SPELL Project Plan Spring 2022

Avinash Persaud, Nandith Narayan, Carlos Cepeda
Advisor: Dr. William Shoaff

The Problem

Vocal synthesis is the process of creating artificial human speech. Just like any other machine learning process, vocal synthesis requires a lot of data. Not only does this require data, but it also requires that data to be labeled. Manually labeling this data takes up a large amount of time and effort.

Goal and Motivation

We aim to develop a standalone and user-friendly tool to assist users with labeling singing data for vocal synthesis and machine learning purposes.

The current tools have the following issues that we are trying to remediate:

- There is no one tool that can do it all, and this leads to developers needing to rely on multiple tools for labeling singing data
- The currently existing tools are also unintuitive and not user-friendly resulting in lots of time spent learning how to use the tool

Recap of Label Types

There are multiple ways to label singing data, and the most common methods are:

- Phonemes
  - Individual consonant or vowel sounds in a language that come together to form words
- Syllables
- Notes

Our application will allow the user to label the singing data using any of these methods.

Our Approach

- Allow the user to create mono labels graphically
    - Mono labels are a special label format that contains information about the data such as:
        - Start Time
        - End Time
        - Phoneme IDentifier
- Ability to export the user generated label data in the HTS full label format
    - User will be able to export all their with the click of a button
- Ability to automatically detect which phonemes are present in the audio
    - This helps reduce the amount of manual work the user has to do
- Ability to automatically align phonemes with the audio
    - This tool will detect where each phoneme must start and end, and place the phoneme accordingly, further reducing manual work on the user's part
- Ability to automatically save the users work given an interval of time between saves
    - This will ensure that the user will less likely lose the progress they have made

Our Approach (Cont.)

- Allow the user to export an existing project
    - This allows the user to transfer a project to another computer
- Allow the user to activate common tasks by using keyboard shortcuts
    - Labeling singing data requires the user to repeat simple tasks, and shortcuts can help alleviate the manual labor required.
    - These actions include:
        - Copying
        - Pasting
        - Duplicating
        - Moving
        - Etc.

Novel features

- The ability for the user to copy and paste phonemes
  - Other tools, such as praat, do not allow the user to easily copy and paste these phonemes.
  - Plugins would also need to be used to accomplish this task in other programs
- The ability to automatically detect and align phonemes in the same tool is also novel
  - As previously stated, singing data labeling usually requires use of multiple tools

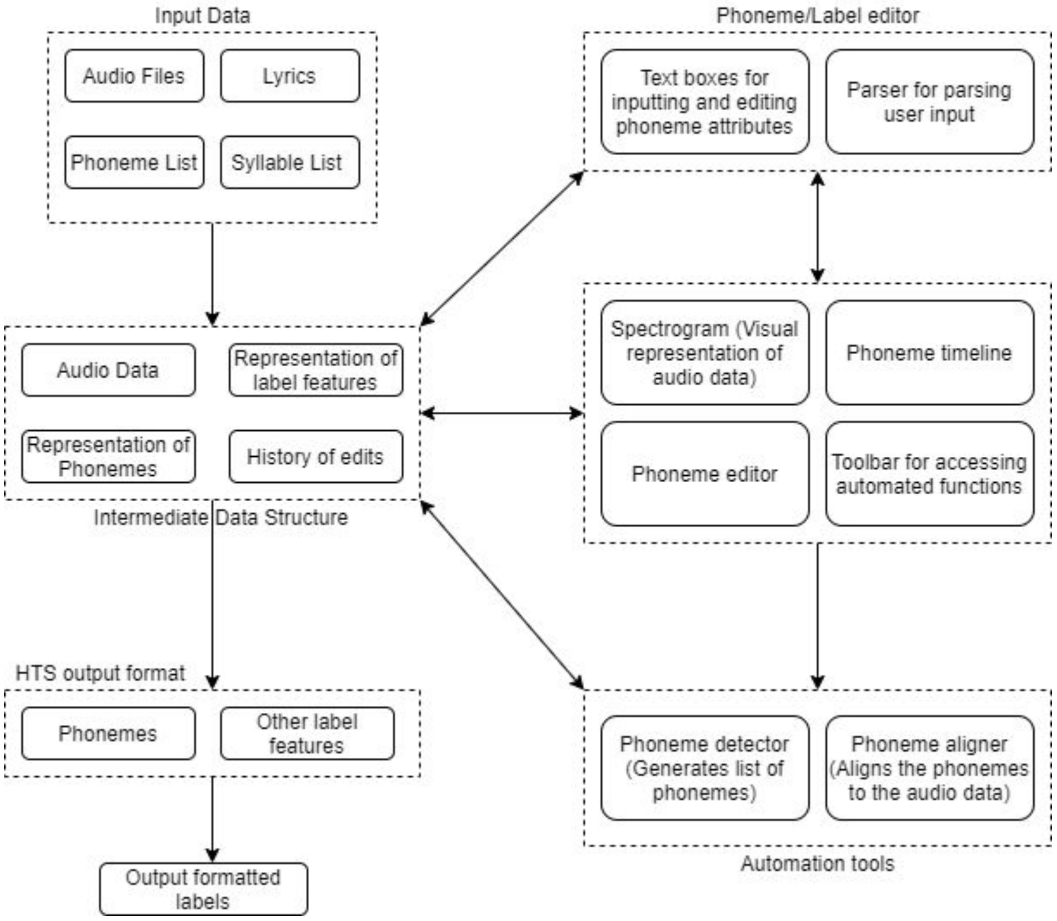
Technical Challenges

- Target output format (HTS) is challenging to interpret by eye, and it contains a lot of specifications making it challenging for us to format our output data to meet those specifications
- Automating phoneme detection posed a big enough challenge that it was split into 2 parts:
    - Phoneme Identification
        - We used a hierarchical classification model which converted the sound into MEL spectrograms and passed it through a series of CNN based classifiers
    - Phoneme Detection
        - Used to detect where the phoneme starts and ends

Technical Challenges (Cont.)

- Increasing the speed of our spectrogram generation is another challenge that we have come across during our testing
    - We are currently using Cooley-Tukey's FFT algorithm to compute the spectrogram, but our implementation seems to be too slow for high sample rate audio
- Smooth dragging of UI elements is another challenge due to the limitations of our GUI library

System Architecture Diagram

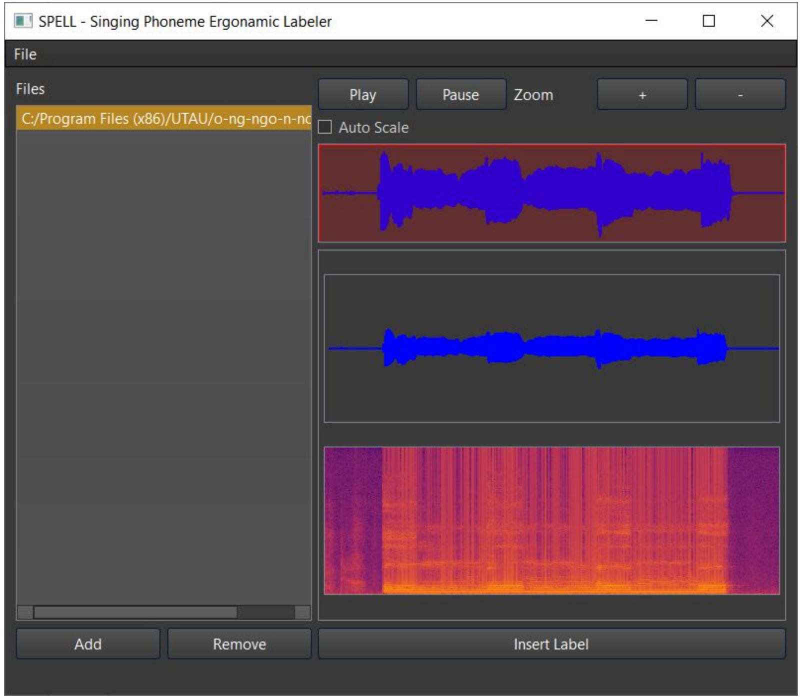

Input Data
Audio Files
Lyrics
Phoneme List
Syllable List
Phoneme/Label editor
Text boxes for inputting and editing phoneme attributes
Parser for parsing user input
Audio Data
Representation of label features
Representation of Phonemes
History of edits
Intermediate Data Structure
Spectrogram (Visual representation of audio data)
Phoneme timeline
Phoneme editor
Toolbar for accessing automated functions
HTS output format
Phonemes
Other label features
Phoneme detector (Generates list of phonemes)
Phoneme aligner (Aligns the phonemes to the audio data)
Automation tools
Output formatted labels

Evaluation

Success of our project will be measured using the following metrics:

- Fluidity of the UI
  - We will measure the fluidity of the UI by consulting the client about the usability of the application
  - We can also count the number of clicks required to complete an action
- Speed
  - We will measure speed by timing how long every action takes such as generating the spectrogram, loading a song, switching song and exporting data
- Accuracy
  - We will measure accuracy by comparing the auto generated phonemes wit the output from other tools like praat
- Reliability
  - We will measure this reliability by automating stress testing via a script to automatically add hundreds of songs ata time and perform actions like adding phonemes

Current UI layout



SPELL - Singing Phoneme Ergonamic Labeler
File
Files
C:/Program Files (x86)/UTAU/o-ng-ngo-n-nc
Play
Pause
Zoom
+
-
□ Auto Scale
Add
Remove
Insert Label

Milestone 4 (Feb 14)

- Basic phoneme feature input
- Basic project file IO
- Exporting labels to an output file
- Build test song database
- Create label output
- Add better navigation

Milestone 5 (Mar 21`)

- Complete timeline view
- Integrate phoneme boundary network
- Start note view
- Dictionary system
- Create label output

Milestone 6 (Apr 18)

- Implement, test, and demo which features/modules
  - File I/O
  - Phoneme labeling
  - Phoneme boundary detection
  - Output
  - Note Input
  - Lyric separation
- Test/demo of the entire system
- Evaluation results
- Create user/developer manual
- Create demo video

Milestone 4 Task Matrix

Task	Avinash	Nandith	Carlos
Complete intermediate data structure	50%	50%	0%
Create timeline feature widget	20%	20%	60%
Create project file format and I/O	20%	60%	20%
Create label output	50%	50%	0%
Build test song database	100%	0%	0%
Add better navigation	0%	80%	20%