# Speech Emotion Recognition

Hima Siva Kalyan

June 2020

## 1 Abstract

Speech emotion recognition(SER) is one of the most challenging tasks in speech signal analysis domain and it heavily depends on hand-engineered acoustic features, which are typically crafted to echo human perception of speech signals. It is a research area problem which tries to infer the emotion from the speech signals. In recent times, Speech emotion recognition has gained lot of attention due to availability of high computation capabilities of Human Machine Interfaces(HMI). Also, the importance of emotion recognition is getting popular with improving user experience and the engagement of Voice User Interfaces (VUIs). For example, customer services, recommending systems, and healthcare applications. There are many ways proposed to identify the emotion in the speech, all of which require proper classification methods, selection of suitable feature sets and finding proper data sets too.

## 2 Introduction

Now-a-days, HMI technology is used by almost all industrial organisations and also a wide range of companies, to interact with their machines and optimize their industrial processes. The most common roles that interact with HMIs are operators, system integrators, and engineers, particularly control system engineers. HMIs are essential resources for these professionals, who use them to review and monitor processes, diagnose problems, and visualize data. Because of this, recognition of human emotion by computer has been an active research area for the past few years. There's a need to make the Human emotion recognition system more efficient to make the interactions between the machine and human more natural. Recognition of emotion in speech helps recognize human emotion in most efficient manner. This is called SER. The variations in prosodic parameters like contrasts in pitch, duration between successive segments etc play a major role in Speech emotion recognition. Prosodic features provide a reliable indication of the emotion. Other parameters like Utterance intensity, F0 contour, voice quality also are important.
Although, there is a significant improvement in speech recognition but still researcher are away from natural interplay between computer and human, since

computer is not capable of understanding human emotional state. The recognition of emotional speech aims to recognize the emotional condition of individual utterer by applying his/her voice automatically. It is also important that the system recognizes the emotion correctly irrespective of the subjects cultural background,language, race etc. The audio features including prosodic, MFCC and format frequency are extracted from the speech to map the emotional speech to corresponding feature space.
Other techniques like Emotional conversion, where the parameters of input speech emotion are analysed and manipulated to the target emotion also plays an important role in many applications like improving performance of Speech recognition systems, AI systems etc. Though models like GMM(Gaussian Mixture Model) based spectral conversion are applied to emotion conversion. It was found that spectral transformation alone was insufficient to convey the target emotion.
In this project, we are building and training simple Speech Emotion Recognizer that predicts human emotions from audio files using Python, Sci-kit learn, librosa, and Keras. Firstly, we will load the data (RAVDESS dataset), extract features (MFCC) from it, and split it into training and testing sets. Then, we will initialize two models (MLP and LSTM) as emotion classifiers and train them. Finally, we will calculate the accuracy of our models.

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os # to use operating system dependent functionality
import librosa # to extract speech features
import wave # read and write WAV files
import matplotlib.pyplot as plt # to generate the visualizations

# MLP Classifier
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score

# LSTM Classifier
import keras
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import *
from keras.optimizers import rmsprop
```

## 2.1 Speech Database

There are various speech databases used so far for SER models.Many researchers in this field created databases for speech emotion recognition. Nevertheless, the number of public databases is low. Among all, Berlin Emotional database and AIBO are most commonly used. In German database, 5 male and 5 female actors have participated in providing the dataset by reading a given sentence. Whereas the emotions recorded were - anger, fear, disgust, happiness, sadness, neutral. AIBO was collected in real conditions by interacting and playing of fifty one children with sonys robot aibo governed by human operator. In this

too, there were five emotions recorded - anger, rest, emphatic, positive, neutral. There are also other commonly used databases like BHUDES, VAM, Audiovisual Thai Emotional Database.
In this project, we used Speech audio-only files (16bit, 48kHz .wav) from the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset. This portion of the RAVDESS contains 1440 files: 60 trials per actor x 24 actors = 1440. The RAVDESS contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. Speech emotions includes calm, happy, sad, angry, fearful, surprise, and disgust expressions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression. We should also make sure of converting the dataset into a np array which fits the model we are using.

```
##### load radvess speech data #####
radvess_speech_labels = [] # to save extracted label/file
ravdess_speech_data = [] # to save extracted features/file
for dirname, _, filenames in os.walk('/kaggle/input/ravdess-emotional-speech-audio/'):
    for filename in filenames:
        #print(os.path.join(dirname, filename))
        radvess_speech_labels.append(int(filename[7:8]) - 1) # the index 7 and 8 of the fil
e name represent the emotion label
        wav_file_name = os.path.join(dirname, filename)
        ravdess_speech_data.append(extract_mfcc(wav_file_name)) # extract MFCC features/fil
e
```

And also, it is to be noted that the dataset is split into 80 - 20 percent of training set, test set. Train-test-split function will split arrays or matrices into random train and test subsets, in our example, training set 80 percent and testing set 20 percent.
x-train,x-test,y-train,y-test= train-test-split(np.array(ravdess-speech-data-array),labels-categorical, test-size=0.20, random-state=9)

## 2.2 Feature Extraction

Various features have been used in recent years to achieve improvements in Human emotion recognition. They are Mel-frequency cepstral coefficients(MFCC), prosodic features, Linear predictive cepstral coefficients(LPCC). In this project, we are extracting the features using MFCC. The sounds generated by a human are filtered by the shape of the vocal tract including tongue, teeth etc. This shape determines what sound comes out. If we can determine the shape accurately, this should give us an accurate representation of the phoneme being produced. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the job of MFCCs is to accurately represent this envelope. MFCC has features like simple calculations, good ability of distinction and anti-noise too.

```
def extract_mfcc(wav_file_name):
    #This function extracts mfcc features and obtain the mean of each dimension
    #Input : path_to_wav_file
    #Output: mfcc_features'''
    y, sr = librosa.load(wav_file_name)
    mfccs = np.mean(librosa.feature.mfcc(y=y, sr=sr, n_mfcc=40).T,axis=0)

    return mfccs
```

## 2.3 Classification Approaches

Here we used two different classifiers which resulted in two different accuracy , they are MLP classifier and LSTM classifier.
A multilayer perceptron (MLP) is a deep, artificial neural network. It is composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP.
model=MLPClassifier(alpha=0.01, batch-size=256, epsilon=1e-08, hidden-layer-sizes=(300,), learning-rate='adaptive', max-iter=500)
Recurrent neural networks, of which LSTMs are the most powerful and well known subset, are a type of artificial neural network designed to recognize patterns in sequences of data, such as text, genomes, handwriting and the spoken word. Here, apart from test and train sets, we also create a validation set.

```
# Split the training, validating, and testing sets
number_of_samples = ravdess_speech_data_array.shape[0]
training_samples = int(number_of_samples * 0.8)
validation_samples = int(number_of_samples * 0.1)
test_samples = int(number_of_samples * 0.1)
```

```
# Define the LSTM model
def create_model_LSTM():
    model = Sequential()
    model.add(LSTM(128, return_sequences=False, input_shape=(40, 1)))
    model.add(Dense(64))
    model.add(Dropout(0.4))
    model.add(Activation('relu'))
    model.add(Dense(32))
    model.add(Dropout(0.4))
    model.add(Activation('relu'))
    model.add(Dense(8))
    model.add(Activation('softmax'))

    # Configures the model for training
    model.compile(loss='categorical_crossentropy', optimizer='Adam', metrics=['accuracy'])
    return model
```

# 3 Training both the models

We thereby trained both the models.
Train using MLP model
MLPClassifier(activation='relu', alpha=0.01, batch-size=256, beta-1=0.9,
beta-2=0.999, early-stopping=False, epsilon=1e-08,
hidden-layer-sizes=(300,), learning-rate='adaptive',
learning-rate-init=0.001, max-fun=15000, max-iter=500,
momentum=0.9, n-iter-no-change=10, nesterovs-momentum=True,
power-t=0.5, random-state=None, shuffle=True, solver='adam',
tol=0.0001, validation-fraction=0.1, verbose=False, warm-start=False)
Train using LSTM model
model-A = create-model-LSTM() history = model-A.fit(np.expand-dims(ravdess-speech-data-array[:training-samples],-1), labels-categorical[:training-samples],
validation-data=(np.expand-dims(ravdess-speech-data-array[training-samples:training-samples+validation-samples], -1),
labels-categorical[training-samples:training-samples+validation-samples]), epochs=100,
shuffle=True)

We got an accuracy of around 55 percent for MLP model and 70 percent for LSTM model.
Now that we have models with fairly good amount of accuracy, we moved on to the next part of the project - To convert neutral emotional speech to a target emotion.

# 4 Neutral to Target emotion conversion

Firstly we need a dataset with speeches of neutral emotion to be converted to a target emotion. Though there are many datasets with neutral emotional speeches, it'd be difficult to extract separately all the neutral emotional speeches from such large datasets. Thus we used 'Librispeech' dataset. LibriSpeech is a corpus of approximately 1000 hours of 16kHz read English speech, prepared by Vassil Panayotov with the assistance of Daniel Povey. The data is derived from read audiobooks from the LibriVox project, and has been carefully segmented and aligned. Acoustic models, trained on this data set, are available at kaldi-asr.org and language models, suitable for evaluation can be found at http://www.openslr.org/11/. Now to convert the neutral speeches to target emotional speech, we used the software named DAVID. DAVID stands for Da Amazing Voice Inflection Device, DAVID is a real-time voice transformation tool able to "colour" any voice recording with an emotion that wasn't intended by its speaker. DAVID was especially designed with the affective psychology and neuroscience community in mind, and aims to provide researchers with new ways to produce and control affective stimuli, both for offline listening and for real-time paradigms. Now after converting various neutral emotional speeches to various target emotions (particularly above models eight emotions), they've

been used as datasets again to be tested for accuracy. In other words, using DAVID we first converted the neutral emotional datasets to various target emotions, thereby creating a new dataset entirely. Then that dataset is used as a test dataset on models created above to get the accuracy. MLP model gave close to 45 percent accuracy and LSTM model gave close to 60 percent accuracy. The accuracies resulted were different whenever trained, they varied from 40 to 55 percent in MLP model and 55 to 65 percent for LSTM model, thus the above mentioned accuracies are averages of the both models trained for 10 times. Though the mentors have suggested us to create a GAN based model, due to time and system requirement insufficiency, we were unable to create one and is hence left as future work.

# 5 Conclusion

- As SER is one of the most important emerging fields, and also a technique which might of great use in many areas, it is important to build a good model for accurate recognition of emotion in speech. Though the above built models are not so efficient in terms of accuracy, they can be improved by combining with other models like HMM, CNN. As we can observe that the MLP model gave accuracy of 55 percent for RAVDESS test set and 45 percent for dataset created using DAVID, we can also assume that the DAVID software conversion of neutral to target emotion can be further improved and that increases accuracies of our model too. And lastly, GAN is a model which is devised recently compared to other models, thus it is still in an upcoming / developing stage and in future can be combined with our models to increase their efficiency too.