How Can a Lazy Professor Give a Lecture Without Speaking?

Lusha Li, Meiyi He, Siya Li
Advisor: Andrew B. Kahng

USER-SPECIFIC SPEECH SYNTHESIZER
- Existing text-to-speech synthesizers allow input of specific scripts and return synthesized output audio
- Weaknesses of current synthesizers:
  - Non-user-specific: Machine-like voices, not natural-sounding!
  - User-specific: huge databases (> 20hr of recordings needed) and processing times for speech synthesis, painful to use!
- We develop a text-to-speech synthesizer that is Fast, User-Specific, Natural-Sounding, and Incrementally Trainable

KEY FEATURES OF OUR SYSTEM
- Maintain separate text + audio Library for each User
- Store complete phrases to minimize loss of naturalness
- Tag words and audio in the Library according to context
- Produce minimum recording requests to cover new scripts

WORKFLOW
User Interaction
- Actual Use
  - User inputs script
  - System synthesizes
  - User library updates
- Training
  - System records audio
  - Present scripts

SYSTEM INTERACTION
- Challenge #1: Given a User’s recording of a Script, how to find words and their boundaries in the audio? We solve this Alignment Problem using a tuned instance of Prosodylab-Aligner
- Challenge #2: Given a new script that is not completely covered by the User Library, what is the minimum set of sentences and words that must be recorded? We formulate this as the Minimum Set Cover problem and solve it with a greedy heuristic
- Challenge #3: Given a new script that is covered by the User Library, how can we synthesize the audio with a minimum number of pieces from the Library? We solve this with Dynamic Programming

CHALLENGES + SOLUTIONS
- The user library requires a huge data which slows down the whole system
  - Use Set Cover to minimize the data request from the user
  - (2165 sentences → 972 sentences, 44% coverage rate)
- The correctness of speech aligner is only 40% in the beginning
  - Noise of short pauses can be distracting to speech alignment
  - Our method: identify and delete inaccurate pauses → shrink the audio
  - Improved: 11 out of 2997 affecting 19 words
- Optimized the usage and the naturalness of user inputs
  - Design the way of adding tags to distinguish audios from different contexts
  - Use Dynamic Programming to minimize the number of subsequences to cover the synthesized speech
  - Below is a graph that illustrates how our library is constructed and how our program retrieve the data from the library

FUTURE WORK
- Polishing quality of synthesized audio based on learnings from “focus group” and other studies
- To develop a lightweight text-to-speech synthesizing website where users are registered and libraries are built based on user account
- To design a mobile phone application to apply the speech synthesizer for silent scenarios that people cannot talk to make phone call

ACKNOWLEDGMENTS
Many thanks to Christine Alvarado, Andrew Kahng, Aditi Mavalankar and NSF. We use Prosodylab-Aligner (Department of Linguistics, McGill University; K. Gorman, J. Howell and M. Wagner, “Prosodylab-Aligner: A Tool for Forced Alignment of Laboratory Speech.” Canadian Acoustics. 39.3. 192–193. [preprint] [paper]