TECHNICAL/USER MANUAL FOR CEBUANO SPEECH TUTOR

The Cebuano Speech Tutor Team using a Linux dual boot system for the entirety of this project. The team does not guarantee the same results for future users that use linux virtual machines, or users that use WSL/WSL2. Below is the installation/changes necessary to get the project to barely run.

Note: If future user is reading this 2+ years from now on, you may try and contact the team responsible for this project, but expect them to have forgotten 80% of the configuration of this project

1.) Download and Install Kaldi ASR from Github (read https://kaldi-asr.org/doc/install.html)
2.) Download TDNN-HMM-Hybrid Model (https://github.com/ChristianIbaoc/TDNN-HMM-Hybrid-Cebuano)
3.) Create a new folder (might be easier to name this folder phoneTest for consistency purposes but not necessary)
4.) LEARN How to activate bash scripts through the terminal.
5.) The folder and files are already working HOWEVER one might be inclined to change directory paths in necessary files:
    a.) Mfcc.conf
    b.) run.sh
    c.) GRANDRUN.sh
6.) Create a folder link of the src folder in kaldi/egs/wsj/s5 directory and place that link inside your project folder.
7.) STUDY the run.sh file, it will contain the necessary calls to specific scripts that are provided by KALDI.
8.) NONE OF THESE FILES WILL RUN IF YOU DO NOT HAVE DATA.
9.) Notice the run.sh files contains directories to data folders that do not exist in this folder, that is because we could not add them to the repository (duh).
10.) Your data should contain 1 channel only (mono data), or upgrade to 2 but it really only adds more complexity for 0 reason.
11.) Also ask your adviser (ideally someone with speech recognition knowledge) on what possible processing should you be doing to your data. Or contact me, hopefully i'll remember (but im no expert so 0 credibility, source: trust me bro).
12.) Depending on your scope, you might be able to make use of the python scripts inside data/train and data/test named prep.py. Study them, they essentially automate the transcription process for your audio files assuming they are formatted in the following way:

a.) All audio files of the same speech or lines are in one folder
b.) The folder's name is the line, separated by underscores
c.) All audio files are the same (still the line, but not separated), with an ID at the end (1m,1f,2m,2f for example, its your discretion though)

Of course, this is not necessary, and you can manually transcribe your data if you want (reject technology, go back to monke).

13.) Data prep is necessary BEFORE running the run.sh or grandrun.sh

14.) Grandrun.sh is just run.sh running over multiple iterations of datasets that you don't see (again, because we cant upload that much to github). It might be good to follow in our footsteps in doing K-Fold testing, ie, separating your data into multiple groups, rotating all of them for training and testing. Just google it.

15.) Learn to run run.sh with a passed parameter (stage number). If you study run.sh, it is separated by stages.

16.) INSTALL CUDA! Kaldi needs Cuda to run its more sophisticated scripts, specifically for neural networks. The last stage of run.sh will not run without CUDA.

17.) Study KALDI's documentation and examples, and crosscheck it with our project to determine what we changed, although you can read our paper and determine what major changes we made.

18.) Now download the CSTW files (https://github.com/Nigel333/CSTW)

19.) This is the website part of our project. You will need sql for this, so learn how to use and setup your own sql server in linux. Also read the readme to see what you need to know. There is another run.sh file here, and it is the run.sh that is used AFTER (MAY I REPEAT THE AFTER!) the model is already created.

20.) Study the directories called to determine how they are used.

21.) Good Luck scholar!