Spoken Language Processing - Lab2 System Description

Anonymous submission to INTERSPEECH 2023

Abstract

The spoken language identification challenge describes the efforts in classifying the original language of a spoken record in audio files. It is very relevant in today's world due to the big importance of speech processing. It has applicability in tasks such as speech recognition, language translation and audio assistency. The approaches described in this paper use audio files from six distinct languages, each containing a large collection of spoken audio. In the first part of the project, we implemented key computations to the baseline feature extraction process such as the Shifted Delta Cepstrum, SDC, and the Voice Activity Detection, VAD. We were able to reach accuracy levels of 0.51 without the usage of pre-trained models. In the second part, we tried to leverage, to no avail, the usage of neural networks like MLP and KNN to achieve an some accuracy. Overall, the pre-trained models analysed and performed in the second part reach a higher levels of accuracy as it was initially expected.

1. Introduction

The following system description covers the two systems as implemented for **Part I** and **Part II** of the second laboratory assignment from the Spoken Language Processing course, by Group #9 (Afonso Araújo - 96138, Santiago Quintas - 93179). The goal of this laboratory was to give the students a better grasp of how to train a neural network on audio files from 6 different languages - Basque, Catalan, English, Galician, Portuguese and Spanish.

2. Part I

Part I of the laboratory assignment consisted of taking a certain amount of MFCC, Mel Frequency Cepstral Coefficients, features and applying different techniques to it in order to increase the accuracy performance of our model. To achieve this we implemented four crucial functions. One for the computation of deltas that adds temporal information to the extracted features, improving the overall representation of the audio signal. the second function computes the Shifted Delta Cepstrum, SDC. It adds more comprehensiveness to the representation signal through spectral dynamics. The third function is for Voice Activity Detection, VAD, and it helps handling noise and increasing the efficiency of the algorithm by removing unwanted segments from the audio signal. The last function normalizes the features from the audio signals by computing the mean and variance.

2.1. Delta Computation

The purpose of this function is to create a vector of delta impulses from the MFCC's vector. This function takes as parameters to note the δ_{order} (n such that, ${\delta_{features}}^n$) and the *"keep_static"* parameter,

$$n = 0 \implies \delta_{features} = \text{MFCC}_{\text{features}} \tag{1}$$

$$n = 1 \implies \delta_{features} = \text{MFCC}_{\text{features}} + \frac{\partial}{\partial \text{MFCC}_{\text{features}}} \tag{2}$$

$$n = 2 \implies \delta_{features} = \text{MFCC}_{\text{features}} + \frac{\partial}{\partial \text{MFCC}_{\text{features}}} + \frac{\partial^2}{\partial {\text{MFCC}_{\text{features}}}^2} \tag{3}$$

where $\delta_{order} = n$ dictates whether or not to append the 1^{st} and 2^{nd} order derivatives of the $\text{MFCC}_{\text{features}}$ to $\delta_{features}$,

$$\textit{keep\_static} = \text{True} \implies \delta_{features} = \delta_{features} + \text{MFCC}_{\text{features}} \tag{4}$$

$$\textit{keep\_static} = \text{False} \implies \delta_{features} = \text{Void} \tag{5}$$

and *keep_static* is responsible for appending or not the $\text{MFCC}_{\text{features}}$ vector to $\delta_{features}$.

2.2. Shifted Delta Cepstrum

For the shifted delta cepstrum, the goal was to create an SDC matrix, $\text{SDC}_{\text{features}}$, from the $\text{MFCC}_{\text{features}}$ vector, such that its dimension took into consideration the number of frames and features per frame of the original $\text{MFCC}_{\text{features}}$ matrix.

$$\text{SDC}_{\text{features}}[i][j] = \delta_{\text{features}}[i + z] \tag{6}$$

Here, j represents the parsing through the i-th column of $\text{SDC}_{\text{features}}$ from dim elements to dim elements at a time, where

$$\text{dim} = \text{MFCC}_{\text{features}}.\text{shape}[1] \tag{7}$$

which is the number of columns, frame features, that $\text{MFCC}_{\text{features}}$ has. From Equation 6, z represents the parsing ratio at which δ_{features} is scanned, such that

$$z = K \times P \tag{8}$$

Here, K represents the number of intervals or segments, and P represents the interval for selecting previous deltas.

In this way there is this macro-equation that defines this function:

$$\text{SDC}_{\text{features}}[i][K \times dim : (K+1) \times dim] = \delta_{\text{features}}[i + K \times P] \tag{9}$$

However, a problem still surfaces should the dimensions of δ_{features} do not take into account the fact that, while being parsed and to preserve the structural integrity of $\text{SDC}_{\text{features}}$, there may be some values out of reach of the matrix. Hence the need for a zero padding before applying 9,

$$\delta_{features} = \text{zpadding}(\delta_{features}) \tag{10}$$

2.3. Voice Activity Detection

For the **Voice Activity Detection** removal subsystem, the group went for a RMS implementation.

This mean that,

$$\text{energy}_{\text{features}}(i) < 0.025 \longrightarrow \text{MFCC}_{\text{features}}(i) = \text{False} \quad (11)$$

filtering a new vad_{vector} and its labels. The choice for 0.025 for threshold value came for trial and error, being the one with the better accuracy, as indicated in the table below.

2.4. Cepstral Mean and Variance Normalization

In this function we compute the mean and variance of the extracted features along the time axis. When the variance is equal to zero, we subtract the mean value to the features. When the variance is different than zero we subtract the mean value and divide the result by the square root of the variance.

3. Results

For **Part I** the group extracted accuracy of the model for two different instances - the first one when only applying VAD and CMVN, Cepstral Mean and Variance Normalization, and the second one adding on the SDC computation:

Without SDC	33%
With SDC	52%

For **Part II** the group did not have time to implement a working algorithm able to reach high levels accuracy. However, our ideal implementation would revolve around a KNN for 5 neighbours. The group was also considering MLP and SVM but due to time constraints the KNN, $5 - NN$, was the chosen model, as it required less computing power.

4. Conclusions

For this spoken language identification challenge, we implemented two methods to try and achieve a high level of classification accuracy. One using pre-trained models and another without. Without pre-training, our application reached a 0.51 accuracy. Unfortunately, we were not able to write a functioning code for the second part. Nonetheless, our approach focuses on a KNN implementation using 5 neighbours. Throughout the project we obtained extensive knowledge about methods of speech analysis and classification.