UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

Faculty of Engineering Science and Technology
Department of Computer Science and Computational Engineering

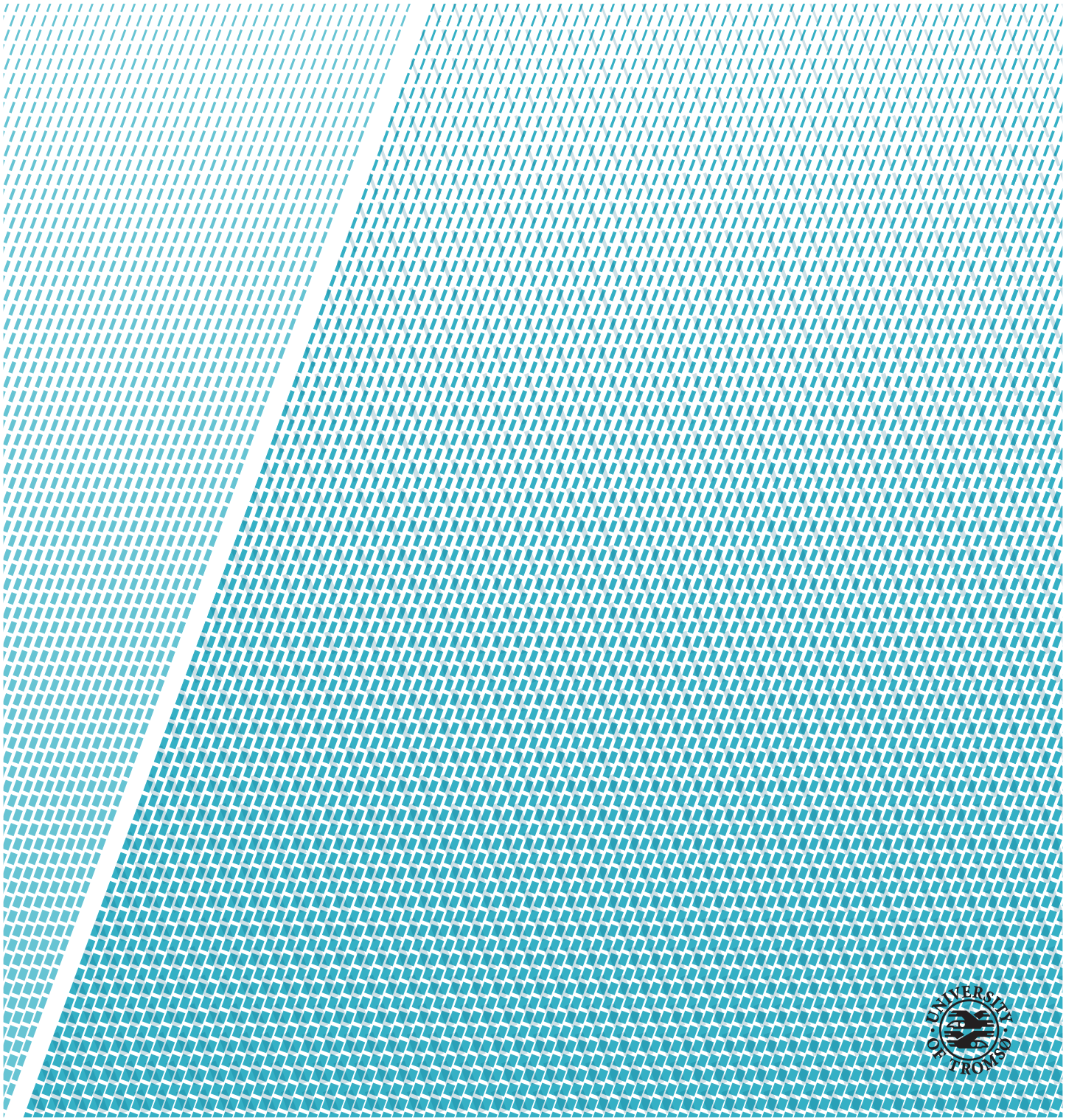
Design Proposal for a Software Tool for Speech Therapy

Modern Application Structure of Visual Speech Therapy App for Children

—

Daniel Aaron Salwerowicz

Master thesis in Applied Computer Science, June 2019



To my amazing fiancé, Christopher Thomas Lister, as well as my wonderful parents, Artur and Elżbieta Salwerowicz.

Thank you very much for the immeasurable support you gave me.

Abstract

This thesis describes results of study into development of a Visual Speech Therapy tool for use in therapy of speech impediments. It describes what Visual Speech Therapy tools are, their benefits, and what tools are available on the market. Main focus being therapy for children and teenagers with dysarthria, hearing problems, and functional speech impediments.

After introducing the Visual Speech Therapy programs it presents technologies available to development of such tools, weighing their limitations and benefits against each other and presenting the best one. It also proposes how such a tool would be developed and used by its end users.

Lastly it suggests how to move forward with the project and develop a working product from this proposition.

Acknowledgements

I would like to thank Nina Opdahl, a speech therapist in Narvik municipality, for presenting her idea of creating new Visual Speech therapy tool, as well as her support during the project.

My sincere thanks go out to Rune Dalmo and Tanita Fosli Brustad, supervisors at UiT-IVT, for their help and feedback during my thesis, as well as the whole faculty at Department of Computer Science and Computational Engineering at UiT for getting me this far.

Last but not least I want to thank my colleagues Christopher Kragebøl Hagerup, Kent Arne Larsen, Olav Kjartan Larseng, and Hans Victor Andersson Lindbäck, for their support during my master studies, great discussions we had during the project, and their valuable feedback.

Contents

Abstract iii

Acknowledgements v

List of Figures ix

List of Tables xi

1 Introduction 1
- 1.1 What are VSTs? . . . 1
- 1.2 Why are VSTs useful? . . . 2
- 1.3 Intended Features and Functionality . . . 3
- 1.4 Existing programs . . . 5
  - 1.4.1 IBM Speech Viewer III . . . 5
  - 1.4.2 Phone and Tablet Apps . . . 6
  - 1.4.3 Electropalatography . . . 7

2 Methods 9
- 2.1 ASR . . . 9
- 2.2 Recurrent Neural Networks . . . 10
- 2.3 Long Short-Term Memory Networks . . . 11
- 2.4 Hidden Markov Models . . . 13

3 Design of VST Using HMM 17
- 3.1 ASR Structure for Speech Therapy . . . 17
- 3.2 VST Design . . . 20
- 3.3 Program Interaction . . . 22
- 3.4 Technology . . . 23

4 Results & Discussion 25

5 Conclusion 29

Bibliography 31

Appendices **35**

Appendix A List of Abbreviations **35**

Appendix B Problem Description **37**

Appendix C Initial Survey **43**

List of Figures

1.1 Exercise overview in IBM Speech Viewer III 4
1.2 Examples of mobile apps for speech therapy 7
1.3 Electropalatography system diagram 8

2.1 Diagram of Recurrent Neural Network's structure 10
2.2 Long Short-Term Memory node connection diagram 12
2.3 Overview of data processing in LSTM network nodes 12

3.1 Data flow diagram in ASR for VST 18
3.2 HMM structure for single phoneme recognition 19
3.3 System structure design proposal for VST 21
3.4 Example of an exercise for the VST program 21
3.5 Interaction overview between VST and its users 23

List of Tables

4.1 Comparison table measuring proposed system and current products against IBM Speech Viewer 27

1 Introduction

This chapter will describe both what Visual Speech Therapy (VST) tools are, and their benefits in therapy, focusing on younger patients. There's no reason to believe that such tools will not be beneficial for the adult patients. However there are no conducted studies that tested VSTs on older patients, and as such there is no scientific basis to back up that statement.

1.1 What are VSTs?

Visual Speech Therapy tools are a type of computer programs used by therapists to aid them in treatment of speech impediments in their patients, for example learning deaf patients proper pronunciation. VSTs have a goal to provide immediate and accurate feedback to users based on their skills, as well as track the progress they make during their treatment. This task is achieved by various means of parsing the audio input and comparing it with pattern of expected pronunciation for a given phoneme, word, phrase, or sentence.

The main problem with VSTs is that there are no up to date programs available on the market, with IBM Speech Viewer III being released in 1995, that have all needed functionalities (Destombes, 1993). There are various small programs available on the market for both computers and tablets, with various useful functions. However none of them are fully satisfactory and often lack various features found in the Speech Viewer program. In addition to these complaints,

very few have support for languages other than English.

There have been some attempts to create more up to date solutions, such as Ortho-Logo-Paedia (OLP) from around 2004. This product showed some promise, but for various reasons has not been completed (Protopapas, 2004).

After conversations with Nina Opdahl, a speech therapist at Narvik municipality, main features needed in an up to date VST tool have been identified and the next sections will describe them in detail.

1.2 Why are VSTs useful?

Various studies have been conducted showing how using VSTs during therapy sessions improve patient's speech patterns. In cases where damage to patients vocal systems have been too high to achieve full recovery, they were successful in stabilizing patients speech. Examples of such extreme impediments are: accident injuries and genetical defects (Sue, 2018; Oster et al., 2003; Thomas-Stonell et al., 2006; Valadez et al., 2012).

Of course VSTs do not function as replacement for a therapy with a qualified speech therapist, and only serve as additional aids in therapy. They can also work as a personal training tool where after a therapy patient would receive exercises from their doctor. They could then go and practice at home with help of feedback from the VST program without need of professional oversight. Although they are meant to be mainly used during therapy sessions with the tutor, allowing therapists to adjust settings of the program, and give patient better feedback.

VSTs aim to give immediate and unbiased results to the end user, allowing them to easily see if they have achieved correct pronunciation. Using visual aids it can also help patients see how they can achieve proper pronunciation, by showing them how to position their tongue and mouth during articulation. Speech therapists would also benefit from getting more technical feedback on patient's speech development, in form of waveform comparison, long-term statistics, consistency rating, etc.

Visual speech therapy tools are a great addition to therapist's toolbelt, allowing them to achieve better results faster than normal. In addition they can help deaf or hearing impaired users "see" their speech components like: pitch, voicing, and general speech patterns, allowing them to properly communicate with others (Wempe and Lunen, 1991; Oster et al., 2002).

1.3 Intended Features and Functionality

There were many principles that IBM used during development of their Speech Viewer I-III programs. Seeing their long lived success, this project aims to be based on the same principles, updating some of them and implementing new ones, in order to create an up to date solution which will help its users to achieve healthy pronunciation.

One of the principles used by IBM France was constant feedback from users, both the therapists and the patients. Based on the conversation with Nina Opdahl and exploration of Speech Viewer and similar programs the following list of desired features was compiled:

1. Efficiency.
2. Ease of use.
3. Adaptability.
4. Useful feedback.
5. Attractive visuals.
6. Interesting and engaging training.

Given how the main focus group are children and teenagers, ease of use is imperative for program's success. It's possible to see this in the way IBM Speech Viewer is designed and organized. For end users all it takes to start training are a mouse click or two, as shown in figure 1.1. That way they will easily get into the flow of training their speech, without needing extensive coaching in using the program. Therapist should also be able to use the software without much trouble, given how not everyone is a computer expert able to tweak a complicated computer system. This task can easily be achieved by cooperating with some therapists on creating predefined exercise sessions that are aimed towards specific training areas, like voicing, pitch, consistency, etc.

Figure 1.1: Screenshot from IBM Speech Viewer III showing overview of exercises available to its users, starting each exercise requires a few mouse clicks. Taken personally from the software.

Lack of adaptability features, was one of the main complaints raised against IBM Speech Viewer (Wempe and Lunen, 1991). This complaint will be easier to solve in modern computer systems, given a better space for storing data, in addition to access to better technological solutions that will be discussed further. A good example shown by Wempe and Lunen was lack of pitch correction and higher sampling range. This made it harder for males to use this program as usually their pitch is below 75Hz, which is the lowest limit for Speech Viewer program. Access to devices capable to record higher fidelity sounds is more common, they are cheaper, and a simple laptop microphone is capable of recording sound of high enough quality to let voice control work in programs like Microsoft's Cortana (Microsoft Corporation, 2018).

To make the feedback useful it needs to be immediate, easy to comprehend, and contain the necessary information for its reader. In order to work as a therapy tool for learning, the software needs to give its users feedback as they articulate given messages. Identifying the mistakes at once, and possibly giving guidance on how to fix them is the most beneficial feedback. Additional feedback forms likeb scoring and long term improvement are also useful.

Audiovisual feedback is the easiest one to understand, as our brains are capable of registering it faster than any other feedback; it has been used in speech therapy systems dating as far back as 1970s (Destombes, 1993). Therefore

combining this feedback with a more long term feedback in form of completion scores, stats, and progress records will give patients necessary feedback, alongside an incentive for continuing training. Therapists need more professional feedback than simple 3 out of 5 stars, more scientific and quantifiable feedback is much more useful for making a decision on how to improve further training. Cooperation with some therapist during design phase of feedback system will be of much help.

In general the program needs to listen to patient's speech and give them feedback dependent on the type of exercise they do. This can be something as simple as speech strength, to more advanced topics like voiced and unvoiced sound detection, pitch control, phoneme sound recognition, etc. This will involve technologies like signal processing, voice recognition, advanced machine learning and graphing.

Making the program attractive, engaging, and interesting might prove themselves to be the biggest challenges, but getting it right will help considerably in its effectiveness (Destombes, 1993). Such visual tool will be helpful for therapists and patients, although it will not serve as replacement for professional help from a trained professional as noted in (Sue, 2018).

1.4 Existing programs

As last part of introduction, it is worth to consider what products are already available on the market, their features, benefits, and limitations. Research into them formed a basis of what the proposed system will try to deliver.

1.4.1 IBM Speech Viewer III

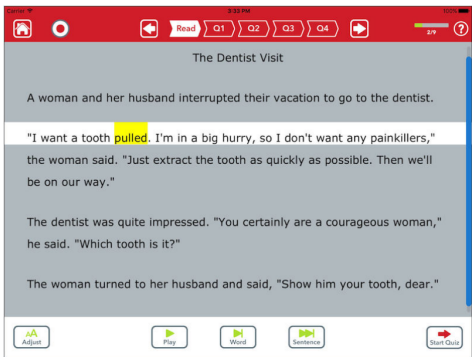
The most advanced tool to date, that is available on the market, providing advanced speech therapy tools is the IBM Speech Viewer. It is however severely outdated and as such proves itself to be quite useless. It requires users to run it on Windows XP with extra patches on a 800x600 display. The graphics that were impressive in 1995 are quite sub par by today's standards. It shows itself to be quite lacking in precision as it has quite low sampling rate and its voiced/unvoiced sound recognition is unreliable, (Wempe and Lunen, 1991). Gameplay elements found in it are also outdated and would require updating in order to be relevant for today's youth.

1.4.2 Phone and Tablet Apps

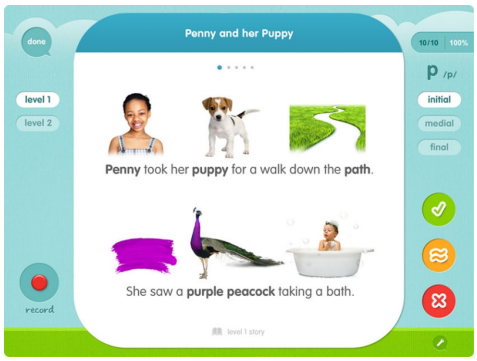
The most advanced application that was found and tested during the research was the Advanced Reading Therapy app (Tactus Therapy Solutions, 2014a). It gives the user a text to read, with option of being read to them so they can hear the proper pronunciation. However this application is far from perfect, as it gives no feedback to the user as they read the text. It lets users record their reading and play it back to them, however that's all it offers in the terms of feedback. Another issue is lack of features that would engage younger audience, not to mention requiring the user to be able to read. As such it's not very child friendly.

There are more child friendly applications on the market, such as Articulation Station that offer more child friendly interface (Little Bee Speech, 2019a). This application uses images and recorded words to help children know what they are supposed to say and how to pronounce it. However they require feedback from the therapist to validate their work, as Articulation Station has the same feedback as Advanced Reading Therapy app, that is recordings of speech.

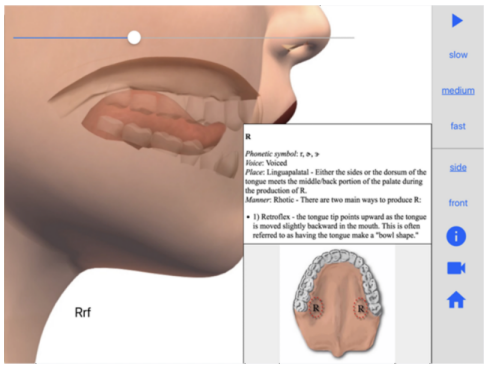
Another useful tool found was the Speech Tutor app, (Synapse Apps, 2008a). It shows the user proper placement and movement of the tongue required to pronounce a given phoneme. Such a tool is very useful for patients that struggle with it, however it is limited by its one use. It has no way of validating if the user started to make proper sounds or not, and it doesn't provide more functionality than that. Example pictures of the aforementioned tools can be seen in figure 1.2.



(a) Screenshot from Advanced Reading Therapy app showing one of its exercises, source (Tactus Therapy Solutions, 2014b).



(b) Screenshot from Articulation Station app showing one of its colorful exercises, source (Little Bee Speech, 2019b).



(c) Screenshot from Speech Tutor app with one of its animations showing proper tongue placement for a given sound, source (Synapse Apps, 2008b).

Figure 1.2: Example pictures from mobile applications for speech therapy found on today's market.

1.4.3 Electropalatography

Although beyond the initial scope of this project it is worth mentioning another technology used in speech therapy called Electropalatography (EPG) (Rose Medical Solutions, 2019b). It uses an artificial palate, usually with 62 electrodes, to determine the placement of the tongue and its contact with the palate during speech. It shows how users place their tongue when pronouncing words. This is important during speech as it changes which sounds are produced, like hard and soft "r" phoneme. Figure 1.3 shows how an EPG system is built. Systems using EPG have been shown to help users improve their speech (Hitchcock et al., 2017).

In addition to audio input the program implementing such a system would be able to determine whether or not the user places their tongue correctly and let them correct their mistakes easily. However given their current price and bulkiness, they wouldn't see much use by normal users, unless prices for this system gets more affordable.

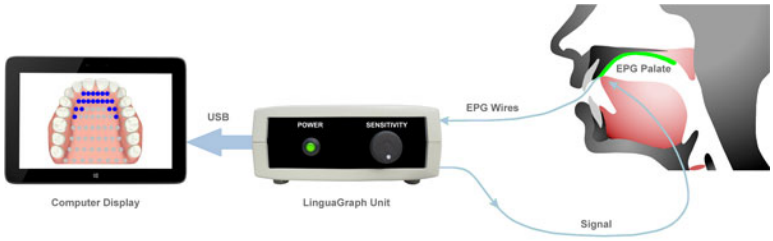


Figure 1.3: Diagram for an Electropalatography system showing how it registers and displays users tongue to palate placement, source (Rose Medical Solutions, 2019a).

2 Methods

Now that merit for VSTs has been established, it's important to see what is needed for such a system to be implemented. In recent years there was an incredible progress in the field of Automatic Speech Recognition (ASR), and there are two main technologies to consider. These two are Deep Learning alogrithms and Hidden Markov Models (Amodei et al., 2016; Rabiner, 1989).

2.1 ASR

There are two main categories of ASR systems. They are called speaker-dependent and speaker-independent. Their main difference is the size of needed vocabulary, how much prerecorded training data it needs to operate properly. As stated by Kitzing et al, a speaker-dependent system requires about 20 words to start working reliably, whereas a speaker-independent system requires at least 20 thousand words (Kitzing et al., 2009).

The sheer size of required data for speaker-independent systems makes it hard to run on its own on small systems like tablets and phones, however it comes with several benefits in form of being able to recognize continuous and spontaneous speech. Whereas speaker-dependent systems require users to read words separately and aloud.

Both systems would work quite well on their own for various applications. A

small self training application for phone would work well with just a speaker-dependent model. On the other hand a proper speech therapy tool would require a speaker-independent system, so it could be able to help users with more complex speech.

2.2 Recurrent Neural Networks

Recurrent Neural Networks (RNN) and its more specialized variant the Long Short Term Memory networks (LSTM) have been used for speech recognition in commercial applications. Examples of systems utilizing them are speech based assistants on smartphones like Siri, and cloud ASR solutions like Google's Cloud Speech-to-Text system.

Recurrent Neural Networks in opposition to traditional Convolutional Neural Networks (CNN) "remember" previous occurrences better, letting them keep persistent memory of past events. This is achieved by feeding results from one layer to previous layers as shown in figure 2.1. One can therefore think of RNN as being a series of copies of the same network, where each copy sends data to its successor.

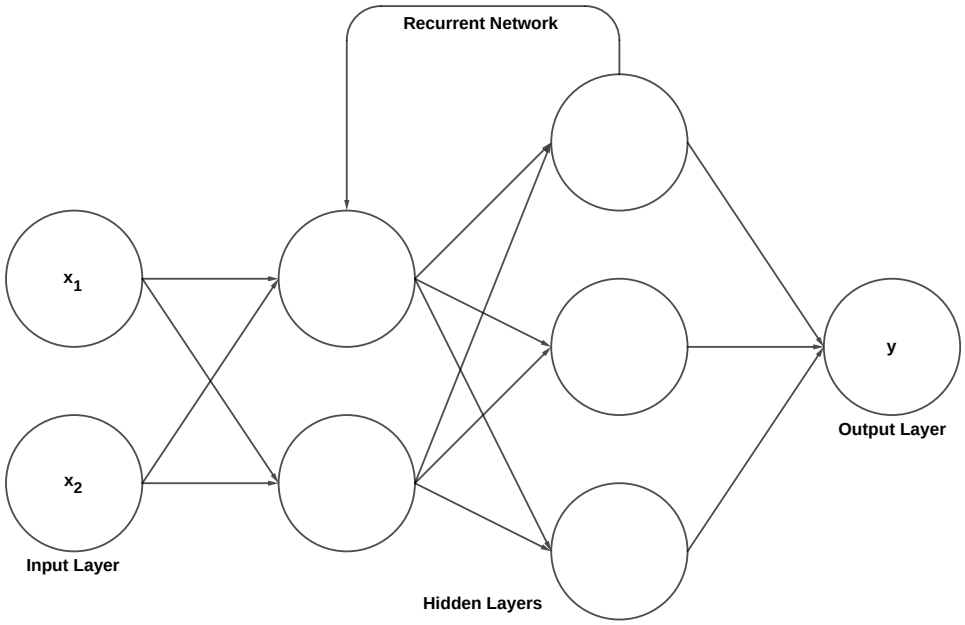


Figure 2.1: Diagram showing basic structure of Recurrent Neural Network.

These networks are able keep that information using hidden state variable h

that is calculated using equation (2.1). Where t is the timestep for which h is calculated, tanh is the hyperbolic tangent function, W is the weight matrix for each element in the input vector x, and I is the projection matrix.

$$h_t = \tanh\left(Wh_{t-1} + Ix_t\right) \tag{2.1}$$

Using that variable h, nodes in the network are able to produce output vectors y by using W and h for each timestep t, as shown in equation (2.2). Softmax function creates a normalized probability distribution for each existing class in the network (Pascanu et al., 2014).

$$y_t = \text{softmax}\left(Wh_{t-1}\right) \tag{2.2}$$

It also beats CNNs by the fact that it's not fixed to a specific size of input and output vectors, there can be systems with one to one connection, as wells as many to one, one to many, and many to many. In Recurrent Neural Networks sequences of vectors are the main operating units. This allows them to be adjusted for specific needs, like image captioning, machine translation, speech recognition, etc. However some of these tasks require more specialized version of RNN, which are the Long Short-Term Memory networks.

2.3 Long Short-Term Memory Networks

Long Short-Term Memory networks are a subcategory of RNN where they aim to fortify the short-term memory of the network, letting it remember "small details" longer. In the task of speech recognition such long term memory can help the network in understanding speech based on previous utterances from the user. The ideas for such networks were first introduced by Hochreiter and Schmidhuber (Hochreiter and Schmidhuber, 1997). Their main concepts and general structure are the same as in standard RNNs, however where each node in RNN has a single layer, like tanh, LSTMs have much more complex structure, as can be seen in figure 2.2. Thanks to this memory for details it can predict the next words in sentences based on previous information. For example "I was born in France ... as such I know how to speak *French*", where "*French*" would be predicted by the network based on knowledge of the person being born in France.

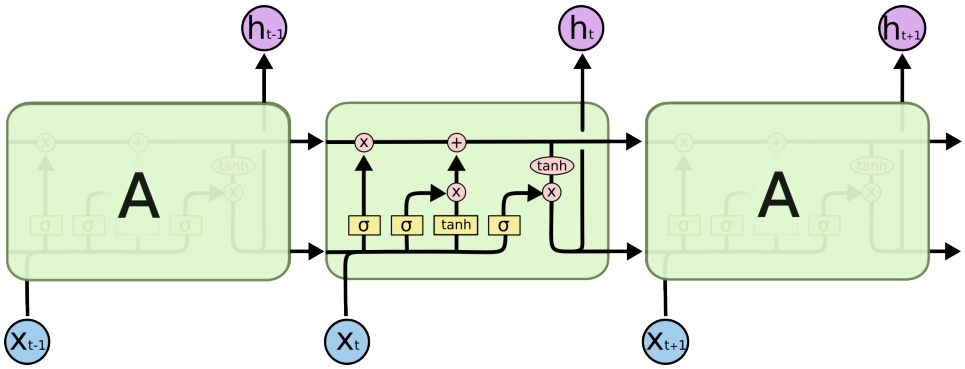


Figure 2.2: Diagram showing basic connection structure of Long Short-Term Memory nodes, it visualizes how data flows between nodes for each timestep, source (Olah, 2015b).

Shown in figure 2.3 is an example of how each node receives input for each timestep x_t, along with old cell states c_{t-1} and the old hidden state variable h_{t-1}. It then calculates the f_t at the "forget gate layer", then it calculates i_t at the "input gate layer", and lastly it calculates new candidates for cell states $\widetilde{c}_t$. Using these new values it updates the old cell states c_{t-1}, calculates the output o_t and the new hidden state h_t. How each of these values are calculated is shown in equation (2.3) (Olah, 2015c).

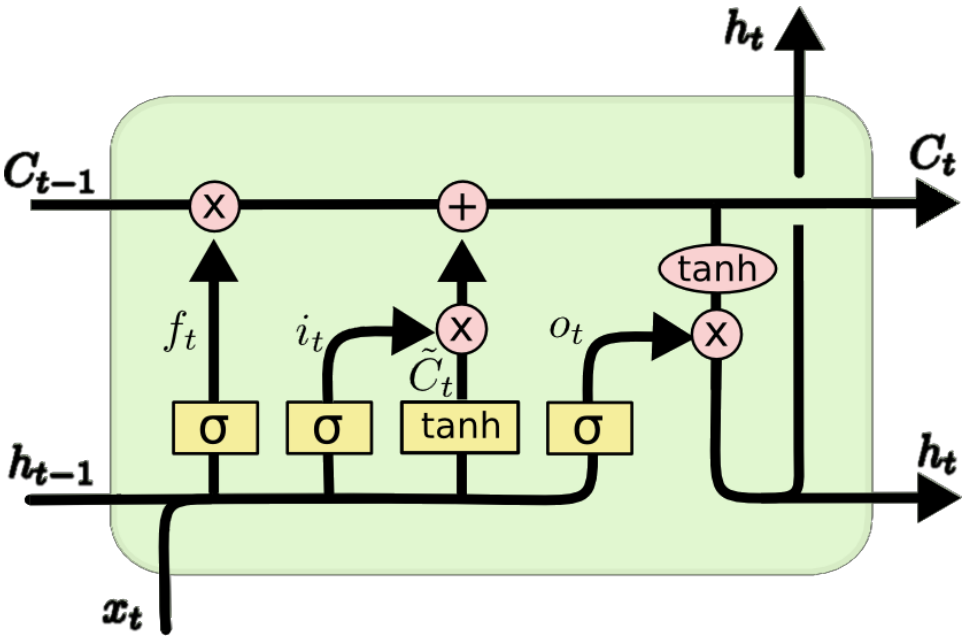


Figure 2.3: Diagram visualizing how data in an LSTM network node is processed, where each value in the node is calculated, and where it is sent, source (Olah, 2015a).

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\widetilde{c}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\
c_t &= f_t \times C_{t-1} + i_t \times \widetilde{c}_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t \times \tanh(c_t)
\end{aligned}
\tag{2.3}
$$

All variations of W in these equations are weight matrices, and all variations of b are bias vectors, all of them are initialized with random numbers, and are updated as the network is training. σ is a simple sigmoid function. One special case is the b_f vector, or "forgetfulness bias vector", which shouldn't be initialized with random values, but with one's or two's as suggested by Józefowicz et. al. to avoid a vanishing gradient problem (Józefowicz et al., 2015).

This kind of network setup gives LSTM and its variations a great capability in recognizing human speech and parse the words they are saying. Which is great for implementations like speech commands, transcripts, captioning, etc. However it is not that well adapted to the the application in VSTs. These programs don't need to recognize what its user is saying, as they are saying words and sentences that the program asks for. What is needed is the ability to compare what the user said to the baseline speech in healthy individuals and other patients with similar problems. It is needed in order to be able to give proper feedback on correctness of the speech.

Another issue with LSTM based ASR systems is that they require wast amounts of computing power and data to operate properly. Systems like Google's Cloud Speech-to-Text requires big data centers with multitude of servers running on multiple GPUs to work efficiently. Of course such a system is vastly more complex than what is needed for a VST system. LSTM networks provide great results, but sadly not the results that a VST system needs. As such a LSTM based solution isn't well adapted to the problem at hand.

2.4 Hidden Markov Models

Hidden Markov Models (HMM) are purely mathematical structures that are deterministic in nature and when applied properly work remarkably well for speech recognition tasks. They have the same benefits of being trainable as LSTMs are. Given their comparatively less complex structure, they require less resources to work than LSTM networks. Where modern solutions involving

RNNs are offloaded to clouds, a HMM based system can be operated on a stationary computer.

Discrete Markov Process that HMMs are based on considers a system of N distinct states S, where the system undergoes state change at fairly regular time intervals t. These state changes happen according to some set of probabilities associated with the state. A simple representation of the probabilistic system described above would look like equation (2.4), where q_t denotes the actual state at time t. Such a Markov chain is called a First Order, Discrete Markov chain (Uchat, 2012).

$$P(q_t = S_j | q_{t_1} = S_i, q_{t_2} = S_k, \dots) = P(q_t = S_j | q_{t_1} = S_i) \tag{2.4}$$

Considering processes where right hand of equation (2.4) is independent of time, then a set of state transition probabilities $a_{i,j}$ can be found, defined by equation (2.5). These state transitions have given properties $a_{i,j} \geq 0$ and where the sum of all state change probabilities is equal to 1. In short there are various probabilities associated with the state that dictate which state change is most likely to occur. For example if the sentence is "UiT the Arctic University of ..." then "Norway" would be the most likely candidate, based on previous states of the sentence. Whereas "I like to eat ..." have many different candidates, and the probability for one or the other state change is hard to calculate, without any previous knowledge to go on.

$$a_{i,j} = P(q_t = S_i | q_{t_1} = S_j) \quad 1 \leq i, j \leq N \tag{2.5}$$

Extending that idea further to Hidden Markov models makes the observations themselves a probabilistic function of the state. In this case it's important to know what state in the model corresponds to what and how many states there are. For example in the case of simple coin flip there are two states, where one corresponds to a coin landing on heads, and the other on tails. Using that idea the elements of HMM are as follows:

N: number of states.

M: number of distinct observation symbols per state.

V: observation symbol.

$\boldsymbol{a_{i,j}}$**:** state transition probability as defined in equation (2.5).

$B_j(K)$: probability distribution for symbol observation for each state, $B_j(K) = P(V_k \text{ at } t | q_t = S_j)$.

π: the initial state distribution equal to $\pi_i = P(q_1 = S_i) \quad 1 \leq i \leq N$.

There are three well known problems with Hidden Markov Models that have elegant solutions. The first problem being the evaluation problem, which revolves around efficiency of calculating the probabilities of observation sequence O given the model $\lambda = (A, B, /pi)$. This problem is solved by using a Forward Algorithm, described in depth in (Uchat, 2012) and other places. In short instead of enumerating over each possible state sequence $O = O_1, O_2, \ldots, 0_T$, it uses a forward variable $a_t(i)$ defined in equation (2.6) which calculates probability of a partial observation of sequences from O_1 till O_i and state S_i at time t given the model λ. Then using inductive calculations system requires to do only $N^2 \cdot T$ calculations instead of N^T calculations (Rabiner, 1989).

$$a_t(i) = P(O_1, O_2, \ldots, O_t, q_t = S_i | \lambda) \tag{2.6}$$

The hidden state determination problem revolves around decoding, where for observation sequence $O = O_1, O_2, \ldots, 0_T$, and model $\lambda = (A, B, /pi)$, how does the program chooses the most meaningful state sequence $q = q_1, q_2, \ldots, q_T$? The Viterbi Algorithm aims to answer that problem, by helping HMMs find the best state sequence in a dynamic way. Where to find the best state sequence q for observation sequence O as defined before, one defines quantity $\delta_t(i)$ as shown in equation (2.7). This variable allows deduction of the sequence along the path leading to state S_1 by backtracking for the maximal state S. It works similarly to the Forward Algorithm described above, except for backtracking, the major difference is the finding maximum state instead of addition of all states as is done in the Forward Algorithm (Uchat, 2012; Rabiner, 1989).

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(q_1 q_2 \ldots q_t = i, O_1 O_2 \ldots O_t | \lambda) \tag{2.7}$$

The last problem is the problem of learning. How should the system adjust variables A, B, and π for model λ, so that it maximizes $P(O|\lambda)$ for training sequences. The problem in this case being no efficient ways of finding global optimum. The Baum-Welch Algorithm solves that problem by use of heuristic procedure for finding local optimization instead of global optimization. This time the procedure starts by defining a variable $\xi_t(i, j)$, which defines probability of transitioning from state S_i to S_j in a single timestep, given observation sequence O and the model λ. Definition of this variable is shown in equation (2.8). Formal explanation of this procedure is quite lengthy, however it is

a well understood procedure, therefore for brevity it is omitted, see (Rabiner, 1989) for more details.

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \tag{2.8}$$

Thus using these three algorithms, the HMM systems are able to fix the three main problems. Combining them into one procedure yields a convergence algorithm, as shown below.

1. Initialize $\lambda = (A, B, \pi)$.
2. Compute a, δ, and ξ.
3. Estimate $\bar{\lambda} = (\bar{A}, \bar{B}, \pi)$.
4. Replace λ with $\bar{\lambda}$.
5. If the result has not converged, go back to step 2.

After long consideration, a HMM based system was found to be best suited for the needs of a VST program, even though it has been supplanted by RNNs in other applications. This conclusion is based on the lightweight computation in comparison to LSTM systems, ease of finding similarity score that will be discussed further in the document, as well as success of projects based on similar technologies, for example Ortho-Logo-Paedia (Protopapas, 2004; Oster et al., 2002, 2003).

3

Design of VST Using HMM

This chapter will present proposed design of a Visual Speech Therapy tool based on tools and methods described in Methods chapter. It will focus on how ASR system for speech therapy differs from its usual speech recognition design, how the VST program should be designed, developed, and used.

3.1 ASR Structure for Speech Therapy

Structure for an ASR system used in Speech Therapy differs a bit from a system used in pure speech recognition. As speech recognition systems need to calculate or deduce what words the user is speaking, a speech therapy system knows in advance what the user is going to say, as it chooses it for the user. Therefore some parts of the system are not necessary or work differently than in normal ASR systems.

ASR systems start with acoustic speech soundwaves being transformed into electric currents by the microphone. Said current is then adjusted for constant volume, and filtered to avoid noise and aliasing which might skew the results. After that the analog signal is converted to a digital signal using typical sampling frequencies (25 kHz, 16 bit) and is then framed using Hamming Windows. Frames are then acoustically analyzed and using Mel Frequency Cepstral Coefficients (MFCC) and Linear Predictive Coding (LPC) feature vectors are extracted from the processed signal. Feature vectors undergo categorization

according to their spatial qualities and are coded to reduce the amount of data in them, allowing for faster computation and easier pattern matching. As exemplified by Kitzing (Kitzing et al., 2009), data sampled at 16 kHz gives a window size of 512 samples that result in 13 MFCCs. With their first and second order derivatives program produces feature vectors with dimension of 39. Of course these variables might vary depending on sampling frequency and window size choices.

Using Gausian Mixture Models (GMM) along with HMMs, one can calculate similarity for signals, compared to the training speech corpus. These are then used to form phone sequences, that form the basis for full words and sentences. Using the available data for the VST program, in form of knowing which phones used should be form and probable mispronounced phones, one can quickly find out if the user pronounced the words or sentences correctly, how correct they were, and point out eventual mispronunciations at the phone level. For ease of understanding the flow, figure 3.1 shows how data would flow in the proposed ASR system.

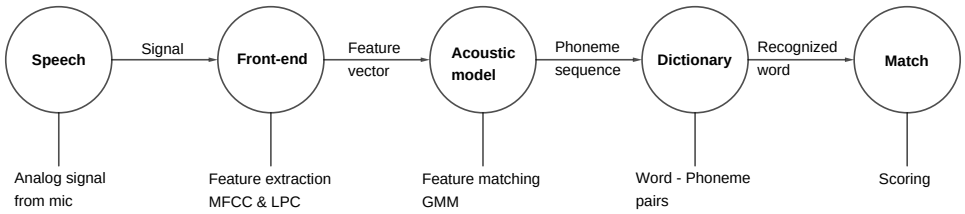


Figure 3.1: Diagram showing data flow through ASR system for VST programs, from microphone to the scoring of utterance by user.

Norwegian language has 44 phonemes and those serve as building block for the whole spoken language, each word in the Norwegian dictionary can be made out of these phonemes (Hay, 2019). In order to function correctly the ASR system needs to be able to recognize these phonemes and correctly concatenate them to form words and sentences. Creating a simple and unique HMM for each of those phonemes and stringing them together is the usual way of creating ASR programs. Example of a structure for a HMM used to recognize a single phoneme is shown in figure 3.2 and chaining them together forms words.

As noted by Uchat and others, when the system looks at small samples of about 10 to 25 msec the characteristics of a speech signal are almost stationary, letting the HMMs validate each part of the phoneme easily (Uchat, 2012). In the case of VST the task is simplified. The program doesn't require language context recognition for words with similar or identical phonemes, like the English "car keys" and "khakis", since it only needs to measure what's been said to what it expected.

One problem arises in the case of Norwegian language as its speakers have many dialects where pronunciation of words can vary drastically between regions in Norway. Easiest solution in the beginning would be to focus on pure bokmål dialect, and then when the system gets bigger and more advanced, include options for some of the most used dialects. That would require of course getting training data for each of them.

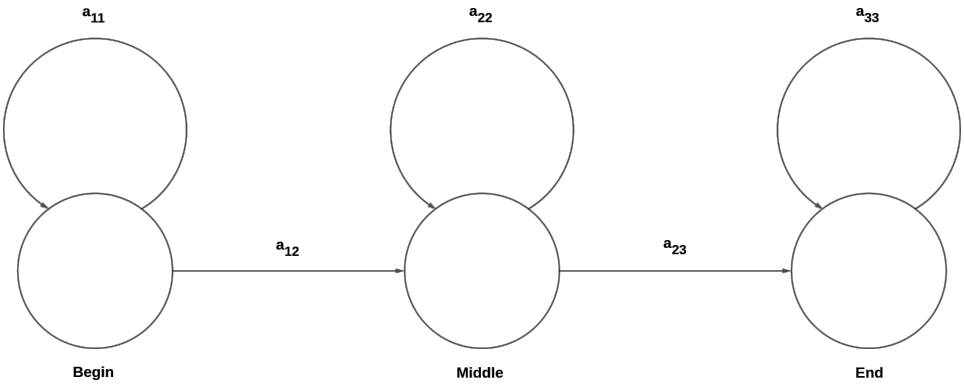


Figure 3.2: Diagram showing how a HMM for a single phoneme is built, each phoneme has a begin, middle and end part. Stringing them together into a long chain allows for recognition of each word in the spoken language.

In addition to how relatively easy it is to set up HMM systems for ASR, they have a benefit of not needing that much data to be trained letting the system train itself on speakers own speech. Of course a system trained on such data will not be perfect for all cases, however it's been shown to work remarkably well in VSTs. This has also the added benefit of being adjustable to speakers with serious speech impediments that make their speech unrecognizable, or very hard to recognize. The program would then be able to better measure improvements in patient's speech patterns, letting them stabilize their speech, maybe even improve it (Oster et al., 2002).

Another benefit of ASR systems based on HMMs is their ability to supply scoring for their performance, which can be used in speech therapy, computer assisted language learning, etc. This is useful for applications where self monitoring would not be enough to improve, as a significant number of patients is not able to hear their mispronunciations. This can be caused by their hearing problems or inability to differentiate between correct and incorrect speech, (Kitzing et al., 2009).

3.2 VST Design

After long and careful consideration, online solution has turned out the be better than a local solution. This is based on factors like ease of expansion, where client and server side of the program can be developed and extended separately. This would allow the more expensive calculations to run on servers capable of handling the more expensive calculations, without requiring users to invest in high performance equipment. It would also allow for easier portability of the program to systems other than computers, like tablets and mobile phones, as it would not require to have all the needed data samples be saved locally on user's device.

Building the VST program in such a way allows users to easily continue training at home, as their profiles with progress, scores, and exercises would be kept in the database. Users would need to consent to having some of their data be kept in the database, in order to comply with General Data Protection Regulation (GDPR) set forth by the EU. However the benefits resulting from a program capable of correcting their speech should outweigh any objections they could have for sharing some basic information with the system. Of course data kept by the program would be minimal, nothing more than some identification, general information about gender, age, and speech impediments, as well as voice samples used for training and evaluation.

Added benefit of utilizing the client server structure would be ease of implementing new features, either on the client side, in form of new exercises, or on server side, in form of improved and new functionalities. It would allow the big and heavy parts of the program to be run and changed, without requiring users to constantly update the program. It would also serve to help in delivering uniform service for users.

Of course there are drawbacks to such a solution, main one being the fact that it would require internet connection to operate. However given how internet is more and more used and accessible, and the internet service in Norway is constantly improving, the benefits will far outweigh the drawbacks of such a solution.

Figure 3.3 shows how such a system would be built up. On the client side patients would interact with GUI based exercises, akin to IBM Speech Viewer's exercises, although heavily overhauled and updated. An example of such an exercise could be a "Jack in the Box" game, where for each correct pronunciation a green box is activated with a happy clown, or unhappy one for bad pronunciation, see figure 3.4. Patient's speech signal would be picked up on the microphone, after which it would go through a simple pre-processing phase, where it would be cleaned up and sampled to a digital signal, which could

then be transferred to the server where features could be extracted, matched, decoded, and scored. After which the server would send back the scoring and other data necessary for giving feedback to clients.

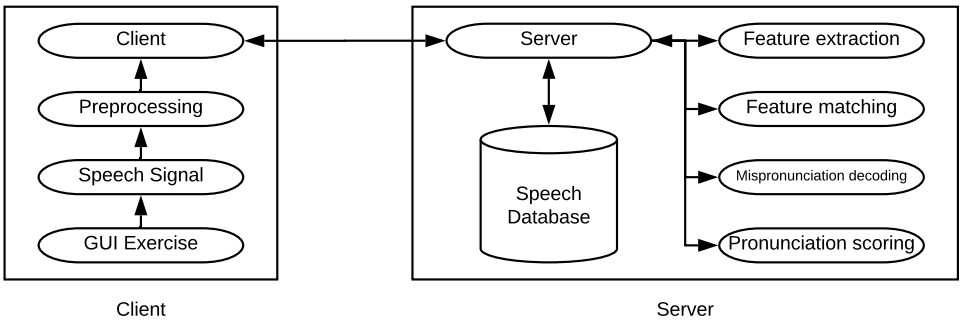


Figure 3.3: Diagram showing system design for the proposed VST system, where client and server sides of the program and data flow are visualized.

A speech database would be built up from samples gathered from users, letting the program to be adjusted to specific speech impediments, as well as samples of properly pronounced words. Most of the records would be kept as feature vectors for training the HMMs, however some can be kept as audio files. Audio samples would be used for audible comparison by therapists or users, so they can hear their improvements after therapy.

Figure 3.4: Example picture showing how an exercise in VST program could look like, with the program reacting based on user's speech.

3.3 Program Interaction

When it comes to general interaction between VST, therapists, and patients it would happen as shown in figure 3.5. The therapist would start by registering and configuring the system for specific needs of the patient, as each pathology be it dysarthia, craniofacial disorders, hearing impairments, or general functional articulation disorders, require specialized training (Oster et al., 2002). There would be a set of pre-programmed exercises and settings, custom made for each disorder in cooperation with therapists It would allow everyone to use the system easily, though maybe not optimally, with more advanced users being able to tweak setting to their needs. Therapists would be welcome to create their own training sets and share them online with others. Such a system would make it significantly easier for new therapists and patients to start using it, as it wouldn't require specialized training to use, letting them learn more advanced features as they go.

After choosing a set of exercises, the VST would require some training to adjust itself to patients speech, which would also incorporate a tutorial for the patient. After adjustments, the patient could begin training on the exercises with the therapist, getting feedback from both human and computer, helping them learn proper pronunciation. Rewards in form of victory animations, achievements, better scores, maybe even a leaderboard, would help to motivate the end user to continue training. Exercises themselves would be adjusted accordingly to not only include the type of disorder that the patient suffers from, but also their age, gender, reading level, etc. as well as level of complexity, from phones to full sentences. Exercises would also be made available as simple picture based exercises for patients too young to read.

After training sessions therapists would be able to access more detailed data from the system. The data would be overall scoring, error levels and error rates, speech consistency, articulation speed, waveform graphs, etc. Such data will be useful for therapists to see what they need to focus on more in the next training sessions, as well as what they can congratulate their patients on, using it as positive reinforcement and incentive to continue training. In future when the system implements support for EPG they could also see the tongue to palate orientation, helping them give better therapy. Having support for uploading own training data in form of words and sentences along with speech samples would help them customize the training to their own style, as well as help develop the program further. Tests done by Öster et al (Oster et al., 2003) show that even small sample data from which to train the system, allows HMM based programs to correctly recognize speech, and give useful feedback to users.

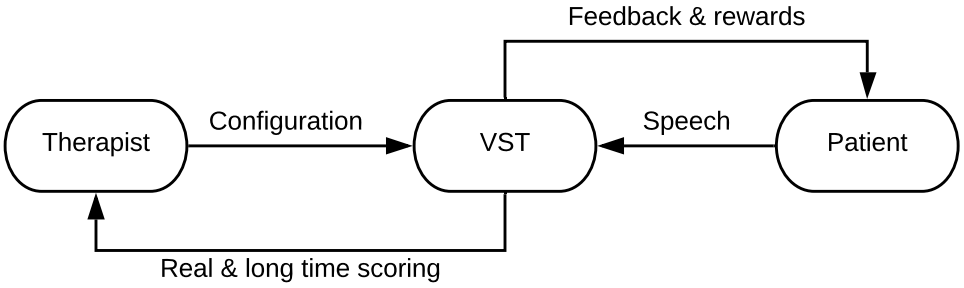


Figure 3.5: Diagram showing how users would interact with proposed VST system.

3.4 Technology

Technology choice for the application is based on ease of use and performance, as well as security. Therefore some potentially useful or enticing technologies will not be used. When considering what technologies would be used in creation of the program, speed of operation and development was weighted agaist each other.

Most of the application can be implemented in Python programming language, as it offers good balance between performance and ease of use, however critical parts of application that need to operate as quickly and efficiently as possible will be programmed using C++. Main areas where C++ will be useful are the signal processing and HMM training, as it offers splendid performance in comparison to Python (Stein and Geyer-Schulz, 2013). When it comes to the GUI part of the program there are a lot of tools to choose from, for example PyQt5 (Riverbank Computing Limited, 2018) which allows Python programs to access the Qt (The Qt Company, 2018) framework built on C++ which offers C++ performance and Python straightforwardness. Another possible framework for creating 3D graphics in Python is Panda3D (Carnegie Mellon University, 2018).

Client/Server communication will not use pure User Datagram Protocol (UDP), but instead Datagram Transport Layer Security (DTLS) protocol (Rescorla and Modadugu, 2012). DTLS is beneficial in its guarantee of security and reliability, whilst still utilizing UDP as basis, making it easier to stream data like audio files to and from server. This comes at the speed benefit in comparison to Transmission Control Protocol (TCP) as it avoids both the handshake phase and the so called "TCP Meltdown" which can significantly impact throughput of data in a program. (Honda et al., 2005).

4

Results & Discussion

This chapter will address how the proposed system has potential to be better than existing products. It will compare several important aspects and features of VST systems. The list of features is based on previous research into qualities in IBM Speech Viewer, showing what functioned well, and what has left something to be desired out of the program. Lastly it will compare existing solutions, and possible capabilities of the proposed system to features of IBM Speech Viewer, whether or not they are better, worse, or the same.

There were several functionalities that IBM Speech Viewer was measured up against in studies done by Destombes, Wempe and Lunen (Destombes, 1993; Wempe and Lunen, 1991). These were: sampling rate, frequency range, word recognition, pitch range, voicing recognition, loudness, voice onset, and feedback. After explaining what they are and how they were measured it will be possible to set up comparison between IBM Speech Viewer and other products.

Sampling rate refers to conversion of analogue speech signal to a digital signal. Speech Viewer has a sampling rate of 9.6kHz, 8-12bit, whereas proposed solution can sample sound at 25kHz 16bit while still keeping the feature vector at a manageable size. Of course it does require testing to see if a built in microphone in iPad, laptop, or smartphone is capable of providing good enough data for the program. However standards for HMM based systems in the middle of 2000s already beat the IBM Speech Viewer, which had offered no more than telephone quality sampling (Wempe and Lunen, 1991).

Frequency range that was supported by IBM Speech Viewer is between 75-960Hz, which led to several problems as males tend to produce pitches lower than 75Hz. It has also relied too much on lower frequencies in detection of voiced and unvoiced sounds, which according to Wempe and Lunen does not occur so often, especially in speech produced by healthy subjects (Wempe and Lunen, 1991). These problems resulted in some parts of the program giving random results, instead of reliable feedback. Modern systems have built in microphones that support much larger frequency ranges, typically from 10-50Hz to 10-20kHz, which will help to combat these issues.

As noted by Destombes newer versions of IBM Speech Viewer came with support for recognition of simple words, however it was quite limited and required very good conditions and microphones to work reliably (Destombes, 1993). Given how HMM systems were reliable in not only recognizing simple words, but also complex word and even continuous speech recognition, the proposed system is expected to have a clear advantage over Speech Viewer and other basic programs on the market.

Voicing recognition was seriously flawed in IBM Speech Viewer, where even trained voices of healthy individuals were not properly recognized (Destombes, 1993; Wempe and Lunen, 1991). Program was based on technology that measured signals containing high energy in lower frequency range of speech that were either not picked up in male individuals, or weren't strong enough as healthy speech have steeper glottal pulses that result in lower energy waves. Given how HMM is based on phonemes and their pronunciation, the system based on it should be much better at recognition of voiced and unvoiced phonemes.

Methods used in measuring loudness of the voice, although sufficient, were not correct in IBM Speech Viewer (Wempe and Lunen, 1991). When it comes to this point there's no data that was available at the moment to test whether or not the proposed system would fare better. Although measuring loudness of the voice does not seem to be hard task to do, given the technology available today, which has significant advantages over IBM Speech Viewer.

Voice onset tests measured by Wempe and Lunen showed good response rate in IBM Speech Viewer. It is quite reasonable to assume that the same will apply for the proposed system, as it only needs to measure whether speech is continuous and kept at a relatively stable level.

Feedback in the case of exercises and results left something to be expected. Speech Viewer either didn't register properly the user speech, or visual feedback wasn't helpful in correcting speech. Spectral graphs were so unstable that their usefulness was very limited (Wempe and Lunen, 1991). Tracking of progress

needed to be done manually, and depended much on therapists own judgment and feedback (Destombes, 1993).

Putting all of these quantifiable metrics in a comparison table results in table 4.1. Both the proposed system and currently available applications were compared against IBM Speech Viewer. In case of the current programs Advanced Reading Therapy, Articulation Station, and Speech Tutor were considered. For each point on the table scoring was given based on the best program available. It is important to note that these programs were not tested rigorously, simple feature tests were carried out to see how these application function. These tests were mostly focused on seeing whether any of these applications contain same, or better features than IBM Speech Viewer.

Functionality	**Current Mobile Apps**	**Expected Functionality**
Sampling rate	Better	Better
Frequency range	No Data	Better
Word recognition	Worse	Better
Pitch range	Better	Better
Voicing recognition	None	Better
Loudness	None	No data
Voice onset	None	Better
Feedback	Worse	Better

Table 4.1: Comparison table where currently available applications, and proposed solution were measured against IBM Speech Viewer. Based on studies by Destombes, Wempe, and Lunen on IBM (Destombes, 1993; Wempe and Lunen, 1991).

Based on these findings the proposed system shows promise in being a good replacement for IBM Speech Viewer, and a strong opponent to the currently available products.

5

Conclusion

In this chapter final thoughts about the feasibility of the proposed product will be discussed, along with challenges its development might face, and lastly some proposed future developments.

Even though there have been many various tools developed and used by speech therapists, from cathode tubes in 1946 (Destombes, 1993), through Speech Viewer, and modern apps, not one of them has been proved to provide the necessary functionality needed to perform effective therapy. Although some may be very good in one specific area of therapy, research has not found a one-in-all solution for therapy. While Speech Viewer is packed full of useful features, they are either outdated or unreliable, and are hard to use on modern machines. Newer programs lack several vital features, most notably some form of immediate and objective feedback, in addition to often being English only tools.

Developing a tool capable of combining those features and presenting them in an easy to use and engaging manner is something worth pursuing, as it would give therapists and their patients a vital tool in therapy. Of course it won't replace a human interaction and tutoring, but it will be a handy addition to employ in helping people with speech impediments.

As it has been shown in the Results & Discussion chapter, the proposed application has possibility to be better than IBM Speech Viewer and currently available applications. Although it's hard to guarantee success of this program based

only on the comparison of technological differences and capabilities, it is worth mentioning that Ortho-Logo-Paedia that was based on similar technologies had very good results (Oster et al., 2002, 2003). This leads credence to the feasibility of this project succeeding in the long run.

There are several questions that are still worth answering. One of them being system requirements like microphones, processing speed, recording quality, etc. Answering them will necessitate development of a simple application based on the proposed structure system, and testing how well it will operate. The proposed way forward is to use that simple program as a starting place. Using similar workflow to development of IBM Speech Viewer will allow creation of a tool that will be easy to use and help its users to get better results faster than using old therapy methods (Destombes, 1993). In this workflow patients and therapists were actively involved in development, giving constant feedback and ideas for improvements.

After a working computer program is developed, then new features should be implemented depending on the needs voiced by its users. These features could be a support for EPG systems, mobile application port, or implementation of LSTM-FCN based speech recognition system for possible spontaneous speech feedback.

LSTM-FCN or Long Short-Term Memory Fully Convolutional Network has already been tested in the speech therapy area where Pishgar et. al. has developed a system for classifying pathological voice conditions from speech samples (Pishgar et al., 2018). The system proposed by them mixed Mel-Cepstrum Vectors and Support Vector Machines and had performed quite well with 88% sensitivity score, even though it used quite small training sample.

In conclusion the system based on Hidden Markov Models for speech recognition, along with other help functions shows a promise in being an invaluable tool for modern speech therapy. The proposed system structure model allows easy development and portability to mobile applications. It remains to see how successful the finished product will be, however results of the research lead me to believe that with hard work and dedication it will succeed.

Bibliography

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L. V., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., and Zhu, Z. (2016). Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 173–182.

Carnegie Mellon University (2018). Panda3D Open Source Framework for 3D Rendering & Games. https://www.panda3d.org/. Accessed on 2019-06-04.

Destombes, F. (1993). *The Development and Application of the IBM Speech Viewer*, pages 187–196. Springer-Verlag Berlin Heidelberg.

Hay, M. (2019). What Languages in Europe Have the Most or Least Phonemes? https://www.eupedia.com/linguistics/number_of_phonemes_in_european_languages.shtml. Accessed on 2019-03-01.

Hitchcock, E. R., McAllister Byun, T., Swartz, M., and Lazarus, R. (2017). Efficacy of Electropalatography for Treating Misarticulation of /r/. *American Journal of Speech-Language Pathology*, 26(4):1141–1158.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural computation*, 9:1735–80.

Honda, O., Ohsaki, H., Imase, M., Ishizuka, M., and Murayama, J. (2005). Understanding TCP Over TCP: Effects of TCP Tunneling on End-to-End Throughput and Latency. *Proc SPIE*, 104.

Józefowicz, R., Zaremba, W., and Sutskever, I. (2015). An Empirical Exploration of Recurrent Network Architectures. In *Proceedings of the 32Nd International Conference on Machine Learning - Volume 37*, volume 37 of *ICML'15*, pages 2342–2350. JMLR.org.

Kitzing, P., Maier, A., and Lyberg Åhlander, V. (2009). Automatic Speech Recognition (ASR) And Its Use as a Tool for Assessment or Therapy of Voice, Speech, and Language Disorders. *Logopedics, phoniatrics, vocology*, 34:91–6.

Little Bee Speech (2019a). Articulation Station Pro. http://littlebeespeech.com/articulation_station_pro.php. Accessed on 2019-02-15.

Little Bee Speech (2019b). Articulation Station Screenshot. http://littlebeespeech.com/images/ac_level_1_story_full.jpg. Accessed on 2019-02-15.

Microsoft Corporation (2018). Personal Digital Assistant - Cortana. https://www.microsoft.com/en-us/cortana. Accessed on 2019-06-07.

Olah, C. (2015a). Annotated LSTM Node Diagram. http://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-focus-f.png. Accessed on 2019-05-29.

Olah, C. (2015b). LSTM Node Diagram. http://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-chain.png. Accessed on 2019-05-29.

Olah, C. (2015c). Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed on 2019-05-29.

Oster, A.-M., House, D., Hatzis, A., and Green, P. (2003). Testing a New Method for Training Fricatives Using Visual Maps in the Ortho-Logo-Paedia Project (OLP). *PHONUM: reports in phonetics,*, 9:89–92.

Oster, A.-M., House, D., Protopapaa D, A., and Hatzis S, A. (2002). Presentation of a New EU Project for Speech Therapy: OLP (Ortho-Logo-Paedia). *Proceedings of Fonetik TMH-QPSR*, 44:45–48.

Pascanu, R., Gülçehre, C., Cho, K., and Bengio, Y. (2014). How to Construct Deep Recurrent Neural Networks. *Computing Research Repository*, abs/1312.6026.

Pishgar, M., Karim, F., Majumdar, S., and Darabi, H. (2018). Pathological Voice Classification Using Mel-Cepstrum Vectors and Support Vector Machine. *IEEE International Conference on Big Data*, pages 5267–5271.

Protopapas, A. (2004). Ortho-Logo-Paedia. http://www.xanthi.ilsp.gr/olp/summary.htm. Accessed on 2019-05-20.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286.

Rescorla, E. and Modadugu, N. (2012). Datagram Transport Layer Security Version 1.2. RFC 6347, Internet Engineering Task Force. Accessed on 2019-06-04.

Riverbank Computing Limited (2018). What is PyQt? https://www.riverbankcomputing.com/software/pyqt/intro. Accessed on 2019-06-04.

Rose Medical Solutions (2019a). Electropalatography Diagram. http://www.rose-medical.com/images/how-electropalatography-works.jpg. Accessed on 2019-02-15.

Rose Medical Solutions (2019b). Electropalatography EPG System. http://www.rose-medical.com/electropalatography.html. Accessed on 2019-02-15.

Stein, M. and Geyer-Schulz, A. (2013). A Comparison of Five Programming Languages in a Graph Clustering Scenario. *Journal of Universal Computer Science*, 19(3):428–456. Accessed on 2019-06-04.

Sue, A. L. (2018). This Review of Virtual Speech Therapists for Speech Disorders Suffers from Limited Data and Methodological Issues. *Evidence-Based Communication Assessment and Intervention*, pages 1–6.

Synapse Apps (2008a). Speech Tutor Pro. https://www.speechtutor.org/app. Accessed on 2019-02-15.

Synapse Apps (2008b). Speech Tutor Screenshot. https://bit.ly/2WlIImb. Accessed on 2019-02-15.

Tactus Therapy Solutions (2014a). Advanced Reading Therapy. https://tactustherapy.com/. Accessed on 2019-02-15.

Tactus Therapy Solutions (2014b). Advanced Reading Therapy Screenshot. https://bit.ly/2Jz6epH. Accessed on 2019-02-15.

The Qt Company (2018). Qt cross-platform software development for embedded & desktop. https://www.qt.io/. Accessed on 2019-06-04.

Thomas-Stonell, N., Hunt, E., and McClean, M. (2006). Evaluation of the Speech Viewer Computer-Based Speech Training System with Neurologically Impaired Individuals. In *Journal of Speech-Language Pathology and Audiology*, volume 15, pages 47–56.

Uchat, N. S. (2012). *Seminar Report on Hidden Markov Model and Speech Recognition*. Department of Computer Science and Engineering Indian Institute of Technology, Bombay.

Valadez, V., Ysunza, A., Ocharan-Hernandez, E., Garrido-Bustamante, N., and Sanchez-Valerio, A. (2012). Voice Parameters and Videonasolaryngoscopy in Children With Vocal Nodules: A Longitudinal Study, Before and After Voice Therapy. *International Journal of Pediatric Otorhinolaryngology*, 76(9):1361–1365.

Wempe, T. G. and Lunen, M. v. (1991). The IBM Speechviewer. In *Proceedings of the Institute of Phonetic Sciences Amsterdam 15*, pages 121–130. IFA publications.

A

List of Abbreviations

Following section includes list of abbreviations used in the thesis.

ASR: Automatic Speech Recognition

CNN: Convolutional Neural Network

DTLS: Datagram Transport Layer Security

EPG: Electropalatography

GDPR: General Data Protection Regulation

GMM: Gausian Mixture Model

HMM: Hidden Markov Model

LPC: Linear Predictive Coding

LSTM: Long Short-Term Memory

MFCC: Mel Frequency Cepstral Coefficient

OLP: Ortho-Logo-Paedia

RNN: Recurrent Neural Network

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

VST: Visual Speech Therapy

B

Problem Description

Following section includes copy of problem description given for this thesis.

Faculty of Engineering Science and Technology
Department of Computer Science and Computational Engineering
UiT - The Arctic University of Norway

Software tools for speech therapists

Daniel Aaron Salwerowicz

Thesis for Master of Science in Computer Science

Problem description

The SpeechViewer suite, developed by IBM during the 1980s and 1990s, constitutes one of the reference software tool packages utilized by speech therapists and for training of the hearing impaired. The availability of up-to-date software tools possessing similar features and capabilities seems to be limited. New tools exploiting modern technologies and platforms, such as, for example, web-based solutions or apps for tablets or smartphones, could be of great interest to speech therapists. This master thesis suggests to determine desired methods and features, provide an overview of possible alternative solutions, and to compare and analyze the differences.

Objective:
The problem is to present some representative examples of software features which would be beneficial for speech therapists, based on what was available in the past, compare it to what is available now, and the way speech therapists work today. Then propose a design and implementation of a prototype capable of providing some selected features.

Structure the problem.
- What kind of typical / difficult sounds are hard to pronounce?
- How can we recognize these issues?
- How much, and which, information do we need?

Recognition part.
- Obtain an overview of possible technologies which could be considered, e.g.
    - Available software libraries
        - Audio / acoustic processing
        - Speech / voice recognition modules
    - Mathematical models
    - Speech aid for the hearing impaired
- Develop a proof of concept based on the chosen method.

Verification part.
- Consider a few alternative approaches, e.g.
    - AI based methods
    - Signal processing
    - ...
- Implement a prototype for analysis of one or more specific types of recognition
- Comparison and analysis
- Errors and tolerance
- Presentation of the results

Implementation.

The choice of programming language, third party libraries, operating system etc. may be restricted per agreement with the supervisors. The same applies for solution architecture and mathematical models.

Information, provided by the supervisors.
- Necessary descriptions, documentation, characteristics and methods.
- Experimental results.
- Assumptions.

Dates

Date of distributing the task:	<07.01.2019>
Date for submission (deadline):	<11.06.2019>

Contact information

Candidate	Daniel Aaron Salwerowicz dsa014@post.uit.no
Supervisor at UiT-IVT	Rune Dalmo rune.dalmo@uit.no
Supervisor at UiT-IVT	Tanita Fossli Brustad tanita.f.brustad@uit.no
External supervisor	Nina Opdahl nina.opdahl@gmail.com

General information

This master thesis should include:
- ❄ Preliminary work/literature study related to actual topic
  - A state-of-the-art investigation
  - An analysis of requirement specifications, definitions, design requirements, given standards or norms, guidelines and practical experience etc.
  - Description concerning limitations and size of the task/project
  - Estimated time schedule for the project/ thesis
- ❄ Selection & investigation of actual materials
- ❄ Development (creating a model or model concept)
- ❄ Experimental work (planned in the preliminary work/literature study part)
- ❄ Suggestion for future work/development

Preliminary work/literature study

After the task description has been distributed to the candidate a preliminary study should be completed within 3 weeks. It should include

bullet points 1 and 2 in “The work shall include”, and a plan of the progress. The preliminary study may be submitted as a separate report or “natural” incorporated in the main thesis report. A plan of progress and a deviation report (gap report) can be added as an appendix to the thesis.

In any case the preliminary study report/part must be accepted by the supervisor before the student can continue with the rest of the master thesis. In the evaluation of this thesis, emphasis will be placed on the thorough documentation of the work performed.

Reporting requirements

The thesis should be submitted as a research report and could include the following parts; Abstract, Introduction, Material & Methods, Results & Discussion, Conclusions, Acknowledgements, Bibliography, References and Appendices. Choices should be well documented with evidence, references, or logical arguments.

The candidate should in this thesis strive to make the report survey-able, testable, accessible, well written, and documented.

Materials which are developed during the project (thesis) such as software / source code or physical equipment are considered to be a part of this paper (thesis). Documentation for correct use of such information should be added, as far as possible, to this paper (thesis).

The text for this task should be added as an appendix to the report (thesis).

General project requirements

If the tasks or the problems are performed in close cooperation with an external company, the candidate should follow the guidelines or other directives given by the management of the company.

The candidate does not have the authority to enter or access external companies’ information system, production equipment or likewise. If such should be necessary for solving the task in a satisfactory way a detailed permission should be given by the management in the company before any action are made.

Any travel cost, printing and phone cost must be covered by the candidate themselves, if and only if, this is not covered by an agreement between the candidate and the management in the enterprises.

If the candidate enters some unexpected problems or challenges during the work with the tasks and these will cause changes to the work plan, it should be addressed to the supervisor at the UiT or the person which is responsible, without any delay in time.

Submission requirements

This thesis should result in a final report with an electronic copy of the report including appendices and necessary software, source code, simulations and calculations. The final report with its appendices will be the basis for the evaluation and grading of the thesis. The report with all materials should be delivered according to the current faculty regulation.

If there is an external company that needs a copy of the thesis, the candidate must arrange this. A standard front page, which can be found on the UiT internet site, should be used. Otherwise, refer to the “General guidelines for thesis” and the subject description for master thesis.

The supervisor(s) should receive a copy of the the thesis prior to submission of the final report. The final report with its appendices should be submitted no later than the decided final date.

C

Initial Survey

Following section includes initial survey into development of a modern VST product. Blank pages added by the formatting have been omitted to save on paper.

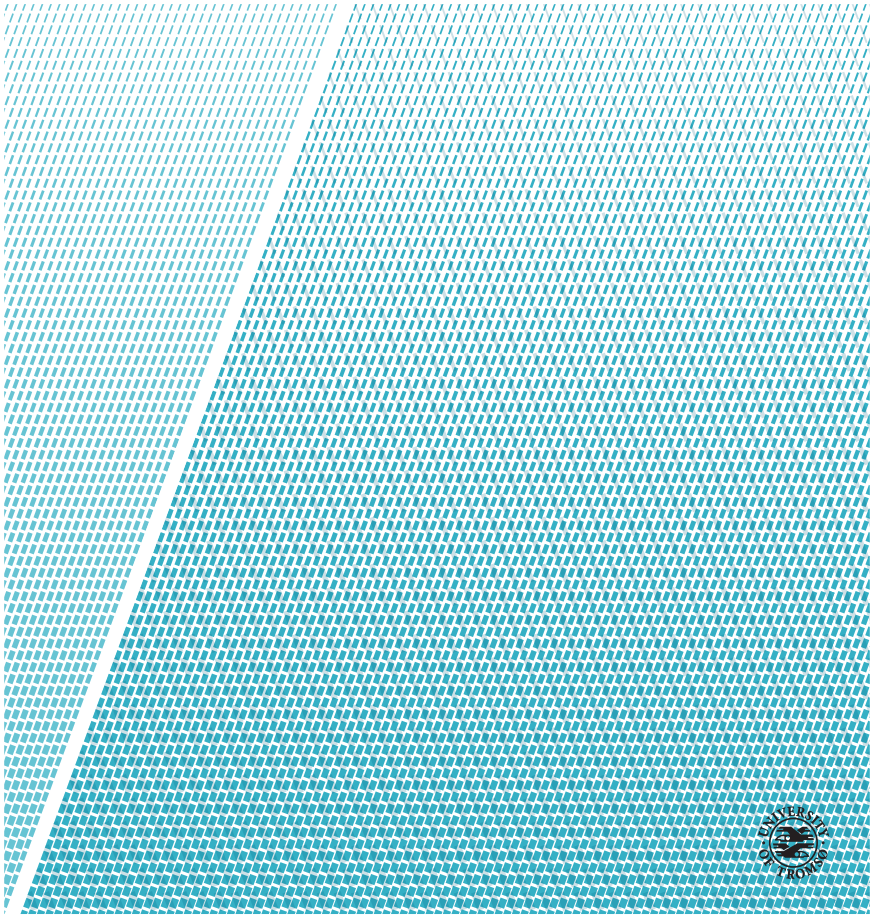
UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

Faculty of Engineering Science and Technology
Department of Computer Science and Computational Engineering

Initial Survey for a Software Tool for Speech Therapists

—

Daniel Aaron Salwerowicz
Pre study for master thesis in Applied Computer Science, February 2019



Abstract

Purpose of this document is to explore and define necessary features for the updated Virtual Speech Therapy tool. It functions as an analysis of findings regarding both the IBM called Speech Viewer III, and other similar tools.

Contents

Abstract i

List of Figures v

1 Introduction 1

2 Intended features and functionality 3

3 Existing programs 7
3.1 IBM Speech Viewer III 7
3.2 Phone and Tablet apps 8
3.3 Electropalatography 10

4 Conclusion 11

Bibliography 13

List of Figures

2.1 Example of exercise in IBM Speech Viewer III 4

3.1 Screenshot from Advanced Reading Therapy app 8
3.2 Screenshot from Articulation Station app 9
3.3 Screenshot from Speech Tutor app 9
3.4 Diagram of Electropalatography system 10

1 Introduction

Visual Speech Therapy (VST) tools are a type of computer programs used by therapist to aid them in treatment of speech impediments in their patients as well as learning deaf patients proper pronunciation. VST's make it possible for patients to get immediate and accurate feedback on their skills as well as track the progress they make during their treatment.

To the best of our knowledge there are no up to date VST programs available on the market, with IBM Speech Viewer III being released in 1995, that have the needed specifications (Destombes, 1993). Although there are various small programs available on the market for both computers and tablets, none of the ones that were tested were satisfactory. They often lacked various useful features found in the Speech Viewer program. In addition to lacking support for languages other than English.

After conversations with Nina Opdahl, a speech therapist at Narvik Municipality, the main features for a new VST application have been identified and this document will describe them.

/2

Intended features and functionality

There were many principles that IBM has based itself on when they created the Speech Viewer programs I-III. Goal of this project is to try and use them as well and develop some of the similar features that are more up to date, using modern technologies and solutions.

One of the main principles was constant feedback from users, both therapists and the patients. Based on conversations with Nina Opdahl and exploration with IBM Speech Viewer and similar programs following list of desired features was compiled.

1. Ease of use
2. Efficiency
3. Adaptability
4. Useful feedback
5. Adjustable input
6. Attractive visuals

7. Interesting and engaging exercises

Given how main focus group are children, the program needs to be easy in use. It's possible to see this in the way IBM Speech Viewer was organized. Recreating this ease of use is something that should be a main focus in the new product. Navigation in the program is easy and most often one mouse click is needed to start the exercise, shown in figure 2.1.

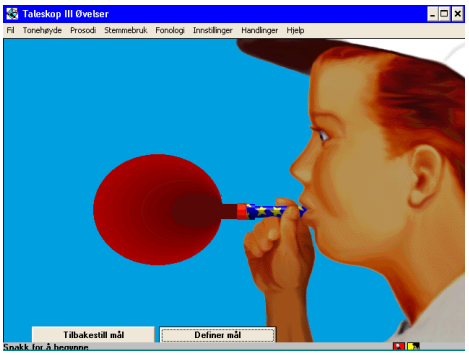


Figure 2.1: An example of exercise found in Speech Viewer, balloon size changes appropriately to the voice strength.

Adaptability, or lack thereof was one of the main complaints raised against Speech Viewer (Wempe and Lunen, 1991). This complaint is something that will be easier to solve these days given a better space for storing data as well as access to more advanced solutions. One of them being machine learning making it possible for the program to learn and adjust itself as necessary. A good example shown by Wempe and Lunen was lack of pitch correction and higher sampling range. This made it harder for males to use this program as usually their pitch is below 75Hz, which is the lowest limit for Speech Viewer.

Of course machine learning is not a cure all solution, and it comes with some drawbacks. One that are worth mentioning are learning set, which often needs to be quite large to give satisfactory results. Big datasets are easier to come by nowadays, but they are not perfect and can give erroneous results, if one is not careful with how they use it. Therefore machine learning is a wildcard when it comes to systems like that and it would require a lot of testing to give proper

results.

Making the program attractive, engaging, and interesting might prove itself to be the biggest challenge, but getting it right will help considerably in its effectiveness (Destombes, 1993). As children's attention span is quite short, therefore combining some game elements, progression and achievements might prove itself to be beneficial in keeping them engaged. Giving stars as reward for completed exercise might be rewarding enough for preschoolers and older children. However it is important that the program produces more constructive feedback for older users as well as the therapists.

In general the program needs to listen to patient's speech and give them feedback dependent on the type of exercise they do. This can be something as simple as speech strength, to more advanced topics like voiced and unvoiced sound detection, tone control, phoneme recognition, etc. This will involve technologies like signal processing, voice recognition, and possibly much more, for example machine learning and graphing.

Such visual tool will be quite helpful for the therapist, although they will not replace the professional help from therapist as noted by Lee in (Sue, 2018).

3

Existing programs

Using IBM Speech Viewer III as a general guideline a short survey has been carried out where several programs were tested. The features were in these programs were tested and studies for their accuracy, usefulness, and therapeutic capabilities based on Speech Viewer and knowledge from meetings with Nina Opdahl. This section will try to evaluate which features are viable to implement in the updates VST program, based on what others have done.

3.1 IBM Speech Viewer III

The most advanced tool to date providing speech therapy tools is the IBM Speech Viewer. It is however severely outdated and as such proves itself to be quite useless, as it needs to be run on Windows XP with extra patches. It also proves itself to be quite lacking in precision as it has quite low sampling rate and its voiced/unvoiced sound recognition is bad, (Wempe and Lunen, 1991). Those problems were already noted in 1991, and almost 30 years of technological development have added new expectations and made this program almost completely obsolete and archaic. Therefore an up to date alternative is necessary to help children learn proper pronunciation and speech patterns.

Gameplay elements found in it are also outdated and would require updating in order to be relevant for today's youth.

3.2 Phone and Tablet apps

The most advanced application that was found during the research and tested was the Advanced Reading Therapy app (Tactus Therapy Solutions, 2014a). It gives its user a text to read as well as reading it to them so they can hear the proper pronunciation. Sadly this type of application is far from perfect, as it gives no feedback to the user as they read the text.

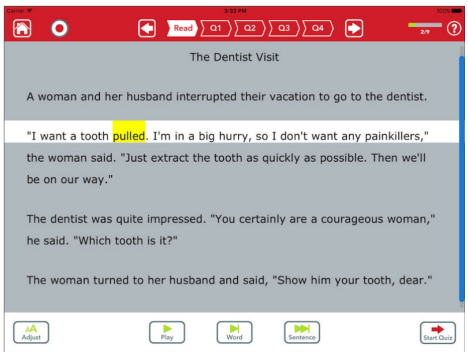


Figure 3.1: Screenshot from Advanced Reading Therapy app showing one of its exercises, source (Tactus Therapy Solutions, 2014b).

It is also not appropriate for young users who cannot read yet, and they would benefit the most from such type of tools. There are more child friendly application on the market, such as Articulation Station that offers a more child friendly interface (Little Bee Speech, 2019a). This application uses images and recorded words to help children know what they are supposed to say and how to pronounce it. However they require feedback from the therapist to validate their pronunciation.

Figure 3.2: Screenshot from Articulation Station app showing one of its exercises, source (Little Bee Speech, 2019b).

Another useful tool found was the Speech Tutor app, (Synapse Apps, 2008a). It shows user proper placement and movement of the tongue required to pronounce a given sound.

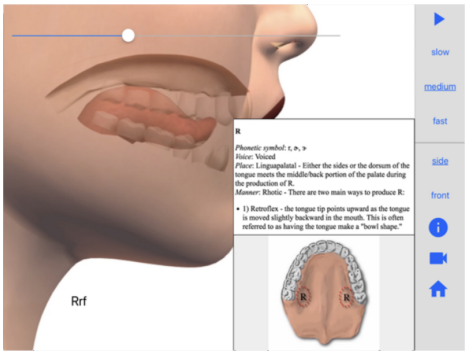


Figure 3.3: Screenshot from Speech Tutor app with one of its animations showing proper tongue placement for a given sound, source (Synapse Apps, 2008b).

3.3 Electropalatography

Although beyond the scope of this project it is worth mentioning another technology used in speech therapy called Electropalatography (EPG (Rose Medical Solutions, 2019b)). It uses an artificial palate with 62 electrodes to determine the placement of tongue and its contact with palate during speech. Such a tool would be a good addition to the program developed during this project, as it would give extra information to therapists. It would give immediate response to users, showing them their tongue placement, and then the correct placement of the tongue.

Figure 3.4: Diagram for an Electropalatography system in use, source (Rose Medical Solutions, 2019a).

4

Conclusion

Even though there have been many various tools developed and used by speech therapists, from cathode tubes in 1946 (Destombes, 1993), through Speech Viewer, and modern apps, none of them are up to date or offer the necessary functionality needed to perform effective therapy. While Speech Viewer is packed full of useful features, they are either outdated or hard to use on modern machines. And new programs lack several vital features, most notably some form of immediate and objective feedback, in addition to being English only tools.

Developing a tool capable of combining those features and presenting them in an easy to use and engaging manner is something worth pursuing, as it would give therapists and their patients a vital tool in therapy. It won't of course replace a human interaction and tutoring, but it will be a handy addition to employ in helping people with speech impediments.

The most promising technology worth focusing on would be the ASR (Automatic Speech Recognition) technology, modified to be used in speech therapy. As described by Kitzing et al in (Kitzing et al., 2009) even the smallest differences in utterance of sounds and words can be easily misunderstood by computers. Developing an individual profile for each user would benefit greatly in the systems ability to parse what they are saying and help them speak in correct manner. One of the backbone for such ASR systems are HMMs (Hidden Markov Models) which allow it to learn automatically. Using GMMs (Gausian Mixture Models) in tandem with HMMs allows the program to guess with high precision

what phones user has articulated. This way the program could distinguish between similar sounds like "c" and "s". It also comes with the added benefit of being able to calculate the *word error rate*, so that the quality of the system can be checked at all times.

Bibliography

Destombes, F. (1993). *The Development and Application of the IBM Speech Viewer*, pages 187–196. Springer-Verlag Berlin Heidelberg.

Kitzing, P., Maier, A., and Lyberg Åhlander, V. (2009). Automatic Speech Recognition (ASR) And Its Use as a Tool for Assessment or Therapy of Voice, Speech, and Language Disorders. *Logopedics, phoniatrics, vocology*, 34:91–6.

Little Bee Speech (2019a). Articulation Station Pro. http://littlebeespeech.com/articulation_station_pro.php. Accessed on 2019-02-15.

Little Bee Speech (2019b). Articulation Station Screenshot. http://littlebeespeech.com/images/ac_level_1_story_full.jpg. Accessed on 2019-02-15.

Rose Medical Solutions (2019a). Electropalatography Diagram. http://www.rose-medical.com/images/how-electropalatography-works.jpg. Accessed on 2019-02-15.

Rose Medical Solutions (2019b). Electropalatography EPG System. http://www.rose-medical.com/electropalatography.html. Accessed on 2019-02-15.

Sue, A. L. (2018). This Review of Virtual Speech Therapists for Speech Disorders Suffers from Limited Data and Methodological Issues. *Evidence-Based Communication Assessment and Intervention*, pages 1–6.

Synapse Apps (2008a). Speech Tutor Pro. https://www.speechtutor.org/app. Accessed on 2019-02-15.

Synapse Apps (2008b). Speech Tutor Screenshot. https://bit.ly/2WlIImb. Accessed on 2019-02-15.

Tactus Therapy Solutions (2014a). Advanced Reading Therapy. https://tactustherapy.com/. Accessed on 2019-02-15.

Tactus Therapy Solutions (2014b). Advanced Reading Therapy Screenshot. https://bit.ly/2Jz6epH. Accessed on 2019-02-15.

Wempe, T. G. and Lunen, M. v. (1991). The IBM Speechviewer. In *Proceedings of the Institute of Phonetic Sciences Amsterdam 15*, pages 121–130. IFA publications.