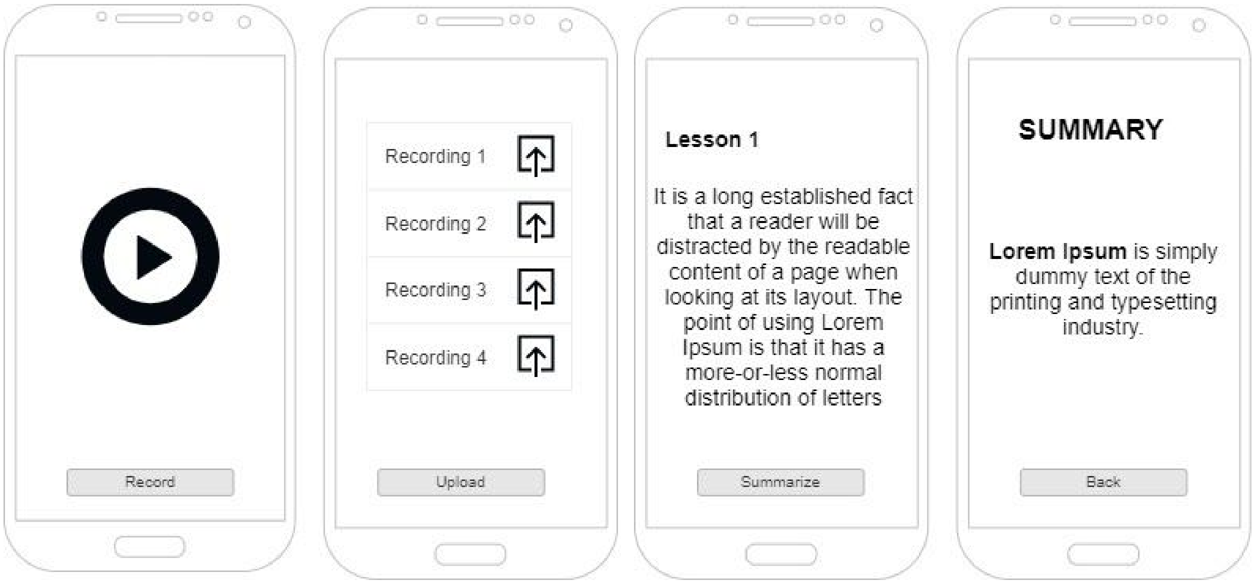
Your Notes: Speech to text API with Annotation Mechanism

2020.09.23

Annie Liu (AnnieLiu@my.unt.edu, acliu96@hotmail.com)
Yinghsuan Lo (Yinghsuanlo@my.unt.edu , vooloaa@gmail.com)
Son Chau (sonchau@my.unt.edu)
Nikhil Gaur (gitrepowizard@gmail.com)

Abstract

Our project is about integrating an AI model into a mobile speech to text app, dubbed "Your Notes", that can assist the user by summarizing lecture recordings, reducing the time you need to take study notes with. The user can input lecture recording as an audio file, and Your Notes will spit out the lecture summary in a bullet format. Additionally, the user will also be able to refer back to prior notes taken. Additional notes that "Your Notes" doesn't capture can be manually added by the user. While the app is aimed at college students, this does not limit who can use its functions: for example, corporate employees can utilize this at professional meetings.



It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Figure 1: "Your Note" mobile UI interface. The core features summarization are pictured on the left. On the right side, audio recording and transcript features that may be implemented with extra time. The app will display the summarized text.

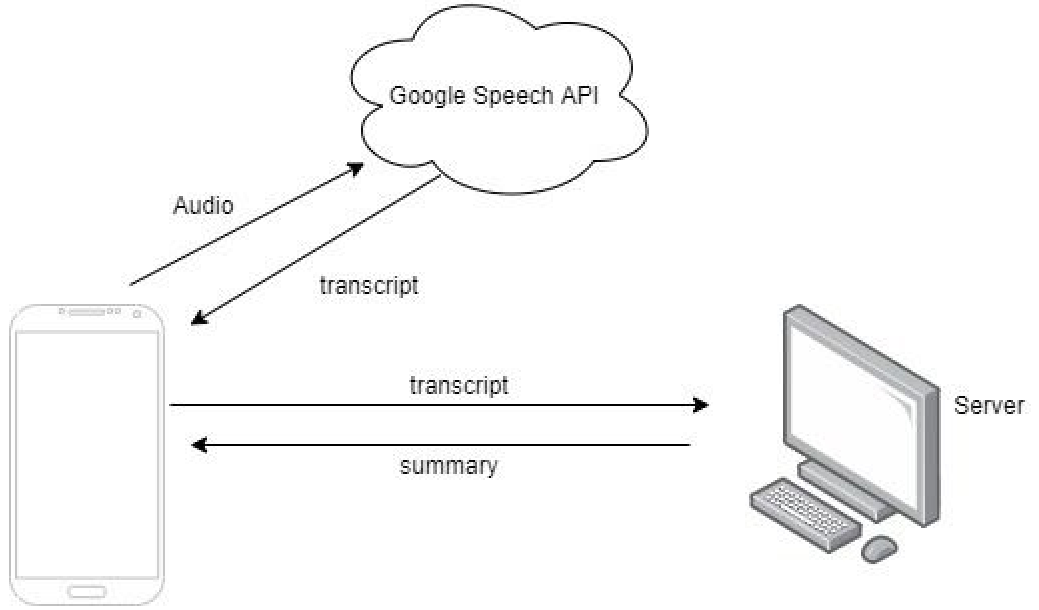


Figure 2: "Your Note" architecture. The audio will be sent to Google Speech API for transcript and display on the phone. Then, the transcript will be processed on another server (or API) for the final summary and returned to the user.

Design Overview

The project is divided into three parts: first, we tackle inputting the audio file and transferring the audio data to a transcript using Google speech to text app. Second, we prepare the transcript to be a bullet point text summarization. Regarding text processing, we are going to use NLTK to clean and tokenize text before we start analyzing our text with Textrank and Pagerank algorithms along with GloVe word embeddings.We may also consider Frequency-Inverse Document Frequency(TF-IDF), and implement into our encoder-decoder model. The text processing will be launched on Jupyter NoteBook for sharing and discussing. Third, we display our results in mobile UI.

The dataset we would like to use for training our model is the Open SLR TED-LIUM Release 3 set, a compilation of talks and texts from Ted Conferences LLC. It contains 2,351 audio talks in NIST sphere format, 2,351 automatic transcripts in STM format, 452 hours of audio, a dictionary with 159,848 pronunciation entries, and all of it is pruned for English speakers. Ideally with this data, we will try to perform supervised learning on our model for quick and accurate transcription results.

In order to efficiently handle the project, we split our members into two parts. Son Chau is responsible for front end and audio to transcript on Google speech app, while Nikhil Gaur, Annie Liu, and Lori Schuan will focus on text processing and implementing the model. Eventually all of us will help with wrapping up front end implementation.

Goals and Milestones

- Sep.30
  - Complete proposal with project structure set up
- Oct. 7
  - Platform setup / Text processing / Google API implementation
- Oct.14
  - Model setting and platform trials
- Oct.21
  - Final presentation

Specifications

- Nikhil, Annie, and Lori
  - Working on Text Summarization / text processing
- Son
  - Working on front-end app development through Flutter

Workflow

- Communication:
  - Meetings over Discord at least once a week on Saturdays.
  - Meetings over Zoom on Mondays with Dr. Albert.
- Python server:
  - Son will host a server for all members to work on.
- Jupyter Notebook:
  - Train our model to transcribe text.
- Flutter:
  - Front-end development.
- Github:
  - Repository of our project work.

Tutorials

- Extractive summarization using textrank/page rank algorithm and glove word embeddings:
  - https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/
- Encoder-decoder models for text summarization:
  - https://machinelearningmastery.com/encoder-decoder-models-text-summarization-keras/

Other Resources

- Open-source TED talk audio and transcript data:
  - https://www.openslr.org/51/
- Alternative dataset - Kaggle TED Ultimate dataset:
  - https://www.kaggle.com/miguelcorraljr/ted-ultimate-dataset
- Tensorflow Lite for mobile app development:
  - https://www.tensorflow.org/lite
- Bidirectional LSTM / RNN:
  - https://towardsdatascience.com/text-summarization-from-scratch-using-encoder-decoder-network-with-attention-in-keras-5fa80d12710e
- Abstractive summarization encoder and decoder architecture –LSTM:
  - https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/
- Encoder-decoder models for text summarization:
  - https://machinelearningmastery.com/encoder-decoder-models-text-summarization-keras/
- Word embedding overview:
  - https://towardsdatascience.com/the-three-main-branches-of-word-embeddings-7b90fa36dfb9