Team CodeSwitch

CO: Honghao Chen, Nicholas Toldalagi, Marco Yu
18-500 Capstone Design, Spring 2022
Electrical and Computer Engineering Department Carnegie Mellon University

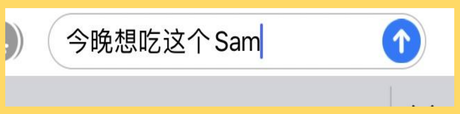
Product Pitch

Siri and voice-typing on our smartphones, the voice assistant on Amazon echo are all examples of successful ASR applications that benefit our life. Though the recognition of dozens of languages are individually supported in these applications, situations in which speakers may switch between languages mid-sentence are not specifically handled. For example, for a person native in both English and Mandarin, it is natural for them to mix some Mandarin words when they speak English, or vice versa. Yet, most ASR applications, such as voice-typing on an iPhone does not have great accuracy in understanding such mixed-language speech.

Our project aims to train a ASR speech-to-text model and create a system that is able to feedback accurate and real-time transcription of speeches mixed in English and Mandarin. Users are expected to see transcription composed of English and Mandarin words matching exactly the way they spoke propagated to their screen as they are speaking.

Product Illustration

When I spoke: 今晚想吃steak还是salmon?
Reference: Want steak or salmon for dinner?
What iPhone thought I meant:
I want to eat this Sam for dinner



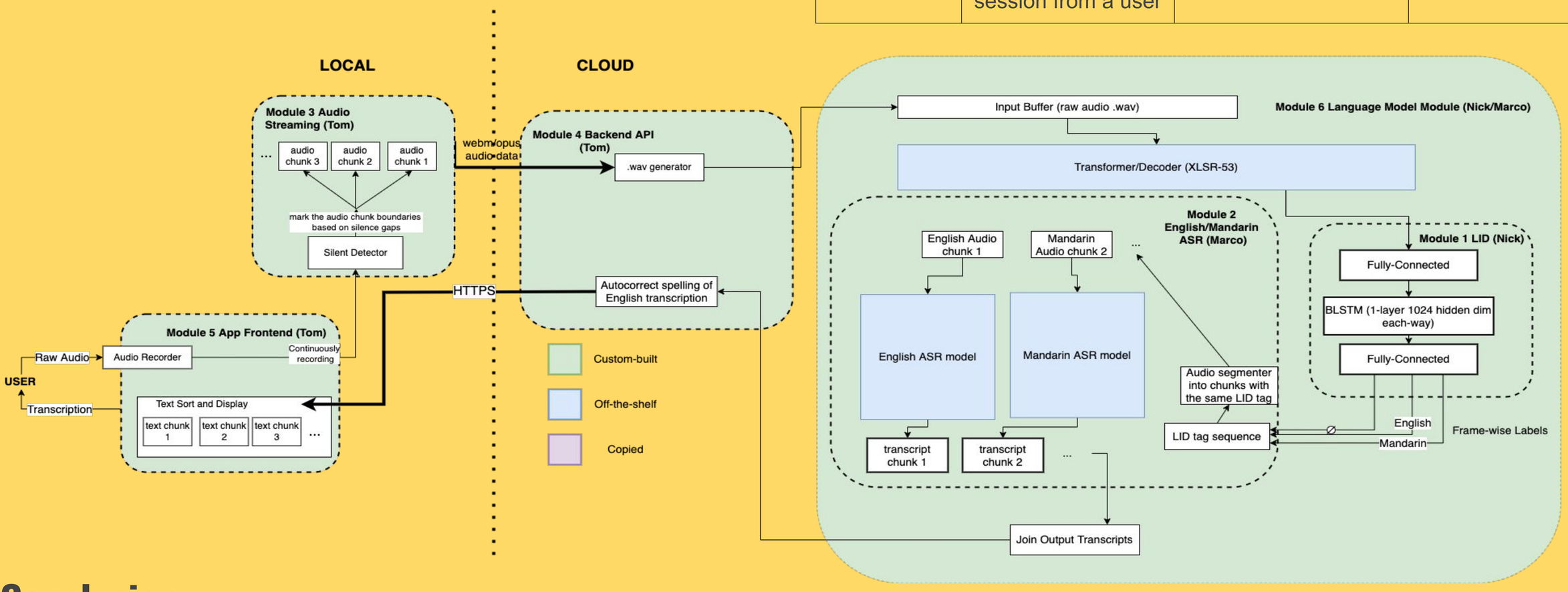
Our model output:

System Description & Architecture

1. **Language Identification (LID):** Classifies frames into English, Mandarin, or Blank
2. **Automatic Speech Recognition (ASR):** Single language ASR model transcribes each audio sequence based on its LID tags
3. **Audio Streaming:** Detect silence gaps within the ongoing recording and calls remote API to transcribe the newest chunk
4. **Backend APIs:** pre-loads model on startup, receives audio chunks, requests chunk transcriptions from model, autocorrect English sequence within predicted transcript
5. **App Frontend:** Simple, real-time I/O interface for user to record messages and receive transcriptions
6. **Language Model:** Encapsulates and combines submodels, contains all end-to-end training and validation

System Evaluation

Name	Description	Testing Procedure	Results
End-to-end Latency	Shall return a translation of the first segment in no more than 1 second	Measuring time difference between sending out a transcription request and receiving response	On average 800 ms
Throughput	Shall process audio faster than audio is inputting, lower than 1.0 s/s	Measuring average server processing time per second of audio input	0.7s for server to process 1s of audio
Character Error Rate	Shall transcribe speech below 25% CER	Testing CER on various datasets	22.94% on code-switch dataset 18.25% on dataset containing English, Mandarin, and code-switch speech
Noise Tolerance	Shall remain below 30% CER when signal to noise ratio in the audio is low	Testing on dataset with signal to noise ratio < 10dB	31.28% CER
Supporting Environment	Shall be usable on a laptop	Testing on Python local environment and through Chrome browser on MacOS and Windows	Local python script runnable on any OS Interactive mode runnable on Chrome browser
Max Data per Input	Shall be able to transcribed a maximum of 1 minute of audio in a single recording session from a user	Testing single long segment of speech longer than 1 minute	Supports 1 minute audio transcription with occasional GPU memory exceeded error



Conclusion

Overall, the system worked better than we had anticipated. There are, however, several aspects that can be significantly improved. The biggest problem we faced is the lack of data. Most machine learning algorithms are data hungry, but we've only managed to gather around 80 hours of speech. Some papers reported lower error rates using over 1500 hours of speech. Another problem we discovered was the ambiguity of filler words. Utterances such as "uh" and "hmm" in English have similar counterparts in Mandarin. This makes it difficult for the system to determine the language of the segment. Finally, the segmentation approach we used sometimes cuts away weak sounds (such as last l in school). This makes it difficult for the ASR model to fully recognize the word.

SCAN ME