Nikole Giovannone

Final Project Writeup

The goal of my project was to create a function that could determine whether there is a pause in an audio file. To do so, I gathered intensity and fundamental frequency information from a set of 100 audio files which each contained one sentence. 79 of the sentences contained mid-sentence pauses, and the remaining 21 contained no such pauses. The frequency and intensity information was extracted using the “Energy” and “f0 (straight)” measures in VoiceSauce, a toolbox for Matlab. This information was then saved to a spreadsheet (**output.xlsx**), which contained a row holding the information for each sentence. I also appended a second sheet to this file which contained a list of the sentence names as well as a binary 0/1 distinction of whether the sentence had a pause or not, with 0 meaning no pause and 1 meaning pause. This sheet served as an answer key with which to test the accuracy of the function.

This function (**pause\_detector.m**) was designed to work both to test different intensity and pause duration values against each other to find the optimal combination for pause identification, as well as return a final accuracy check when only one intensity value and pause duration value was provided. The function takes 9 input variables:

* alldata: the sheet containing pitch and intensity information from my sentences
* sentence\_data: the sheet containing the sentence names and whether they have pauses or not
* test\_sentences: a sampling of 50 sentences in the directory
* lowerthreshold: the lowest dB value to test
* upperthreshold: the highest dB value to test
* thresholdlength: the number of dB values to test
* lowerduration: the smallest possible duration to test
* upperduration: the largest possible duration to test
* durationlength: the number of duration values to test

The function first sets up a matrix which matches each possible intensity value within the range provided with each possible pause duration value. The matrix size would be thresholdlength\*durationlength x 2 in size. For each of these pairs of values, the function then tests how many sentences are identified as having a pause.

One challenge faced while writing this function was how to make sure that it was truly identifying a pause, rather than other moments of silence (such as silence at the beginnings or ends of sentence recordings). For this reason, I chose to begin analyzing the intensity information for each sentence between 1 second post-onset and 1 second pre-finish. In addition, my function identified which samples were at below the intensity being tested, but did not account for whether these samples were consecutive or not. For this reason, I had to write additional code to extract the longest consecutive stretch of samples identified at or below the intensity being tested.

I ran this function 3 times with different values for each parameter (**final592b.m**). The first time, I tested intensity values between 0 and 0.1dB with steps of 0.002dB and duration lengths between 25 and 75 samples with steps of 1 sample. The results of this run led me to constrain the ranges to 0-0.005dB (steps of 0.001dB) and 25-50 samples (steps of 1 sample). For the third run, I further constrained the intensity range to 0-0.001dB (steps of 0.0001dB), but left the duration length range as it was for the second run. These three runs led me to choose 0dB and 35 samples as the optimal measures for pause detection. When these values were used to run the function on the remaining sentences on which the training was not conducted, the function returned a 96% accuracy rate.

One thing that I could have also done with this function is used the pitch information to aid in pause identification. I could have assessed the pitch leading up to the longest consecutive stretch of samples at or below the intensity being tested. In some cases, this would have been a pause, but in others (mostly the non-pause sentences), it would not. I could assess the pitch information in the 100 samples (100ms) leading up to this stretch and find out if there is some kind of linear trend that corresponds to the pitch before the pauses in this random sampling. Then, the identified trend could be used to test whether the remaining sentences have a pause. This, coupled with the previous detection method that is reliant on the duration and intensity of the pause, could raise the accuracy rate of the function.

The ultimate goal of this function is to be able to identify a pause on other stimuli for which there is no record of the correct answers. Seeing as this function was trained on a small set of very similar stimuli, it may not be successful in identifying pauses in other kinds of stimuli. In the future, training the function on many diverse stimuli could be helpful in writing a more accurate function for general use.