RINGR Podcast Analyzer with AI

ECE 445 Design Document

Bhuvaneshkumar Radjendirane, Elliot Couvignou, and Sai Rajesh
Team 70
TA: Shuai Tang
2/25/2020

1 Introduction

1.1 Objective

To preface this proposal this project was sponsored by RINGR who came to us with their objective. Their service is focused on high-quality multi-agent podcast recording regardless of each person's separate location [1]. An issue about podcast recording is that the amount of time to edit audio takes too long and requires use of some external audio editing software. This makes people who want to get into podcast recording to have to be aware of the technicalities of editing audio which drives away a big audience. There are also many more issues that require even more advanced technology use from users, such as background noise, to further touch up their audio. Since RINGR as a service helps with the initial step of recording the audio on individual channels, they have the potential to use audio processing with the raw audio to help bring down these technical barriers as well as saving time. Modern day audio processing and speech recognition through products such as Amazon Alexa and Google Assistant proves that these issues can be dealt with in a commercial setting.

Our goal is to take in the raw podcast recordings and analyze them with an AI model to slice out unwanted portions as well as filtering out unnecessary noise. The scope of this project is large so our main focus right now is to create the overall feature using existing libraries and then further modifying AI models once our entire system works. Once the main component of an edited audio transcription resulting in new edited audio is finished, we can move onto improving models and adding more advanced quality of life features. This project also provides a side effect of transcripting audio which can be extremely useful for use as a new feature as they don't currently have transcription.

1.2 Background

Podcasting as a business has seen tremendous growth over the past couple of years with music streaming platforms like Spotify increasing their podcasting budgets to where their recent acquisition of The Ringer podcasting network "[brought] the cost of Spotify's podcast shopping spree up to $600 million"[3]. With the market for podcast streaming being higher than before, more people are wanting to get into the podcasting industry. The aforementioned technical barrier that exists with editing audio demotivates most people from trying to push their first cuts as more focus is placed editing the content than actually providing it.
Estimates for how long audio editing takes varies depending on the person or team doing it but it is commonly known to be magnitudes larger than the raw audio length. This process has become cumbersome to the point where companies like "We Edit Podcasts" have grown specifically focusing on editing podcasts for other creators [4]. From this, automating even a small portion of this tedious process can prove to be commercially desirable.

1.3 High Level Design

Since this is a software-only project we will go over how a user is expected to run the finished product. Below on Fig. 1 is an image of the app running during a normal podcast recording. We can use our AI model to start transcription at the start of recording so once the recording is finished we should prompt the user of our overall transcript with little to no delay. From there we will have further user inputs to configure our subject recognition AI to edit the inputted audio and keep the user's 'good' parts. The app currently runs on all mobile platforms and on desktop browsers so we should expect ours to run on all of these.

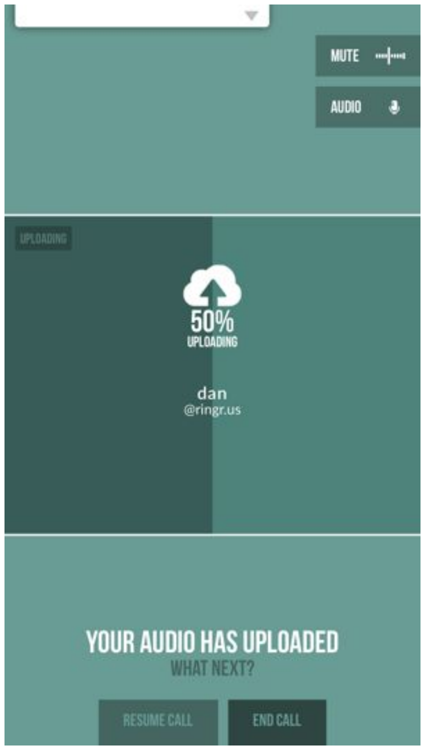


Fig. 1: RINGR mobile app during recording

1.4 High-Level Requirements List

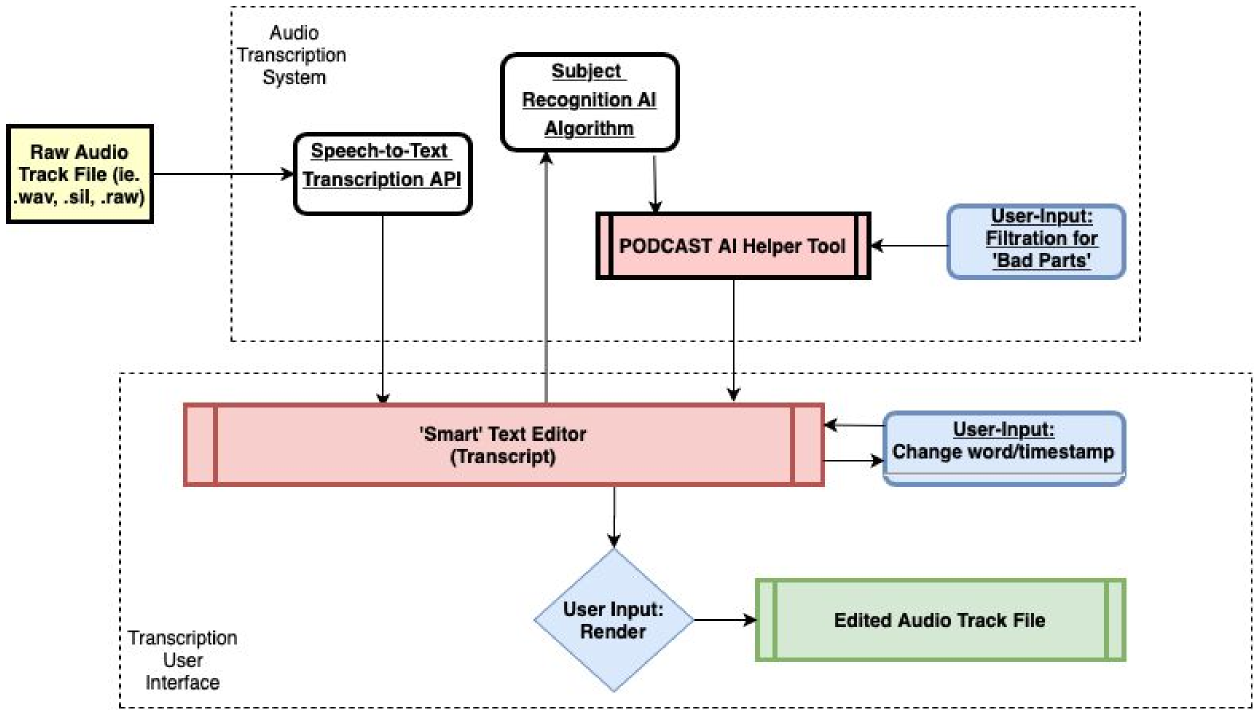
- Since the app is on mobile platforms we aim for our overall feature to be below 1G. If our feature just needs a bigger memory space then we would eventually have to offload the work to a server.
- Runtime of transcription and analysis should at least be quicker than the length of time of the audio input. Our runtime should be more linear (O(n)) to audio input length than anything else.

- Transcription accuracy is below $17 \pm 3\%$ word error rate. Google currently has theirs at around 5% so we can't get too low. $WER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C}$ [5]. *Definitions for these variables are in Reference.*
- Semantic recognition on features like POS tagging needs to have above 85% accuracy and above 75% precision. Our recall also needs to be higher than 50% to reduce false negatives.
- New edited audio should be at least the same audio quality as the input. Quality means we don't introduce any artifacts and disturbances (e.g clicks).

2 Design

2.1 Block Diagram

The block diagram shows the general workflow from the start of recording or inputting audio from transcription to filtration and then to output. Transcription should be quick as we want this displayed so the user so they can edit words and/or word timestamps. We also use the transcription data in our subject recognition model for context based editing based on user inputs. At the moment we wait for a user input to begin rendering the final audio but some features can be done in real-time.


Audio Transcription System
Raw Audio Track File (ie. .wav, .sil, .raw)
Speech-to-Text Transcription API
Subject Recognition AI Algorithm
PODCAST AI Helper Tool
User-Input: Filtration for 'Bad Parts'
'Smart' Text Editor (Transcript)
User-Input: Change word/timestamp
User Input: Render
Edited Audio Track File
Transcription User Interface

2.2 Subsystem Descriptions

2.1 Audio Transcription System

2.1.1 Transcription API

The raw audio tracks provided to the software by RINGR users will be transcribed via a transcription API. For the purposes of our project we will be using IBM Watson's Speech to Text Transcription API. Since the transcribed audio will be used to edit the actual audio file, there must be saved timestamps of start and end points of every word on the transcript. These timestamps must be accurate enough in order to be able to properly edit our words from the transcript and have the changes reflect accurately on the final audio file.

Requirement	Verification
Must be able to return word timestamps within 100ms of accuracy	A. Transcribe a portion of audio with known correct transcription. B. Select a few words at random from the transcription and obtain timestamps for the word from the API. C. Check a spectrogram of the audio at the obtained timestamps and check accuracy of when the final sounds of the words chosen end.

2.1.2 Podcast AI Helper Tool

This component of the final project will run on all podcasts put through our software and will use NLP in order to extract topics or subjects from the transcripts provided by our transcription subsystem [see 2.2.1]. This subsystem will utilize an LDA [Latent Dirichlet Allocation] approach, comparing commonly used words in our transcripts to documents for which subjects have already been recognized.

Requirement	Verification
Must be accurate in recognizing podcast subjects at least 90% of the time.	A. Compile podcast samples on known subjects.

	B. Run Algorithm on the audio files compiled and ensure at least 90% of returned podcast subjects match with initial list.

2.2 Transcription User Interface

2.2.1 User-Input Filtration for ‘Bad Parts’

This subsystem will make it possible to provide RINGR users with a selection of filters that can be applied to the audio file. These filters will allow for features such as ‘stop words filter’ and ‘um/uh’ filters. Users will be able to toggle these filters on and off, thereby making the podcast post production phase simpler for beginners using RINGR as well as high-volume podcasters that wish to save time when editing. In order to implement these filters, we will be utilizing Python NLTK and creating NLP pipelines.

Requirement	Verification
Must accurately detect 85% of instances of ‘word’ [ex: ‘um’ or ‘uh’] that the user wants removed from the transcript.	A. Transcribe at least 30 minutes of RINGR audio. B. Save initial transcription [pre-edited] to a text file. C. Apply filter and specify sound to be removed. D. Check final [post-editing] transcript and compare the number of instances removed to number of instances initially in saved transcript. Accuracy should be above 80%.

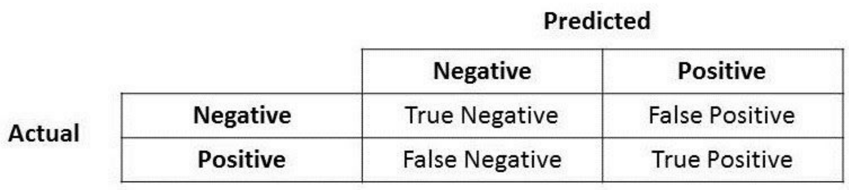
2.2.2 Smart Text Editor (Transcript)

The editor will be filled out with the result of the transcriber, is presented to the user and is always referenced by the Subject Analyzing AI (SA-AI). Any changes made to this transcription in the editor will reflect in the final post production podcast. We will be using timestamps to reference word/phrase edits in the text editor to their appropriate times/locations in the audio file to reflect the changes made. Users will be able to move around and delete segments of the transcript. Once the user has indicated they are finished making their changes, the changes made will reflect in the finally produced audio.

Requirement	Verification
Post-edit audio rendering should not sound fragmented or have words cut off.	A. Transcribe 10 minutes of audio and create spectrogram of original audio file for later comparison. B. Move around sentences and words in the smart text editor and obtain new spectrograms. C. Compare moved portions in new spectrogram to original spectrogram and ensure no words were fragmented in the process.

2.3 Tolerance Analysis

The component that is most essential to this project is most likely the one focused on semantic recognition and applying it to the audio. In our block diagram this component is the semantic AI recognition and the Podcast helper tool. There are quite a bit of statistical calculations to be done for verifying our AI model such as looking at accuracy, precision, recall, and f-score. However these calculations can only be made on things like part of speech tagging and generally easy to verify attributes. Below are is a quick recap on these equations:



		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Fig 2. Variables used in our statistics[6]

$$Accuracy = \frac{(TP + TN)}{(TN + TP + FN + FP)} \qquad Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP} \qquad F1\ Score = 2 * \frac{Precision * recall}{Precision + recall}$$

With all of these statistics we can verify features in our model by verifying that all of these values lead to an intelligent model. Accuracy helps give us a first-glance verification on performance while precision helps with making sure our false positives remain minimal. Recall is helpful in the same way precision is but instead helps keep the false negatives minimal. We want our semantic recognition to accept a wide variety of possible contexts so we aim to achieve a high precision over high recall. Semantic recognition on features like POS tagging needs to have above 85% accuracy and above 75% precision. Our recall also needs to be higher than 50% to reduce false negatives.

These same statistics can also be applied to our transcription component for tolerance as well. This isn't the most essential component but still plays a significant role in the overall quality of the rendered audio.

3 Cost and Schedule

3.1 Cost Analysis

Cost analysis for this project only has two main parts, the labor and potential price rates of transcription/NLP libraries that we use. Since price rates of libraries can't be calculated yet as we haven't decided on the libraries that work best, we will go over the labor part for the immediate cost. A quick Google search shows that on average software engineers make around $33 an hour so we will go off that.

$$LABOR = (\$33/hour) * 2.5 * (16\ hours/week) * (10\ weeks) = \$13,200$$

AI libraries focusing on speech to text offer rates proportional to the audio length. If we end up using some paid transcription model then this does affect our cost analysis since we now add a price for each input we use. On average speech to text rates run at around $0.01 per minute of audio transcription.

Overall this project is expected to cost $39,600 for labor alone and around $0.60 for each hour of audio transcription. A podcast can have many ranges of people included with extremely long segments of audio. If we assume a podcast recording to have around 5 hours of raw data which leads to a transcription cost of $3 per podcast. Since RINGR hopes to use this as a commercial feature for their app, reducing the cost of transcription will be extremely valuable.

3.1 Schedule

This is subject to change as we meet with RINGR and prof. Patel frequently which might redirect the direction of production and focus of this project based on progress.

Week	Bhuvan	Elliot	Sai
2/24	- Move current model from Google Cloud API to IBM Watson - Create pipeline from Watson transcription to	- Render audio from Transcription differences - UI foundation	- Move from current Google speech-to-text API to one with better timestamps. - Work on UI aspect

	editor UI		
3/2	-Work on UI features for editing transcriptions - Fix edit accuracy	- UI - Find good NLP library	- Work on UI; be able to show transcript / begin editing - Get started on NLP topic recognition
3/9	- Begin topic recognition with NLTK	- first NLP results (choose library)	- Work on NLP aspect.
3/16	-Integrate topic recognition with existing UI	- NLP integration	
3/23	-Begin adding filters for various elements (‘um’, ‘uh’)	- UI inputs - NLP related editing	- Work on editing UI -> editing audio
3/30	-Integrate filters into filter tool UI	- Tuning models - Training models??	- Help with training models
4/6	-Fix UI and make interface clean/neat	- Model building (NLP)	- Finetune NLP aspect. - Ensure implemented subsystems are satisfying requirements mentioned in the Design Document
4/13	-Fix miscellaneous bugs and get model optimized for user functionality	- UI	- UI finetuning and testing

4 Discussion of Ethics and Safety

Since this is a software only project, there really isn’t much to consider for safety except for unethical use of audio data. Raw podcast recordings and the output generated by our project “should only use personal information for legitimate ends and without violating the rights of individuals and groups”. [7]Likewise we also don’t want to show any training data used as this goes with the privacy issue. If we do end up uploading the AI models to a remote server, another privacy issue arises since we need to ensure privacy of data on both ends.

It is also our responsibility to cite any resources and libraries used throughout this process in our efforts “to properly credit the contributions of others”[8]. It is apparent that we can’t construct this project from scratch so it is in our best interest to ensure that we respect both the work and it’s right for privacy. Any research that is used for some significant factor of our AI model and how we train it should also be cited when necessary.

Since an “essential aim of computing professionals is to minimize negative consequences of computing, including threats to health”[7], our project must serve the purpose of alleviating editing time and strain from the users. In other words, using our AI should be less tedious for editing audio data than using editing software.

RINGR as a company already holds their own code of ethics in regards to discrimination which follow in line with IEEE and ACM. The ACM ethics code stating that “Computing professionals should foster fair participation of all people, including those of underrepresented groups”[7] means that our AI model shouldn’t place any extra focus on particular users. The ideal goal of this project would be to include recognition of all languages, however our primary focus for now is on english since this composes most of our training data and consumers.

References

[1] ringr.com, “How It Works”2020. [Online]. Available:
https://www.ringr.com/

[2] towardsdatascience.com “How Amazon Alexa works” 2020. [Online]. Available:
https://towardsdatascience.com/how-amazon-alexa-works-your-guide-to-natural-language-processing-ai-7506004709d3

[3] fool.com “Why Spotify Is Buying Bill Simmons' The Ringer” Feb, 2020. [Online]. Available:
https://www.fool.com/investing/2020/02/13/why-spotify-is-buying-bill-simmons-the-ringer.aspx

[4] weeditpodcasts.com, “Home Page” 2020. [Online]. Available:
https://www.weeditpodcasts.com/

[5] martin-thoma.com, “Word Error Rate Calculation” 2020. [Online]. Available:
https://martin-thoma.com/word-error-rate-calculation/

[6] towardsdatascience.com, “Accuracy, Precision, Recall or F1?” March, 2018. [Online].
Available: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

[7] acm.org, “ACM Code of Ethics and Professional Conduct” 2020. [Online]. Available:
https://www.acm.org/code-of-ethics. [Accessed: 13-Feb-2020].

[8] Ieee.org, "7.8 IEEE Code of Ethics" 2020. [Online]. Available:
https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 13- Feb- 2020]