Human emotion identification from speech

Jaid Jashim

Registration No:
2015 33 1514

12th March, 2021

—

Artificial Intelligence

—

Dr Md Forhad Rabbi

PROBLEM

Understanding Human emotion not only from varies *saved audio* files but also *from live speech*.

Basic idea behind this tool is to build and train/test using Artificial Intelligence algorithms including deep learning that will recognize and detects human emotions from speech.

SOLUTION STEPS

1. First, install the necessary dependencies/ packages using pip or conda to for the project

   - librosa==0.6.3
   - numpy
   - pandas
   - soundfile==0.9.0
   - wave
   - sklearn
   - tqdm==4.28.1
   - matplotlib==2.2.3
   - pyaudio==0.2.11
   - ffmpeg (optional): used if you want to add more sample audio by converting to 16000Hz sample rate and mono channel which is provided in **convert_wavs.py**

     Install these libraries by the following command:
     pip3 install -r **requirements.txt**

     and also make necessary imports like librosa, sound file, etc.

2. Four different datasets (including this repo's custom dataset) are used in this project which are downloaded and formatted already in **data** *folder*

- **RAVDESS** : The **R**yson **A**udio-**V**isual **D**atabase of **E**motional **S**peech and **S**ong that contains 24 actors (12 male, 12 female), vocalizing two lexically-matched statements in a neutral North American accent.

- **TESS** : **T**oronto **E**motional **S**peech **S**et that was modeled on the Northwestern University Auditory Test No. 6 (NU-6; Tillman & Carhart, 1966). A set of 200 target words were spoken in the carrier phrase "Say the word _____' by two actresses (aged 26 and 64 years).

- **EMO-DB** : As a part of the DFG funded research project SE462/3-1 in 1997 and 1999 we recorded a database of emotional utterances spoken by actors. The recordings took place in the anechoic chamber of the Technical University Berlin, department of Technical Acoustics. Director of the project was Prof. Dr. W. Sendlmeier, Technical University of Berlin, Institute of Speech and Communication, department of communication

science. Members of the project were mainly Felix Burkhardt, Miriam Kienast, Astrid Paeschke and Benjamin Weiss.

- **Custom** : Some unbalanced noisy *dataset that is located in* **data/train-custom** for training and data/test-custom for testing in which you can add/remove recording samples easily by converting the raw audio to 16000 sample rate, mono channel (this is provided in create_wavs.py script in convert audio(audio_path) method which requires ffmpeg to be installed and in *PATH*) and adding the emotion to the end of audio file name separated with '_' (e.g "20190616_125714_happy.wav" will be parsed automatically as happy)

3. Feature extraction :
   It is time to get in the coding part, Feature extraction is the main part of the speech emotion recognition system. It is basically accomplished by changing the speech waveform to a form of parametric representation at a relatively lesser data rate.
   In this project, we have used the most used features that are available in librosa library including:

   - MFCC
   - Chromagram
   - MEL Spectrogram Frequency (mel)
   - Contrast
   - Tonnetz (tonal centroid features)

4. Emotions available :
   There are 9 emotions available: "neutral", "calm", "happy" "sad", "angry", "fear", "disgust", "ps" (pleasant surprise) and "boredom".

5. Grid search results are already provided in **grid** *folder*, but if you want to tune various grid search parameters in parameters.py, you can run the script grid_search.py by : **python grid_search.py**

6. Now train the model.

7. The model is ready, run and watch the result.

8. We can also check the accuracy of the model

9. Confusion Matrix :

```
print(deeprec.confusion_matrix(percentage=True,
labeled=True))
```

10. Now, You can test your own voice by executing the following command: Wait until "Please talk" prompt is appeared, then you can start talking, and the model will automatically detects your emotion when you stop (talking) *python* **test.py**

11. **Plotting Histograms :**

This will only work if grid search is performed.

```
from emotion_recognition import plot_histograms
# plot histograms on different classifiers
 plot_histograms(classifiers=True)
```

****Python codes, Input Sample, Output Sample are all included within the zip file provided along with other necessary folders**