Meeting Minutes
Date: 9-28-18
Attendees: Hailey Johnson, Wei Zhao, Jordan Cates, Feng Li, Eric Booth

---

Announcements:

- Ujjwal was unable to attend
- We should probably use Github or Bitbucket as they can't use Google Drive.

Q&A:

1. Can we view the individual phones as packets similar to computer networking protocols, just on a small scale?
    a. Each frame will have some overlap with other frames, which is the only real difference. The speech frames are dissected to separate the different speech features. Eric has a paper that he wrote that he will send us that deals with separating the different features of speech from an audio file. The previous year has a tool to visually represent the audio and the different features. The Mel Frequency Cepstral Coefficient, Mel Filterbank, and Discrete Cosine Transform of the filterbank energies can help us extract features. The frequency content can help you determine the different sounds.
2. Do we have a timing range we are trying to stay within?
    a. Real Time Factors involve how long it takes to decode the audio vs how long the audio file is itself. There is a table on the same paper as above that can give you info on the Real Time Factors. We should aim for a RTF of 1.
3. How many bits are within a phone? What is the %error we allow when comparing to the database samples?
    a. When you do recognition of speech, it's not always one output. It will take the sound and give you the most likely phones that it could have been. As the program traverses the audio file it will come up with phones it thinks it heard. There is a pruning bandwidth that determines how many different possible outcomes it will return. Most speech recognizers build a sort of state machine and follow the most likely path through it to determine the things that could have been said. Depending on how it goes through the graph, it will determine the percentage of likeliness.
4. Will we eventually run this on an integrated circuit such as an FPGA or microcontroller, or keep it on the computer?
    a. For usability, we probably will just want to target a computer or tablet. But if you are interested to try targeting something like FPGA's or microcontrollers then feel free to try it. In his research, some people optimized certain parts of the algorithms that took the longest.

5. What is the size of the database needed for the future? Do we have one database that is separated into samples, multiple user information including stats and scoring for therapist, or more than one database?
    a. There are two different “Databases” that are really just collections of speech called a Corpus. It’s a big list of audio samples. CSLU speech database is the biggest one he knows of. There are about 70,000 speech files from different students. It is located in their bitbucket under training and testing. We are correct in thinking that students will have individual user profiles that the therapist could put in to retrieve their past data and show how it has been developing over time. As a team, we should figure out what parts we should focus on, UI, Database, Optimization, Accuracy. The most important part of the project is collecting speech data. Focused on the specific words that are used by Speech Therapists. A tool with a UI, that prompts, takes the students input, tags it, and stores it for future use. This would be very useful.
6. For the interface, are we going to update it to be more child friendly? For example, colorful, characters speaking, countdowns, arrows, etc?
    a. Kids aren’t just a source of data and you need to keep them engaged. So this stuff is super important.
7. Would it be helpful to take the first few seconds of a session to listen to the level of background noise and “calibrate” to it?
    a. He hasn’t thought about it a whole lot. Differences in microphones and location are important. Maybe there is a more clever way to overcome this.
8. What did the previous team write? Did they edit Sphinx, create GUI?
    a. They created tools, one that integrates sphinx, that creates a prompt then uses sphinx to determine what was said. It also provides a spectrogram of the speech. The accuracy however, is very poor because it is using the default acoustic model, because it is calibrated for an adult male. They were trying to train the model last year for child speech. The google, or microsoft recognizers may be more accurate.
9. Is there any speech recognition/processing work that we could do or that all done in Sphinx?
    a. It was pretty much all Sphinx.
10. How much accuracy are we looking for?
    a. Speech is usually measured against the commonly accepted human accuracy, which is around 94 percent accuracy. Other speech recognizers are very accurate because they are adult centric, but since we are dealing with child speech, it is much trickier to deal with. This leads us to the need for more child data.
11. How is accuracy actually calculated/scored?
    a. Chapter 5 of his paper will go into that. There are quite a few metrics that deal with this. If you google Classification Rate, Precision, TP Recall, TN Recall, and F Score, as they are all metrics that are good for grading speech. Get familiar with these metrics.

12. Were the speech samples collected by the students?
    a. No they did not collect any of their own speech samples. One of the students was working with a nearby school to collect data, but it didn't end up panning out. He will email us the contact information of the previous team.
13. Were the sentence/word prompts collected/chosen by the students or were they based off of certain research?
    a. All of their prompts were just random, things that would be in children's books or something. The database we have mostly just consists of common words. He has a list of words that are commonly used in speech therapy that have certain transitions between different sounds that he will send us.
14. The previous team's Wiki involves both "Samples for testing database" and "Samples for training database" in their Specifications. What training elements does the program have/should it have?
    a. The data will be split into multiple portions. 10% goes to testing, 10% goes to development, 80% goes to training the machine learning algorithm. If you're doing a neural network you can play with the different parameters on your development set and then use them on your test set.
15. What about Python caused the program's efficiency to be so low?
    a. It shouldn't have actually taken that long so that is a thing that we could probably debug.
16. Did last group use any ML model to identify the void record or use just use sphinx-4 model.
    a. He does not have an answer for this one.
17. Do you have a suggestion of what sort of machine learning model could we use instead of Sphinx and what kind of model does Sphinx use
    a. Sphinx uses the Hidden Markov model(or something like that) that is in chapter 4 of the file that he will send us. It analyzes the speech that you already have vs the incoming speech data. He liked it for child speech because there isn't a lot of it out there. That's why he likes Sphinx. All of the big companies use neural networks. They use mass amounts of people's data to train their neural nets.
18. What could we do for the data gathering tool?
    a. It could be integrated or be a separate tool, that would be simpler. It could tag and archive all of the specific data for each individual student. It could also create a database of the information with a bunch of different details, such as microphones, setting, student age and name, words used, and be searchable to present data with certain parameters.

To Do for Next Meeting (10-1-18):

- (Everyone)
    - Make BitBucket accounts
    - Read over some of the documents that Eric is sending us

- Read over product requirements of the previous team

***Highlight important things