Automatic Speech Recognition

João Borges
Telecommunications, Automation and Electronics
Research and Development Center (LASSE)
Federal University of Pará
Belém, Brazil
joao.tavares.borges@itec.ufpa.br

***Abstract*—In this work, the Mel-frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) algorithms are implemented in Python to develop an isolated word recognition pipeline. The system currently shows 60% accuracy in word identification when a built-in MFCC function is used, and 40% when an own implementation is utilized instead.**

***Index Terms*—MFCC, DTW, Speech Recognition**

I. Introduction

The use of Mel-frequency Cepstral Coefficients (MFCC) to extract audio feature and Dynamic Time Warping (DTW) to compare the obtained values is a classical method of Automatic Speech Recognition (ASR). This method leverages the Mel frequency scale, that is designed to be a perceptually relevant scale for pitch, meaning that equal frequency distances have the same perceptual difference from the point of view of a human listener.

II. Data Processing Pipeline

This work uses a dataset composed of 105,000 WAVE audio files, with 30 different classes, obtained from Tensorflow[1], from which a subset of 5 classes will be used in the experiments. The dataset is divided into 80% for training and 20% for validation. The remaining process can be split into 3 steps:

- Calculate MFCC for all signals
- Perform DTW between sample and reference signals
- Obtain the mean of the DTW values, which is the recognition rate

In the following subsections each of these steps will be described in details.

A. Obtaining the MFCCs

The MFCCs of all audio files are calculated with the following steps:

- The signal goes through a process of Short-Time Fourier Transform (STFT)
- After that, the Mel filter banks are calculated
- Then, a dot product between the previous results, followed by a dB conversion, are performed to obtain the melspectrogram
- Finally, the MFCC is obtained by applying Discrete Cosine Transform (DCT) to the melspectrogram

[1]https://storage.cloud.google.com/download.tensorflow.org/data/speech_commands_v0.02.tar.gz

1) Short-Time Fourier Transform Step: Due to the highly non stationary nature of speech data, composed of different phonemes with their own frequencies distributed along the time, it is necessary to be capable of visualizing how the signal spectrum changes over time, a feature that is lacking in the traditional Fourier transform. This motivates the method known as STFT, that divides the signal into frames through a process of windowing and then applies the Fourier transform in each one of them. These frames usually need to be overlapped in order to avoid loss of signal. In the current implementation, the STFT adopts the widely used *Hann* window for the windowing process.

2) Mel filter banks: To calculate the Mel filter banks, first is necessary to determine the number of Mel bands that will be used, then obtain the lowest and highest frequencies of the signal in the Mel scale. After that the interval between the lowest and the highest frequencies must be divided into a number of points equal to the number of Mel bands, evenly separated. These points are then converted back to the Hertz scale and rounded to the nearest bin. Finally, the triangular filters are created [Improve]

3) The melspectrogram and the MFCC: In order to obtain the melspectrogram first we need the spectrogram, which is obtained by the magnitude squared of the STFT result. After that, a dot product is executed between the spectrogram and the Mel filter banks, followed by a power to dB conversion, resulting in the melspectrogram. To

B. Performing the DTW

C. Evaluating the system performance

In this step, a process of cross-validation is implemented. *Leave P out* cross-validation is [...] a The code is available publicly[2]

Fig. 1. Example of a figure caption.

[2]https://github.com/joaotavares43/asr-jupyter