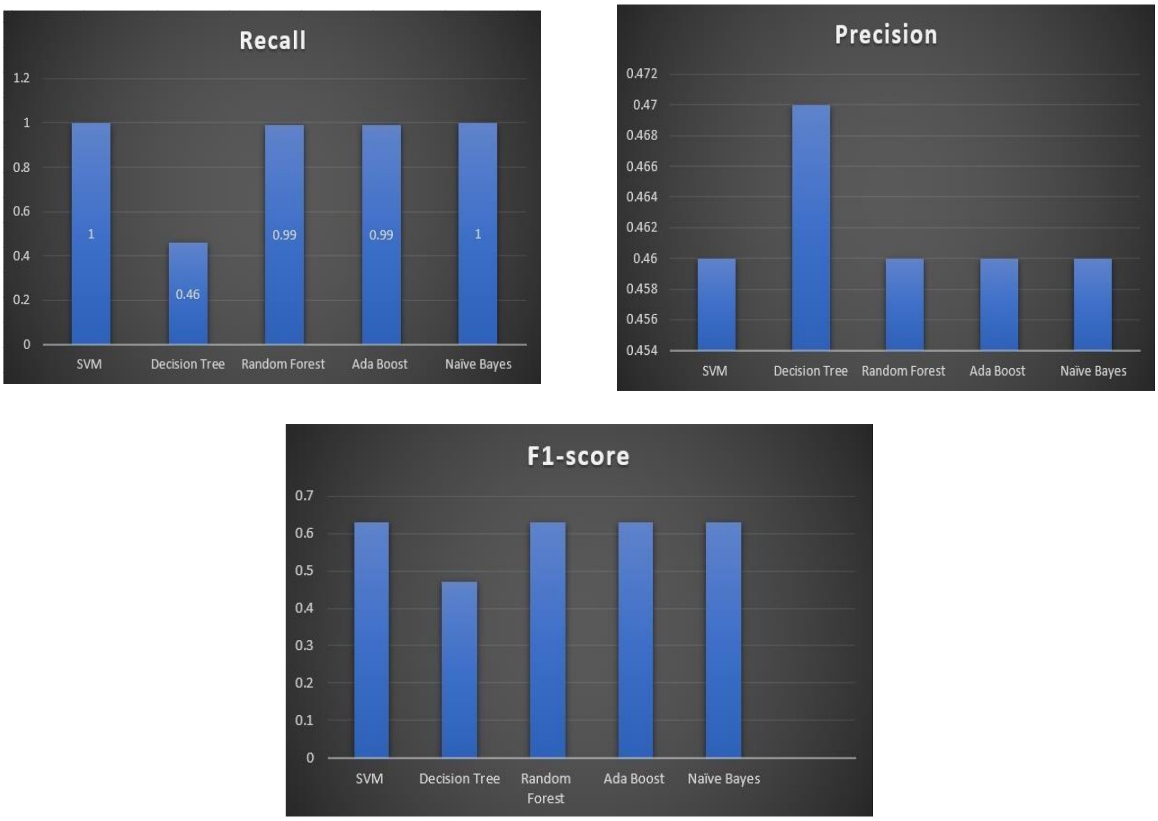


Fig 6.4.3 Precision, Recall and F1-score for baseline models

The above are the graphs for the precision, F1-score, and recall of the baseline models. It is observed that the precision and F1-score remain approximately equivalent for all the baseline models except the decision tree classifier. The decision tree classifier performed worst for our application whereas, the LSTM network outperformed other models with an accuracy of47%.

6.5 Discussion

In this section, the important findings are critically discussed. For this research work, the unimodal model performances of text and audio are been compared. For text emotion recognition, the Bi-LSTM model outperformed the other implemented model with an accuracy of 75%, whereas for audio emotion recognition the LSTM model outperformed the other model with an accuracy of 47%. Here, the same data was used for analysis but with different modalities i.e. text and audio. Data preparation for audio is done by converting the video .mp4 extension files into .wav files as these audio files are not explicitly given in the database repository. During this conversion, some of the .wav files are been corrupted resulting in no voice output because of which the content in these audio files is lost. Thus, the quality of information in audio files is affected as compared to the quality of information in text modality. This is observed to be the initial rationale for the low accuracy of the audio unimodal. Furthermore, the emotion label distribution is not balanced, as it is found that most of the observations belong to the neutral class label with a distribution percent of about 50% of the total count. Thus, the low count emotion labels like disgust and fear are misclassified in both the unimodal. Here, sampling techniques are not performed for balancing the dependent variables because the research work aims on comparison of two different modalities and thus sampling techniques will divert the findings. Also, generating more observations for low count emotions labels and maintain the quality and content information in the audio files will possibly aid to improve the model accuracy. Currently, there are only a few published studies conducted on the MELD dataset with the main aim of implementing emotion recognition using a single modality i.e. text. However, in this research work, an attempt of comparing the model performances of two modalities is carried out. Thus, considering the research question, the performed experiments conclude that there is a comparatively drastic difference between the model performances of the two different modalities models where the performance of the textual modality is more superior and accurate as compared to the audio modality.

7 Conclusion and Future Work

In this research work, a comparison of model performance is been carried out using two different modalities namely- text and audio on the same data. From text unimodal, it is observed that the Bi-LSTM model performed well in recognition the conversational emotions. Here, the textual data is pre-processed using nltk, and a word embedding vector is build using a Word2Vec. Accordingly, audio unimodal is built on the extracted MFCC features from raw audio files using an LSTM network. However, the model performance of audio unimodal is poor as compared to text unimodal with a difference of about 27% inaccuracy. This is because of the corrupted audio files with the loss of information in it. Thus, these findings are completely satisfying the research question.

In this research, two distinct unimodal models are implemented, thus, future attempts can be made to concatenate these two unimodal to combine the results. Accordingly, this will ultimately aid to improve the accuracy and robustness with a combined output as in case of a corrupted information of one modality can be averaged by the same information from another modality.

References

AHMAD, S., ASGHAR, M. Z., ALOTAIBI, F. M. & KHAN, S., 2020. Classification of Poetry Text Into the Emotional States Using Deep Learning Technique. *IEEE,* Volume 8, pp. 73865-73878, doi: 10.1109/ACCESS.2020.2987842.

Al-Omari, H., Abdullah, M. A. & Shaikh, S., 2020. EmoDet2: Emotion Detection in English Textual Dialogue using BERT and BiLSTM Models. *International Conference on Information and Communication Systems (ICICS)* , pp. 226-232, doi: 0.1109/ICICS49469.2020.239539.

Atmaja, B. T. & Akagi, M., 2019. Speech Emotion Recognition Based on Speech Segment Using LSTM with Attention Model. *IEEE International Conference on Signals and Systems,* pp. 40-44.

Azmin, S. & Dhar, K., 2019. Emotion Detection from Bangla Text Corpus Using Naïve Bayes Classifier. *International Conference on Electrical Information and Communication Technology (EICT),* pp. 1-5.

BATBAATAR, E., LI, M. & RYU, K. H., 2019. Semantic-Emotion Neural Network for Emotion Recognition From Text. *IEEE,* 7(1), pp. 111866-111879, doi: 10.1109/ACCESS.2019.2934529.

Deshmukh, G., Gaonkar, A., Golwalkar, G. & Kulkarni, S., 2019. Speech based Emotion Recognition using Machine Learning. *International Conference on Computing Methodologies and Communication* , pp. 812-817.

D, H. et al., 2019. Design and Evaluation of Speech based Emotion Recognition System using Support Vector Machines. *IEEE,* pp. 1-4.

Gürcan, F., 2018. Multi-Class Classification of Turkish Texts with Machine Learning Algorithms. *IEEE,* pp. 1-5.

Huang, K.-Y.et al., 2019. SPEECH EMOTION RECOGNITION USING DEEP NEURAL NETWORK CONSIDERING VERBAL AND NONVERBAL SPEECH SOUNDS. *IEEE,* pp. 5866-5870.

Iqbal, A. & Barua, K., 2019. A Real-time Emotion Recognition from Speech using Gradient Boosting. *International Conference on Electrical, Computer and Communication Engineering (ECCE),* pp. 1-5.

Jin, X. & Xu, Y., 2020. Research on the Sentiment Analysis Based on Machine Learning and Feature Extraction Algorithm. *IEEE* , pp. 366-369.

Manamela, P. J., Manamela, M. J., Modipa, T. I. & Sefara, T. J., 2018. The Automatic Recognition of Sepedi Speech Emotions based on Machine Learning Algorithms. *IEEE,* pp. 1-7.

Suganya, S. & Charles, E. Y. A., 2019. Speech Emotion Recognition Using Deep Learning on audio recordings. *International Conference on Advances in ICT for Emerging Regions,* pp. 1-6.

WU, J.-L., HE, Y., YU, L.-C. & LAI, K. R., 2020. Identifying Emotion Labels From Psychiatric Social Texts Using a Bi-Directional LSTM-CNN Model. *IEEE,* Volume 8, pp. 66638- 66646, doi: 10.1109/ACCESS.2020.2985228.

Yoon, S., Byun, S., Dey, S. & Jung, K., 2019. SPEECH EMOTION RECOGNITION USING MULTI-HOP ATTENTION MECHANISM. *IEEE,* pp. 2822-2826.

Zheng, J., 2019. A Novel Computer-Aided Emotion Recognition of Text Method Based on WordEmbedding and BiLSTM. *International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM),* pp. 176-180, doi: 10.1109/AIAM48774.2019.00042.