ASR-based System for Speech Therapy in Adults

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Brandon Williams
Spring, 2021

Technical Project Team Members
Brandon Williams

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature _____ Date ___04/19/2021__
Brandon Williams

Approved _____ Date ___04/19/2021___
Jundong Li, Department of Computer Science

ABSTRACT

Children and young adults who suffer from hearing loss or auditory processing problems may experience a delay in the development of their speech given that they cannot hear quiet sounds such as “s”, “sh”, “f”, “t”, and “k”, which lead to speech impairments. The problem is that children often discontinue speech therapy once they enter upper-level schooling as these minor speech issues start to become more permanent, leaving them without access to a speech therapy solution. Previous apps found in speech therapy work by helping to fix single words or syllables at a time without considering the inclusion of others' perception of what speech-impaired users may sound like to them. In transcripts of Zoom’s closed captioning services this school year, their platform occasionally outputted words that did not align with what the professor may have been attempting to say, inadvertently pointing out “quirks” in their pronunciations. This topic is worth examining in that this added context may identify words and sounds that users may not have realized they struggled with, a key mechanism which would provide more input in the speech correction process.

Applying this concept to an automated form of speech therapy, I will be proposing a mobile application would prompt the user to read aloud from a pre-selected script into their device, feed this speech signal into a highly accurate open-sourced audio-to-text algorithm that would transcribe the audio, and flag words that did not match the original transcript as “mispronounced”. Users will be given the opportunity to see what the program interpreted their “unclear” speech as, equipping them with a unique insight into “how” they may be mispronouncing sounds through the design of a human-computer friendly interface and a unique application of open-source deep learning audio-to-text algorithms.

1 Introduction

In 2016, the National Institute on Deafness and Other Communication Disorders reported that 7% of American children have been diagnosed with a speech disorder. Of these speech disorders, some causes can be from developmental disorders, neurological disorders, and hearing loss. Although speech impairments can originate from a variety of causes, this tool targets a niche population of individuals whose speech-language impairment went undetected or poorly treated during childhood, which led to long-term consequences as articulation and pronunciation challenges worsened into adulthood.

Without this tool or tools that were similarly devised, it is likely that individuals would need to accomplish similar outcomes through explicit pronunciation teaching with a trained professional. However, there are difficulties associated with this methodology of pronunciation teaching that may make this tool a more attractive option to this untapped market. Pronunciation teaching typically requires individual attention which poses a problem in classroom settings. Additionally, those who have difficulty speaking clearly find it to be a heavy mental task that demands coordination and muscle control which also could make potential students afraid to speak in the presence of others [1]. This introduces the need for a computerized, automated tool with an ability to provide to-the-point feedback when detecting mispronunciations.

It is worth noting that previous research has shown that repeated, patient-engaged therapy with a professional is highly effective in improving intelligibility of speech [2]. Of course, this may not be possible in cases where patients may not have accessibility to a qualified speech therapist. While face-to-face treatment may not be possible for some, it is worth nothing that mobile broadband is now accessed via smartphone or tablet technology in 68% of American households [3]. These tools have the ability to facilitate easily accessible and affordable speech therapy solutions. However, there is currently an absence of impactful speech-language therapy solutions in the mobile-application marketplace for adults. A study that systematically identified and evaluated a broad range of mobile applications for adults with communication disorders and found a lack of evidence-based applications with a focus on human factors and patient-led design approaches. Additionally, from a total of 2680 applications identified from the Google Play Store, Apple app store, and web searches, only a total of 17 applications were designed for speech therapy-- defined as therapy to improve perception and production of speech sounds and speech segments [4]. Of these applications that were generally successful, it was found that functionality and engagement were rated highly, whereas it was noted that feedback was very general in nature and didn’t provide specific, valuable insight.

Luckily, the speech rehabilitation domain has seen a surge in the use of computer-based speech therapy with the use of Automatic Speech Recognition (ASR). ASR is the intersection of computer science and computational linguistics that enables the conversion of speech into text and is otherwise known as a speech-to-text algorithm. Current speech therapy tools leverage ASR to provide patients with autonomous exercises without the intervention of a speech therapist. The proposed tool will incorporate ASR technology as a way to provide automatic feedback on pronunciation.

2 Background

An ASR system consists of an acoustic model and a decoder. This acoustic model takes a speech signal as input and produces an analysis that assigns probability scores of different phoneme states [5]. In linguistics, phonemes are defined as any of

the perceptually distinct units of sound in a specified language that distinguish one word from another. An example of this is p, b, d, and t in the English words pad, pat, bad, and bat [6].

In the figure below, the front-end process is responsible for pre-processing the speech signal from a given user and parameterizing it for the back-end process.

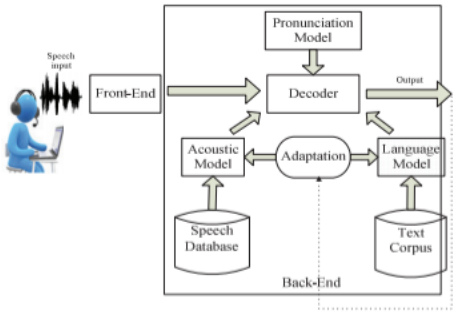

Figure 1: Block diagram of ASR system

The input speech is converted into a vector of numerical values and is known as feature extraction which helps with encoding a speech signal. The most prominent technique for this process includes Mel Frequency Cepstral Coefficient (MFCC), known for its efficiency and simplicity, and Perceptual Linear Prediction (PLP), which are best suited for Deep Neural Network (DNN) classifications [7]. Next, the acoustic model leverages Guassian mixture models (GMMs) to link observed features of the speech signals with the expected phonetics of the hypothesis sentence [8]. Historically, acoustic models have been implemented with hidden Markov models (HMM), which are models built purely in a mathematical manner and are deterministic in nature. Used in combination with Guassian Mixture Models (GMM), similarity scores between the input signals and training data in the speech database are generated and sequence modeling is performed in order to form phone sequences, which form the basis for full words and sentences [9]. Lastly, potential transcriptions are generated with the help of the acoustic, language, and pronunciation models with the best one being selected by the decoder.

However, the described approach is considered generative, meaning that it is based on the probability distribution with the given observations. This is considered a conventional method for acoustic phonetic modeling. Thanks to advancements in the field of speech recognition, advanced models such as Deep Neural Networks (DNN) can now be applied and are considered discriminative approaches, meaning that it has a conditional distribution using a parametric model [10]. Recently, hybrid models have seen a rise in use as researchers refine the conventional GMM-HMM based approach by instead using DNN training [11]. By combining the strength of these two approaches, the word error rate decreases and performance of ASR systems improve, where word error rate is defined as a metric for measuring the accuracy of speech-to-text transcriptions [12].

ASR has been applied tremendously to Computer-aided Pronunciation Training (CAPT) systems which is typically used to teach English as a second language to non-native speakers. Computer-aided pronunciation training is made up of two tasks--detecting mispronunciations and providing corrective feedback. The systems that leverage ASR fall into two categories: speaker-dependent and speaker-independent. The former requires small training set size while a speaker-independent system requires a large training size. The selection of one over the other or selecting a DNN-based model over an HMM-based one, could be the difference between offloading resources to the cloud or operating a computer-aided pronunciation training system on a stationary computer.

3 Related Work

Much of the existing systems relevant to this technical paper will fall under the umbrella of what we defined as Computer-assisted Pronunciation Training (CAPT). Often used to complement the resources for learning a second language, it has been found that the use of immediate and personalized ASR feedback has led to a significant reduction in the number of mispronunciations. While the underlying acoustic models may differ in our application given that the proposed tool targets native speakers with minor speaking impairments who may benefit from identifying mispronunciations in their speech, CAPT still offers fundamental information that will be useful in the design of the proposed system. A successful implementation of such a system is introduced using the Dutch-CAPT as a case study.

3.1 Dutch-CAPT

Developed by researchers within the Department of Linguistics at Radboud University in the Netherlands, Dutch-CAPT is a system built on ASR that provides corrective feedback on problematic speech sounds for individuals learning Dutch as a second language. It was designed as a client-server architecture with communication occurring through two sockets, one to exchange commands and the other to communicate the data associated with speech signals. The user interface is hosted on the client and its

contents are laid out such that the exercises are located on the top half of the screen and the feedback is provided on the lower half of the screen.

When the client is started up, the user enters personal information such as gender, UI language preferences, and a unique ID that allows the system to keep records of the student's account activity [13]. Gender is a required field that allows the server to choose a different speech analysis depending on if the user was a male or female at birth [13]. The client also contains a library of 106 exercises split into 4 units and are completed sequentially for research purposes so that the data can be compared fairly and holistically across the test subjects [13].

The server contains the technology required to perform the speech analysis logic and manages different client processes using a server log file that contains important context occurring in the actively running clients as well as on the server. The flow of communication between these two main parts of the system can be illustrated in the form of a diagram as shown below:

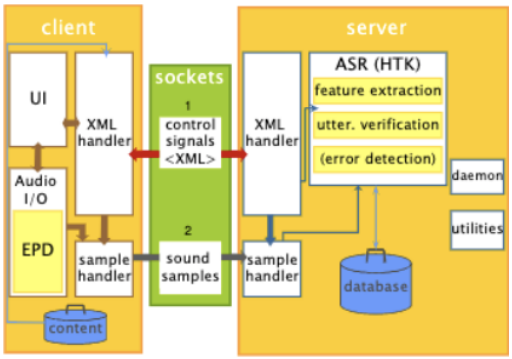

Figure 2: Client-server architecture of Dutch-CAPT system

At a functional level, the server receives a speech signal from the client, determines if it is a valid utterance that aligns with one of the provided scripts, and then assesses the pronunciation provided by the speaker [13]. If there is no error detected in the provided pronunciation, then the UI displays a congratulatory message, the transcript of what was pronounced, and a green smiley face for positive reinforcement. If error is detected, the incorrect phonemes are highlighted red and a red, upset face is displayed.

ASR was implemented using 37 gender-dependent hidden markov models (HMM) that were trained on native speech from the broadcast news of the Spoken Dutch Corpus [13]. After conducting a study that split test subjects into groups receiving automatic feedback on problematic phonemes and groups receiving no feedback on problematic phonemes, it was found that the pre-test vs post-test scores were significantly improved in the non-control group which shows that ASR is still useful in improving pronunciation errors despite a system that wasn't 100% accurate in error detection [13].

The success of this system came from providing feedback on only a limited selection of sounds considered too problematic and from designing a simple, to-the-point feedback message system that didn't overwhelm users of the UI. Despite this system being an early example of a successful speech therapy tool built on ASR in 2009, significant strides have been made in providing accurate feedback since then. Being that it was designed for research purposes, the system also lacks attractive and engaging features. The graphics are not impressive and there is a lack of gameplay elements such as progress bars, scores, and other similar features that promote engagement.

3.2 Rosetta Stone Language Learning Software

A CAPT program based on top of ASR, this software provides a sequence of interactive dialogues where the user selects a "correct" response from several speech sounds that are different on a phonetic level. If the user's response matches well with the stored pronunciation model, the dialogue continues. Otherwise, the user has to repeat the response. The software uses a Spoken Error Tracking System which delivers a global pronunciation score and highlights words in the sentence that were marked as incorrect pronunciations, similar to Dutch-CAPT [14]. Rosetta Stone designed a proprietary speech recognition engine, *TruAccent,* which compares the speaker's voice to native and non-native speakers so that real-time feedback can be provided for accurate pronunciation [15]. This is considered a comparative phonetics-based system, which uses stochastic analysis to compare the phonemes in the native language of the learner to those in the target language. A pronunciation dictionary is typically compiled with correct and probable incorrect pronunciations and used to detect mispronunciations in the speech of the learners [16]. An example of this using the Rosetta Stone software is illustrated below:

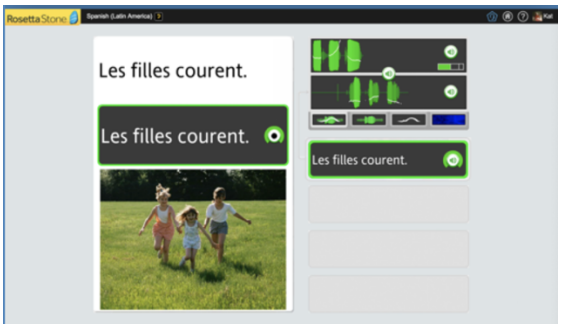

Figure 3: Speech signals of non-native speaker compared to native speaker's speech signal on Rosetta Stone software

However, this speech recognition model was trained with the intent of detecting mispronunciations in speakers learning a non-native language, which does align similarly with the intent of the proposed tool. The purpose of including this software was to introduce the concept of comparative phonetics, which could still potentially add value to the proposed tool.

3.3 Modern Application Structure of Visual Speech Therapy App for Children

This proposal takes into account that a speech recognition system is attempting to transcribe what the user is saying whereas a speech therapy system already knows what the user is going to say given that it provides the script. This eliminated the need for some parts of normal ASR systems. Additionally, the designers take into consideration the relative ease of setting up hidden-Markov models (HMM) for ASR given that they require less training data and train themselves on the speaker's speech [9]. Given that the target population is adults with speech-impairments, their speech may be harder to recognize, so this implementation would better measure improvements in their speech patterns [9].

Rather than implementing the system locally, the designer chooses to separate into a client and server, where the server is hosted in a cloud service, allowing for more expensive calculations to run on servers and letting users avoid investing in high performance equipment. Another reason for choosing the client-service architecture was that it allows for ease of integrating new exercises on the client side or new processing logic on the server side without requiring users to constantly update their program [9]. Client-server communication uses Datagram Transport Later Security (DTLS) protocol in order to ensure security and reliability [9]. A simple diagram of the client-server architecture is shown below:

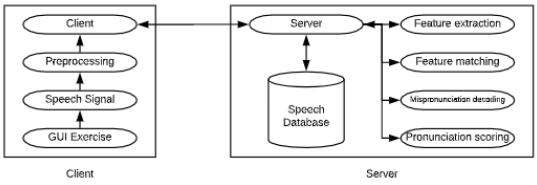

Figure 4: Diagram of system design for proposed VST system

An interesting implementation decision by this designer was that they chose the speech database to be built up from samples gathered from users which would allow the program to adjust to specific speech impediments. This could be a poor design decision if there are not enough users, and it may just be better to train the model on an existing larger database that comprises hundreds of hours of audio from speech-impaired data points. Features of this design to promote engagement includes rewards in the form of victory animations, achievements, and a leaderboard.

3.4 Shortcomings of Related Work

There will always be unavoidable issues with CAPT applications as erroneous feedback will occur through mispronunciation errors going undetected or falsely labeling pronunciations as errors, which can frustrate users and deter them away from use of the application [14]. However, a core issue that persists across these related works is the lack of productive guidance from the computerized assistant. While feedback is useful in highlighting utterances as incorrect or unintelligible, there is a lack of constructive exercises that actually help users to correct this error. The proposed tool addresses this gap by incorporating audiovisual feedback and features that allow users to see how others perceive their speech, offering a unique way for users to consider correcting their pronunciation errors.

4 System Design

A design that aligns with my vision of the proposed tool is one centered around ease-of-use, accurate mispronunciation detection, and constructive feedback features. Given the accessibility of mobile devices in comparison to computers, we will want to develop a system architecture compatible with mobile applications. This is a non-negotiable design decision as we are targeting social groups with speech impairments that cannot afford private, professional help from a trained speech therapist. Mobile applications have the potential to make speech therapy available anywhere that a mobile device can be used. Designing a web development application would only exclude these impacted members of society and further exacerbate the disparity between those who cannot currently access speech care services and those who currently can.

4.1 The Engine (ASR, Mispronunciation Detector, and Feedback Agent)

Before the mobile application architecture can be illustrated, we must first design the "engine" of this proposed speech therapy tool, which will consist of three main components-- an ASR system, a mispronunciation detector, and a feedback agent.

The first step taken was to decide which flavor of ASR system to build. Typically, an ASR system is split into multiple parts-- an acoustic model, a pronunciation model, and a language model. An interesting design decision to take into consideration is deciding what type of data to train the acoustic model of the ASR system on, as speaker-dependent ASR is one that is trained to recognize the speech of an individual speaker whereas a speaker-independent system is one trained using magnitudes more data to recognize the speech of any speaker. On the one hand, if the acoustic model of the ASR system is trained on a database of just the speaker's speech, then the system will just learn how to transcribe the speaker's disfigured speech. We don't want the system to learn what the speaker is trying to say, but rather what the speaker is mispronouncing, and will therefore require a speaker-independent system. For this reason, we will use a pre-trained acoustic model that uses the *LibriSpeech* dataset, which is the most commonly used audio processing dataset in speech research and comprises nearly 1000 hours of labelled speech data and is the standard benchmark for training and evaluating ASR systems. Next, there was the decision on what type of hybrid acoustic model to implement. As described in the Background section, two of the most prominent hybrid models are HMM-GMM and HMM-DNN. In addition to HMM-DNN being considered more state-of-the-art and typically providing better results, it was found that the Phone Error Rate (metric assessing effectiveness of acoustic model) for disordered speech was 43% for the GMM-HMM model while it was just 36% for the DNN-HMM model [17]. Given that one of the design principles of this tool includes accurate mispronunciation detection, the HMM-DNN model is selected. Once the model is trained, it would be my intent to fine-tune the model using a database of audio from those with speech disorders. These can be implemented, trained, and tested using the Kaldi, a toolkit for speech recognition written in C++. There is extensive documentation on Kaldi's website that offers tutorials for setup of training deep neural networks for speech recognition systems on Graphical Processing Units (GPU's), which significantly speed up training time. As for the pronunciation model to feed as a parameter to Kaldi's model training process, Carnegie Mellon University (CMU) has an open-source machine-readable dictionary called *CMUdict* for North American English that contains over 134,000 words and their pronunciations. These models help the creation of a language model (although it is unclear exactly how given its complexity), which determines the probability of a sequence of words. It will be crucial to ensure that this ASR system doesn't use the context of surrounding words to predict what word was said, as this focuses more transcribing the correct word versus detecting mispronunciations. Low-level details aside, this is how the ASR system will be built and below is a diagram of how it will be trained and tested:

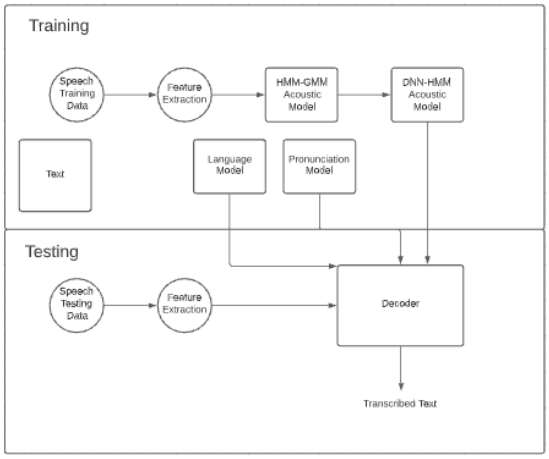

Figure 5: ASR design of proposed system

The implementation of the second critical part of the "engine", the mispronunciation detector, is far less complex. There will be a phrase displayed for the user to repeat into their microphone on the screen, which can be considered the "ground truth" data. The user's speech will be transcribed according to the ASR system built above and compared against the "ground truth". Transcribed words that do not match the words given in the "ground truth" will be flagged as mispronunciations. An alternate method would also be leveraging the probabilistic scores of each transcribed word produced by the ASR system. For example, we could set a threshold where if a transcribed word is only assigned a probability score of below 75%, it is considered a mispronunciation.

Lastly, the feedback agent extends the mispronunciation detection by actually showing what was transcribed against what was expected. This is the key feature of the proposed tool and offers an unprecedented level of context that allows users to make adjustments by seeing how "others" perceive their speech. If feasible, the feedback agent will also include a talking virtual head that shows the lip movements associated with pronouncing the expected word as well the lip movements associated with pronouncing the transcribed word, providing the user with audiovisual feedback, a method which has proved useful in illustrating how a sound is articulated.

4.2 System Architecture

When considering a mobile application design, one is met with understanding the limitations of the number of resources that can be used locally on the device. The enormous size of the required data for our speaker-independent ASR system immediately eliminates the possibility of running the entire tool on a small device like a tablet or a phone. The design decision of offloading resources elsewhere fits into a client-server architecture and this is the choice made here. By performing computationally expensive tasks such as training the acoustic model or processing speech signals through the ASR system online instead of locally, the client isn't bogged down during real-time completion of speech tasks or exercises. More specifically, we will implement a cloud architecture as this means greater accessibility through mobile devices, increased cost effectiveness since we remove buying a lot of physical hardware, customizable security settings to protect user data, and scalability. By splitting into a client and cloud-hosted server architecture, we can keep client and server features separate without an update in one affecting the other. For example, we can update clients to contain new UI features or update processing logic on the server side without impacting the client.

The user interface (UI) will be hosted on the client side and will contain exercises, progress visuals, and feedback agents that will assist the client in improving pronunciation. On the server side, the entire "engine" comprising the ASR system, mispronunciation detector, and feedback information will be hosted here. The server will initially be spun up using an AWS EC2 instance with a possibility for horizontal scaling if enough customers are sending speech signals to the model on the server. However, users will need network access and we likely will have to pay for the bandwidth. Additionally, this cloud server will require an API for processing speech analysis requests from mobile clients and may include things like handling authentication so that only authorized users can access this server. For example, we could use a REST structure and send data from the application to the server in the form of HTTP Post commands and in response the server sends the information associated with a given speech signal inputted by a user on the client side. An additional advantage of this design is that all the logic is hosted on the server, so it becomes easy to port the app to different platforms such as iOS, Android, and web. Additionally, we can spin up an Amazon RDS instance also hosted on the server side that stores information associated with each user such as name, gender, username, audio files, progress, and actions performed by users. This logging system will allow us to learn how the application is being used and provide us with information that may help optimize internal mechanics. A diagram of the client-server architecture can be found below:

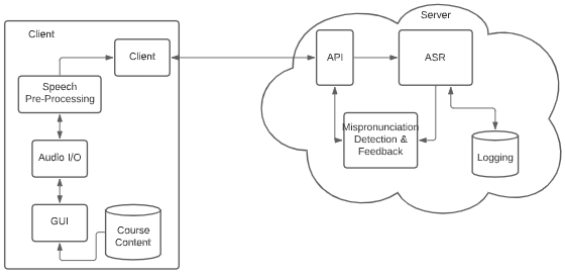

Figure 6: Client-server architecture of proposed speech therapy tool

4.3 Risk Analysis

A major limitation in the design of this tool was existing literature on ASR-based speech therapy systems designed for adults. Performance wise, there may be issues with receiving feedback on the client in real-time given that the audio signal must

first be pre-processed, packaged into an API request, sent to a cloud server, unpackaged from the API request, fed into the ASR engine, parsed for mispronunciation errors, packaged back into an API message, and then sent back to the client. Users may be frustrated if it takes several seconds to receive feedback after speaking into their device. Of course, given that this is merely a design proposal, we lack a physical implementation to run experimental tests on to determine the latency of client-server communication as well as the time it takes to process speech signals. If we were to do this, we could fine-tune the ASR engine to optimize for runtime so long as accuracy isn't compromised beyond an acceptable threshold. Additionally, C++ is a strong option for implementing speech-recognition services and for training machine learning models given that it optimizes runtime performance, so the back-end logic can be implemented using this language to improve performance. A notable limitation is that the accuracy of our feedback is only as good as the accuracy of the audio-to-text algorithm that we implement using acoustic models. It is also worth noting that the training data of this model is heavily lopsided in favor of "clear" speech, so we may get unexpected results when applying this model to the specific segment of adult, speech-impaired individuals

From a security standpoint, client-server communication inherently comes with the risk of messages being intercepted in between the two endpoints. AWS Cloud Services allows for a manual configuration of various security features to protect servers. Measures taken to ensure that customer data won't be compromised include limiting server access to only other instances on the server-side, encrypting sensitive user information packaged into the API message, and limiting incoming and outgoing traffic using Security Groups and NACL's. In addition, customers of this app will likely need to consent to the storage of their user data in order to comply with data protection laws.

5 Procedure

The user interacts with the program by first logging into their account and selecting the exercise they want to complete. They are asked to read a pre-selected phrase by tapping the microphone icon and speaking into their mobile device. The server processes their speech and sends back the transcription. Words that match the original text will be highlighted green and words that do not will be highlighted red. The user does not receive a reward for completion of this exercise until all words match. A bareboned UI that one could expect to see is found below:

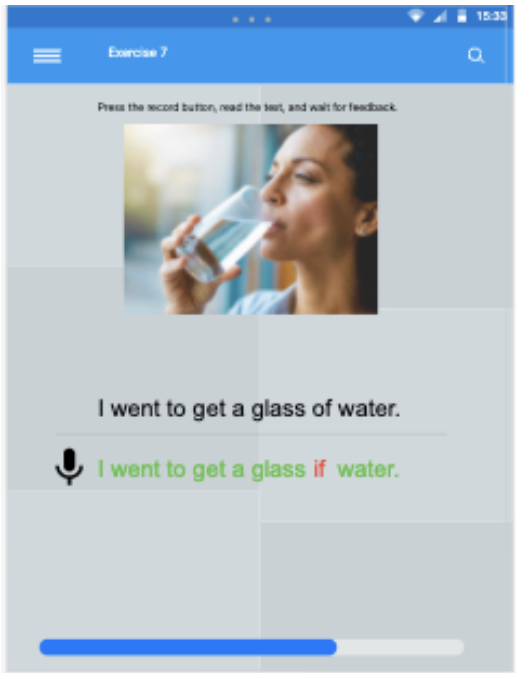

Figure 7: Example exercise where user mispronounces a word

In addition to text feedback, the user can scroll over one page by swiping across the screen to see the spectral diagram of their speech signal of the mispronounced word versus the word they are trying to pronounce. Additionally, on the third page, there would be a virtual, talking head that shows the lip movements as well as the audio associated with articulating that specific word as another form of audiovisual feedback.

6 Results

This proposed design addresses the issue with feedback quality in the current market of mobile therapy apps. While there is no physical implementation due to the limited timeline this semester and working individually on this proposal, there are a number of ways to assess the impact of this app. The study to evaluate mobile applications for speech-therapy language in adults with communicative orders (https://mhealth.jmir.org/2020/10/e18858/) can be repeated and compared against existing solutions. This could involve gathering a group of volunteers belonging to the target population and have them review this app on quality, engagement, content, functionality, aesthetic, information., and perceived impact via feedback. To assess effectiveness, we could additionally repeat the study conducted for the Dutch-CAPT tool as mentioned above. A dozen or so volunteers can use the app where half uses the feedback mechanism, and the other half does not use the feedback mechanism. They can take a pre-test that scores their pronunciations of hundreds of different utterances prior to using the tool and then a post-test that scores the same pronunciations after using the tool. By comparing the pre-test to post-test improvements across the control group and the target group, the effectiveness can be determined by performing tests of statistical significance.

7 Conclusion

In this paper, I have reported on research addressing current applications of ASR in the fields of language learning, mispronunciation detection, and correction of speech impediments. The introduction of an ASR-based speech therapy is introduced to solve the problem of providing a novel, perception-driven feedback system designed to give users a unique framework for progressing on the pronunciations that they struggle with. The proposed tool offers adult-friendly features such as comparisons of spectral diagrams of spoken speech signals vs target speech signals, ASR transcriptions of mispronounced words vs "ground-truth" words, and audiovisual features such as AI that shows proper lip pronunciations to articulate target sounds. The client-server architecture allows for separation of UI-driven features on the device and speech-processing logic on servers in the cloud. If an implementation of this design were to occur, experiments and user studies would be required to test the accuracy of the system and effectiveness of the tool as an overall form of speech therapy in adults. Given the lack of literature that investigates this topic, it still remains to be seen if this methodology would improve pronunciation errors relative to other systems found in this space.

8 Future Work

Given an extended timeline, there are several features and questions that would be worth investigating. For example, there was not enough time to determine the breakdown of training and testing data that would produce an optimal acoustic model for our target population of adult, speech impaired individuals. With the current model trained as a speaker-independent model using "clear speech" data, the exercises may be too difficult for an individual with serious speech disorders to successfully complete. It would be worthwhile to train the acoustic model to adjust to speech impediments such that the progress of their therapy can be tracked easier. Additionally, a critical and rather weak assumption of the proposed ASR system is that it will produce words that match one for one with the provided text that users read from. This is not realistic as it doesn't account for insertions or deletions of sounds and may struggle with words that are slurred together by the user in a continuous manner. Lastly, an advanced feature that leverages AI and machine learning would involve integrating facial recognition technology to capture lip movement and facial expression. This offers another form of input from the user in addition to a speech signal that the feedback agent can use to provide even more valuable feedback. Future work also involves the physical implementation of the proposed system, although the underlying mechanisms associated with building the ASR system are considered to be extremely complex, so it may be worthwhile to purchase and integrate an existing commercial-grade ASR system to save significant development time.

REFERENCES

[1] S. Witt and S.Young, 1997. Computer-assisted Pronunciation Teaching based on Automatic Speech Recognition *Jager, S. Language Teaching and language technology* (1998), 25-35.

[2] Ramig, L. O., Sapir, S., Countryman, S., Pawlas, A. A., O'Brien, C., Hoehn, M., & Thompson, L. L. (2001). Intensive voice treatment (LSVT) for patients with Parkinson's disease: a 2 year follow up. Journal of neurology, neurosurgery, and psychiatry, 71(4), 493–498. DOI: https://doi.org/10.1136/jnnp.71.4.493

[3] U.S. Census Bureau. (2019, May 23). More Than Two-Thirds Access Internet on Mobile Devices. The United States Census Bureau. https://www.census.gov/library/stories/2018/08/internet-access.html

[4] Vaezipour A, Campbell J, Theodoros D, Russell T, 2020. Mobile Apps for Speech-Language Therapy in Adults With Communication Disorders: Review of Content and Quality JMIR Mhealth Uhealth 2020;8(10):e18858 URL: https://mhealth.jmir.org/2020/10/e18858 DOI: 10.2196/18858

[5] Arora, V., Lahiri, A., & Reetz, H. (2018). Phonological feature-based speech recognition system for pronunciation training in non-native language learning. Journal of the Acoustical Society of America, 143(1), Article: 98.

[6] Simpson, J. A., Weiner, E. S. C., & Oxford University Press. (1989). The Oxford English Dictionary. Oxford: Clarendon Press.

[7] AIP Conference Proceedings 1883, 020028 (2017); DOI:https://doi.org/10.1063/1.5002046

[8] Aggarwal, R.K., Dave, M., 2011. Acoustic modeling problem for automatic speech recognition system: conventional methods (Part I). Int J Speech Technol 14, 297 DOI: https://doi.org/10.1007/s10772-011-9108-2

[9] D. Salwerowicz, 2019. Design Proposal for a Software Tool for Speech Therapy – Modern Application Structure of Visual Speech Therapy App for Children DOI: http://dx.doi.org/10.13140/RG.2.2.16038.98885
[10] Cutajar, M., Gatt, E., Grech, I., Casha, O. and Micallef, J. (2013), Comparative study of automatic speech recognition techniques. IET Signal Process., 7: 25-46. DOI: https://doi.org/10.1049/iet-spr.2012.0151
[11] Romdhani, S. (2015). Implementation of DNN-HMM Acoustic Models for PhonemeRecognition. https://uwspace.uwaterloo.ca/bitstream/handle/10012/9061/Romdhani_Sihem.pdf?isAllowed=y&sequence=1
[12] Chen, H. (2021, February 22). Does Word Error Rate Matter? SmartAction. https://www.smartaction.ai/blog/does-word-error-rate-matter
[13] Neri, A., Cucchiarini, C., & Strik, H. (2008). The effectiveness of computer-based speech corrective feedback for improving segmental quality in L2 Dutch. ReCALL, 20(2), 225-243. DOI: 10.1017/S0958344008000724
[14] Rogerson-Revell, P. M. (2021). Computer-Assisted Pronunciation Training (CAPT): Current Issues and Future Directions. RELC Journal, 52(1), 189–205. DOI: https://doi.org/10.1177/0033688220977406
[15] Speech Recognition | Rosetta Stone. (2021). Rosettastone.Com. https://www.rosettastone.com/speech-recognition/
[16] Agarwal, C., Chakraborty, P. (2019) A review of tools and techniques for computer aided pronunciation training (CAPT) in English. Educ Inf Technol 24, 3731–3743 (2019). DOI: https://doi.org/10.1007/s10639-019-09955-7
[17] M. Shahin, B. Ahmed, J. McKechnie, K. Ballard, and R. Gutierrez-Osuna, 2014. A comparison of gmm-hmm and dnn-hmmbased pronunciation verification techniques for use in the assess-ment of childhood apraxia of speech inProc. of INTERSPEECH,2014, pp. 1583–1587.