Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

Sujith Chandra Padaru Shrikantha

Synthetically Generated Speech for Training a Pronunciation Evaluation System

Master's Thesis
Espoo, April 17, 2010

Supervisor: Professor Mikko Kurimo, Aalto University
Advisor: Reima Karhila

Aalto University
School of Science

Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

ABSTRACT OF MASTER'S THESIS

Author:	Sujith Chandra Padaru Shrikantha		
Title:	Synthetically Generated Speech for Training a Pronunciation Evaluation System		
Date:	April 17, 2010	**Pages:**	51
Major:	Machine Learning, Data Science and Artificial Intelligence	**Code:**	SCI3044
Supervisor:	Professor Mikko Kurimo		
Advisor:	Reima Karhila		

Computer-Aided Pronunciation Training (CAPT) Systems are designed to help users acquire speaking skills in a non-native language (L2 Language). Generally, CAPT systems employ speech recognition techniques to give a wellness score for an utterance. The score helps the learner evaluate themselves and gives the support to improve their pronunciation. Scoring from such systems correlate well with human-annotated scores when the uttered sequences are long and the speakers are adult. However, in the Say It Again Kid (SIAK) project, a CAPT game built for children, utterances are short, and consequently the correlation between scores of the system and human annotator is weak. The unavailability of children's speech data for training is the main reason for the poor performance. The thesis shows how to mitigate the problem of the unavailability of transcribed data by generating them using a modern text-to-speech (TTS) system. Such systems have shown to reach a human level of naturalness. In this work, a TTS system is trained to generate Finnish speech in children's accents. The system utilizes a large quantity of adult speech and a small set of children's speech to generate speech with children's accents. Finnish accented English is generated from the same system by mapping English words to their nearest Finnish phonetic representation and inputting them into the TTS system. Thus, the thesis proposes a simple way of achieving accented speech.

We add the generated data to the training of the phonetic recognition model employed in SIAK. The thesis shows that this technique improves the recognition accuracy of the model: the Phoneme Error Rate (PER) reduced from 0.27 to 0.13 for the Finnish children's test set. Unfortunately, this improvement in recognition results does not imply an improvement in the SIAK scoring. This was due to a mismatch between the data used for training and testing the recognition system and the target game words: even though the generated speech resembles the target game words, they belong to different distributions.

Keywords:	Pronunciation training, text-to-speech, synthetic, kids, phoneme, voice, game, score
Language:	English

Acknowledgements

I wish to thank Professor Mikko Kurimo and advisor Reima Karhilla for their support, guidance, and patience throughout the project. I also wish to thank my family and friends who supported me during this time. Thanks to Aalto University for the opportunity and the resources which enabled this thesis.

Espoo, April 17, 2010

Sujith Chandra Padaru Shrikantha

Abbreviations and Acronyms

IPA	International Phonetic Alphabet
SIAK	Say It Again Kid
ASR	Automatic Speech Recognition
HMM-GMM	Hidden Markov Model-Gaussian Mixture Model
DNN	Deep Neural Networks
RNN	Recurrent Neural Networks
TTS	Text-to-Speech
MOS	Mean Opinion Score
GRU	Gated Recurrent Unit
MFCC	Mel-Frequency Cepstral Coefficients
RELU	Rectified Linear Unit
CTC	Connectionist Temporal Classification
MOS	Mean Opinion Score

Contents

Chapter 1

Introduction

The goal of this work is to improve the pronunciation evaluation system of the Say it Again Kid(SIAK) game, which helps Finnish children learn to speak English.

1.1 L2 language learning

A language is a tool for communication. We express ourselves through language and speech. As humans evolved, the need for sophisticated forms of communication led to the development of a large number of languages, with roughly 6500 spoken languages spoken in the world today [1]. Studies have found that learning multiple languages is essential for many reasons. First, knowing multiple languages enables us to communicate with more people and better integrate into different cultures. Learning multiple languages has also been shown to help our brains by improving planning and problem-solving skills. Furthermore, in the business world, knowing more than one language is a necessity to be able to build better global relationships [2].

In terms of language acquisition we all know that newborns learn their mother tongue (L1 language) naturally. According to neuroscience, an essential task for infants in language acquisition is to make sense of the sounds they hear in speech; i.e., infants learn to differentiate between sounds and categorize them [3]. These categories are similar to the linguistic definition of phones, which are the basic unit of sound in speech. Infants acquire the language-specific phonetic rules by learning the distribution of sounds in speech. This language acquisition ability, particularly in terms of phonic identification, has been found to deteriorate with age. This has led to the question of whether adults are incapable of acquiring a non-native language

(L2 language). However, this is not the case; it simply becomes a little harder than it is for infants [4].

There is no singular, definitive method for language learning. Linguists and language teachers use different methods with the main source of agreement being that it involves acquiring the pronunciation and vocabulary through repeated exposure. To be able to understand a new language, it is imperative to understand the words in it, and how it will be used. i.e., the vocabulary and grammar of the language. To be able to speak a new language, a learner must listen and grasp the statistical distribution of phones in the language and be able to identify and produce new phones which were absent in his or her native language.

Not everyone has the time, resources, or the necessity to opt for a human coach to acquire an L2 language. Naturally, a variety of tools have been developed to assist the language learning process including books that teach vocabulary, games with audio-visual features, digital flash-cards, movies, songs, and audio. With the maturing of computer systems, mobile phones, and Automatic Speech Recognition(ASR), a lot of computer-aided language learning tools have emerged, such as; Duolingo, Babel and Busuu. These provide learning content, exercises and automated evaluation, which gives the learner real-time feedback.

1.2 Say It Again Kid

Say It Again Kid(SIAK) is a computer-based game, which aims to assist Finnish children to learn better English pronunciation skills. SIAK was developed in the department of signal processing and acoustics, Aalto University, in collaboration with Helsinki University. The project was developed with the following goals:

- Finnish children learn English in classrooms, but it is often impossible to assist every child with pronunciation as it takes so much time. SIAK is designed to alleviate this problem.
- To validate the hypothesis that using an engaging gaming environment increases the speed of learning.
- To determine if automated feedback on the pronunciation quality benefits the learning process.

1.2.1 Game description

Figure 1.1: A screenshot from Say It Again Kid

The game is aimed at Finnish children with little or no experience in English. The game has 27 levels, and its vocabulary consists of frequently used English words. In each level, the player must complete a path that is obstructed by rotating card icons, as seen in 1.1. When the player clicks on the card they will see a picture of an object and hear the associated word first in Finnish, then in English. The player should try to repeat the word in English while trying to imitate the model pronunciation. After attempting, the player will then hear the model pronunciation and their own utterance, followed by an accuracy based score between 1 and 5. If the score is above 1, the path opens and leads to the next card in the path. If the score is below 1, the player is asked to try again. The game is not currently available for public consumption and is used for research and development only. The scoring is calculated using speech technologies. If the scoring is accurate, then the scores can be used to track the learning process of the children and modify the sequence of presented words based on their current level. The game does not require any reading skills, and the L2 language phonemes are learned by repeating and feedback, which is similar to how children learn their mother tongue.

1.2.2 SIAK game data

A corpus of 20,000 game utterances was collected and scored on a scale of 0-100 by a single teacher, in order to develop the scoring model. The children included 24 UK English native speakers and 153 Finnish children between 8 and 12 years of age. The details of this dataset can be found in chapter 4. Different scoring models have been built for the game with an aim to model and follow human scoring.

1.3 Scoring System for SIAK

There are primarily two approaches to build a scoring system for pronunciation using Automatic Speech Recognition (ASR) techniques. The first approach uses output of the ASR model for the game words to calculate a score for the utterance [5], [6]. A Hidden Markov Model-Gaussian Mixture Model(HMM-GMM) or Deep Neural Network(DNN) acoustic model is trained on the target language of the game, or a combination of the native language and target language. Based on acoustic model predictions for the utterance, a goodness score is calculated either from data or using handcrafted rules. For this to work the speaking style of the players should match the training data used to train the ASR model.

The second approach is to create forced alignment between the utterance and its phonetic representation using Hidden Markov Models and analyzing these individual audio segments for the corresponding phoneme using a discriminative DNN model[7], [8]. The accuracy of the scoring system is determined by the accuracy of the phonetic segmenter and the DNN model.

Previously, models using both of these approaches have been developed for the SIAK game [9], [10]. Both of these systems achieve a correlation of around 0.5 between the predicted score and the score given by the teacher. A much higher correlation is desirable, but the data inherently poses some challenges in achieving this which are discussed in the next section.

1.3.1 The challenges

When the utterances are long, i.e., above 3 minutes, pronunciation scoring systems have been built, which correlate very well with the human scores[11]. The correlation is as high as 0.85 for a few datasets, but this degrades when

short speech segments have to be evaluated. The various challenges with scoring SIAK game words are as follows.

- There is no in-domain data for training the model.
- The target data is speech from children, whereas the training data is mostly speech by adults.
- The target utterances are short, and the training data contains predominantly long utterances. It is possible to segment the long utterances into shorter ones, but this will inherently add some error to the training setup.
- The target utterance might contain phones which come from both English and Finnish. There is no such transcribed data available for training.
- Training data only contains the correct pronunciation for every word in either language. The model might over learn the phonetic distribution of either language and be biased against predicting a mixed phonetic representation for an utterance.
- Since the utterances are short, it is not easy for the teacher to score them well. The human bias in the SIAK game data might make the challenge more difficult, resulting in a worse correlation.

1.3.2 The approach

It seems that the challenge in building a good scoring model for SIAK game data is that there is no in-domain data for training. There are sizeable audio corpora for English and Finnish, but the portion of child speakers within them is significantly small. Moreover, there is no L2 English transcribed data spoken by Finnish children; the pronunciation scoring model has to be built on L1 data while being expected to capture the mispronunciations of L2 data.

Significant leaps in Text to Speech Synthesis(TTS) technology have been achieved in recent years with end-to-end sequence-to-sequence encoding techniques [12]. The state-of-the-art systems have been able to obtain speech naturalness close to human speech [13]. These sequence to sequence architectures allow conditioning for speaker, prosody, and other characteristics, enabling controllable human-like speech synthesis [14]. These results have led to the further exploration of:

- If in-domain data for the SIAK game can be generated using TTS synthesis, i.e., synthesizing Finnish accented English data in children's voices. The aim is to generate mispronounced utterances which are not found in the L1 English.
- Will using the synthesized data for training the CAPT models improve the correlation between human expert score and the CAPT system score?

Chapters 2 and 3 describe how such a system can be built, while Chapter 4 details the experiments that were conducted to evaluate the effect of synthetic speech on the scoring model.

Chapter 2

Phonemes and Phonology

All languages have an inherent structure which is formulated as the rules of that language. The rules governing the structure of the language are its grammar, while the collection of all the words in a language is its lexicon. Words in the lexicon are combined using grammatical rules to form sentences. Any number of concepts can be communicated with the same framework, and native speakers need not think about these rules while communicating through language, as They are internalized. Notably, the structure and the vocabulary of a language are not fixed; rather, these are subject to constant change, and languages are constantly evolving[15]. In this chapter, the basics of linguistics and phonology are discussed as these form the basis for Finnish accented English speech generation.

2.1 Phonemes

While grammar and lexicon model the composition of a language, phonemes and phonology model the speech. "Phonemes are defined as the smallest unit of sound that may cause a change of meaning of an utterance within a language" [16]. By themselves, phonemes do not have any meaning. However, for every word in a language, a phonetic representation is defined depending on the sounds in its pronunciation. For example, consider the word "tear" which can either mean "to pull something apart" or "a drop of liquid from the eye". The difference in meaning arises from how it is pronounced; if "tear" is pronounced /t ɛː/ which is the phonetic representation for the word, then it means "to pull something apart" and if it is pronounced /t ɪ ɘ/ then it refers to "a drop of liquid from the eye." Here; *t, ɛː, ɪ, ɘ* represent the individual phonemes corresponding to the individual unique sounds, which create a

change in meaning English.

An important point to observe is that although a standard phonetic representation for every word can be defined, all the different ways people vocalize a single word cannot be uniquely characterized by a single set of symbols. So often, phonemes are viewed as an abstract underlying the representation of pronunciation, and the speech signal, as a realization of that representation. In other words, phonemes represent a sound or set of sounds that are perceived similarly by people of one particular L1 language[17]. The sounds that are perceived to be different in a language can be perceived as the same in another language. For example, /k/ and /k^h/is perceived to be the same phoneme in English while it is two separate phonemes in French as it can change the meaning of a word.

This impreciseness gives rise to the possibility of transcribing the same language with a different set of phonemes. ASR systems which require precise representation for the utterance, usually use triphone representation for extra precision. A triphone representation assumes that the realization of a phone is dependent on its adjacent phones. Consider the word "that"; a regular phonetic representation of it would be /ɵ a t/. An ASR system which uses triphone representation would transcribe it as / {- ð a} {ð a t} {a t -} /. So there is no single correct phonemic representation for the language, but it is necessary to choose a system that is useful for the application at hand[18].

2.2 International Phonetic Alphabet

International Phonetic Association was formed in Paris in 1886 to develop and use phonetic notations in schools to help children learn foreign languages. In the early 19^{th} century, the association grew, with language teachers from Western Europe joining the association, and the International Phonetic Alphabet (IPA) was created. The alphabet aims to provide an internationally agreed set of symbols for the sounds of languages.[16]. IPA has become a standard in phonetic representation and is used by phoneticians world-wide. In IPA, phones are represented either as single letters or as a combination of letters and diacritics, which are signs around a letter that give additional information about the exact pronunciation. The alphabets in IPA are organized as a chart, as described in Figure 2.1.

At the top of the table, 59 pulmonic consonants are represented, which are produced by obstructing the mouth or vocal cords and subsequently letting out the air from the lungs. The different consonants are further categorized

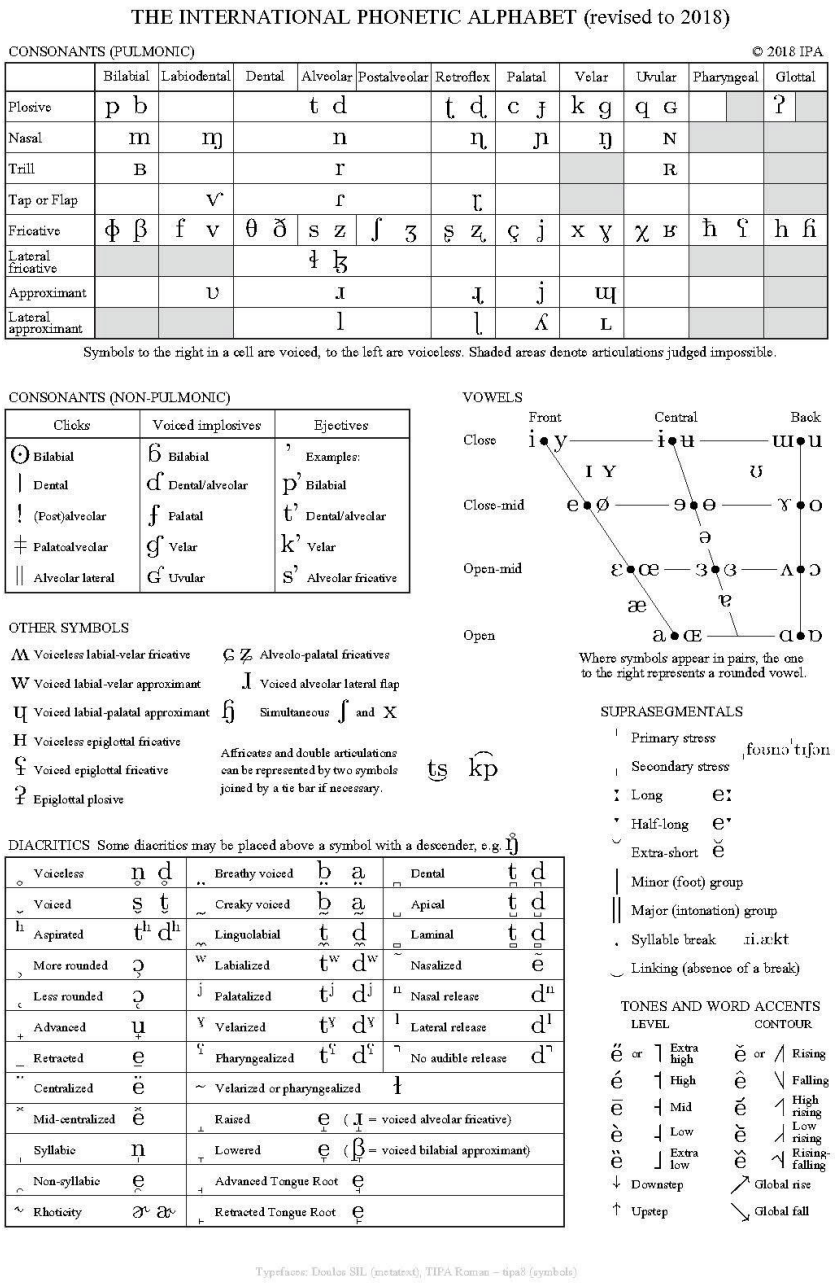

Figure 2.1: The International Phonetic Alphabet

into rows, based on how exactly they are produced (manner of articulation) and into columns based on where they are produced within the vocal tract.

These consonants make up most of the sounds that humans produce to communicate through languages.[19]

The position of the tongue while producing a vowel is used to place the different vowels in the IPA Chart. Vowels vertical position is determined by vowel height, i.e., The height of the tongue in the mouth while pronouncing the vowel. The horizontal position of the vowel is determined by vowel backness, i.e., how back or front the tongue is placed in the mouth while pronouncing the vowel.

Non-pulmonic consonants are those whose production does not depend on the airflow of the lungs. Clicks, implosives, and ejectives make up this set of symbols. Although IPA has 173 symbols, only a small subset of these is used to represent any language.

2.3 Phonesets of English and Finnish.

Each language has its own speaking style and rules which govern the possible distribution of phonemes. The speaking style of a language is characterized by,

- The phoneme set that is used in the language
- The distribution of phonemes in the language.
- The manner in which the different phonemes are stressed.
- The rhythm and intonation that is used in sentences.

These details are difficult to grasp for a language learner if these patterns are missing in their native language and it is not straightforward to list how exactly two languages sound different. The usual pitfalls of a new language learner can be listed; for example, the following observations have been made about the Finnish L1 speakers learning to speak English.

- Finnish learners have difficulty pronouncing English words that contain front vowels and back vowels in the same word. This is because the Finnish language follows vowel harmony, where front vowels and back vowels never appear together in a word.
- Finnish learners sometimes confuse long vowels for short and vice-versa; for example, ”sip” and ”seep.”

- Among consonants, /θ/ is often mispronounced as /s/ or /f/; for example "thick" and "three."
- English words that cluster more than two consonants together are often difficult to pronounce for Finnish speakers, as such patterns are absent in Finnish; for example, "street," "stretch" and "Christ."
- The stress pattern in English words and sentences is difficult to grasp for Finnish speakers. In Finnish, the first phoneme of the word is usually stressed, which is not the case in English.
- It is difficult for Finnish speakers to produce sentences in a rising tone, which is a requirement in English.
- In Finnish, /p/, /t/, /k/ are never aspirated, which is not the case in English.
- In Finnish, /t/ has a dental place of articulation, meaning it is produced with the tongue against the upper teeth. Also /r/ is usually trilled in Finnish. So, related phonemes in English can be difficult to pronounce for a Finnish learner.

2.4 Phonological features

For a long time, phonemes were considered as the basic units of speech representation. In the 1980s, however, linguists realized that individual phonemes could be seen as a surface form realization of the combination of fundamental units, which are termed as phonetic features[20][21]. Taking an analogy from physics, Phonetic features can be seen as the subatomic particles that make up the atom, which is a phoneme. Following this abstraction, linguists could express phonemes through a smaller set of features. A phoneme is represented by either presence or absence of phonetic features, which can be better understood with an example. The phonological features for three phones which represent the word "pin" are described in the table 2.1.

	p	**ɪ**	**n**
syllabic	0	1	0
sonorant	0	1	1
continuant	0	1	0
high	0	0	0
back	0	0	0
voiced	0	1	1
.	.	.	.
.	.	.	.
.	.	.	.

Table 2.1: Phonetic Feature representation for the phonemes p,ɪ,n

The vector representation of the phoneme enables analytical manipulations, which are used in this thesis. Again, it should be noted that the choice of phonetic features to represent a phoneme is not unique and is dependent on the application[20]. The phonetic features that are used in this work are listed in tables 2.2 and 2.3.

Phonetic Features for Vowels
dipthong, long, rhotic, unround-schwa front
nearfront, central, nearback, back, open
nearopen, openmid, mid, closemid, nearclose, close, rounded, unrounded
diphthong-forward, diphthong-backward
diphthong-opening, diphthong-closing, diphthong-rounding
diphthong-unrounding

Table 2.2: Phonetic Feature for Non-Vowels

Phonetic Features for Non-Vowels
affricate, approximant, fricative, plosive, nasal, trill, alveolar, bilabial
coronal, dental, dorsal, labial, labiodental
lateral, postalveolar, velar pulmonic, retroflexed
syllabic, palatalized, aspirated, lenis, fortis
labialized, voiced, unvoiced, geminated

Table 2.3: Phonetic Feature for Non-Vowels

2.5 Distance measure between phones

The phonetic feature vector representation for each phone allows the definition of a distance metric over the phone space. The distance, d_{ij} between two phones p_i and p_j is given by,

$$d_{ij} = \frac{\text{Number of dissimilar features between } p_i,p_j}{\text{Total number of features}} \tag{2.1}$$

The above equation provides an approximate measurement of the distance between two phones. It is approximate, since not every feature will affect the acoustic realization of the phone in the same way. Once a distance measure between two phones is defined, it is straight-forward to define a distance measure between phone sequences. Levenshtein Distance can be used as a metric to measure this.

2.6 Finnish to English phonetic mapping

As discussed in the introduction, a large quantities of Finnish accented English needs to be generated, utilizing a TTS system. It is achieved by mapping English phonemes to the nearest Finnish phonemes and inputting them to a Finnish TTS system. Ideally, English phonemes should be mapped to nearest Finnish phonemes in an automated way, as hand-coding will require much effort. The method of automatically generating Finnish phonetic representation for English words is described below.

- An English phonetic representation is obtained for the English words by looking up a dictionary.
- A shared phone space is defined for English and Finnish, and the phoneme distance map between each phone is calculated using the equation in 2.1.
- For each phoneme in the English transcript, the three nearest phonemes in the Finnish phonetic system are obtained by looking up the phoneme distance map. Only three nearest phones are sufficient as, for every English word with $'n'$ phonemes, we get $'3^n'$ nearest representations in Finnish considering all the combinations.
- Twenty Finnish transcripts are chosen at random from these $'n^{3'}$ combinations. The choice of twenty is arbitrary; however it represents a

sufficiently large number from which to create a large corpus. It is assumed that the 20 random choices will incorporate the common errors made by the learner: since handpicking the words based on observed error pattern is time consuming.

The Table 2.4 below lists the five nearest Finnish phoneme representations for a SIAK English game word. The transcripts are generated automatically by a script.

Game Word	**English Rep**	**Nearest Finnish Rep**	**5 Finnish Representations**
girl	g ɜː ɬ	g i l	'g i l', 'g i lː', 'g i n', 'g ɑ l', 'g æ l'
hello	h ɛ l əʊ	h i l y	'h i l y', 'h i l ø', 'h æ l øy', 'f eː lː ø'
book	b ʊ k	b y k	'b y kː', 'b y p', 'b ø k', 'm y k', m o p
learn	l ɜː n	l i n	'l i n', 'l i nː', 'l i rː', 'l ɑ n', 'l ɑ nː'
tree	t ɹ iː	t rː iː	't rː iː', 't rː i', 't rː æː', 't l iː', 't l i', 't l æː', 't n iː', 't n i', 't n æː'
more	m ɔː	m o	'm o', 'm u', 'm uː', 'b o', 'b u'

Table 2.4: Finnish Phonetic Representation for Game Words

Finnish accented English is generated by inputting these transcripts into a Finnish TTS system. The hypothesis here is that the output will sound similar to Finnish accented English. It can be seen that these transcripts indicate subtle changes in the pronunciations. The aim is to determine if the TTS system can reproduce these minute variations without errors. The next chapter describes the text-to-speech system that is used to generate speech from the above transcripts.

Chapter 3

Speech Synthesis

3.1 Background

3.1.1 Types of text to speech system

Text-to-speech synthesis is a sequence to sequence mapping problem in which a shorter input sequence of text has to be mapped into a much longer waveform sequence whose output waveform has to be intelligible and natural. Also, a single input representation can have multiple output representations based on speaker, dialect, accent, prosody, and other vocal characteristics. This challenge has interested inventors and researchers for a long time. Mechanical and electrical systems were built to produce speech before the era of computers; historically, text to speech systems found use cases in assistive technologies with applications such as giving voices to people with deformed vocal tracts, or the production of audio-books for the blind. In recent times with the improvement of technology, more avenues have opened up for its use; voice assistants like Amazon Alexa, Google Home, and Siri are enabling people to interact with machines with their voices. The four prominent methods of speech synthesis which have found commercial use cases over the past few decades are listed below.

- Corpus-based concatenative speech synthesis.
- Rule-Based F0 formant synthesis.
- Statistical parametric speech synthesis.
- End-to-end neural network speech synthesis.

The methods are described briefly below.

In concatenative speech synthesis, speech is generated by joining together small segments of recorded speech. An extensive database of audio segments corresponding to phones, diphones, words, or other defined units, is created from a corpus using ASR hard-alignment techniques. These segments are then indexed with acoustic metadata. Speech is synthesized by selecting the best audio segments that correspond with the text by using weighted decision trees and joining the segments together. Digital smoothing filters are applied to the joined frames to reduce the artifacts of joining. The advantage of this method of synthesis is that the resulting audio sounds natural. The drawbacks of the system are, a need for an extensive database, and not having an option to condition the audio on different speakers, accents, and styles[22].

Rule-based F0 synthesis is developed based on the source-filter model of speech synthesis. As the name indicates in the source-filter model, speech is modeled to be produced by applying a filter(vocal tract) to sound source(vocal cords). The distinct characteristics of the source, i.e., the fundamental frequency(F0), and the filter can be identified for each phone. The source for all voiced phones is modeled as a periodic waveform, while for non-voiced phones, the source is modeled as noise. The shape of the vocal tract and the position of the tongue determine the transfer characteristics. For speech synthesis, these transfer characteristics are hand-coded for each phone. The design of these rules is complex and is done manually. Speech is reconstructed using the source model and the designed filter characteristics. Algorithms and techniques involved in the production of the audio waveform from vocal or filter parameters are referred to as Vocoder. [22].

In Statistical Parametric Speech Synthesis(SPSS), the probability distribution of the output waveform is learned from the data conditioned on the input text sequence. During training, the acoustic model is estimated automatically from a large speech corpus. At the time of synthesis, acoustic features are predicted from the learned acoustic model. The parameters modeled are usually spectral parameters such as Mel-Frequency Cepstral Coefficients (MFCC) and excitation parameters like fundamental frequency(F0). These parameters are modeled and conditioned on context-dependent input text sequence using a Hidden Markov Model. Speech is reconstructed from these features. This method has better flexibility compared to previous methods.

All the above techniques requires careful parsing and cleaning of text data to convert it into a detailed phonetic representation. This is usually referred to as text front-end and might involve many steps; text normalization, part of speech tagging, phonetic disambiguation, and phonetic clustering for context-dependent phonetic representation. Producing speech from phonetic representation is often referred to as speech back-end. For SPSS, the

back-end involves training the HMM-acoustic model, training the duration model for each context-dependent phone, synthesizing the acoustic parameters corresponding to text, fine-tuning the parameters, and synthesizing using Vocoder. All the different techniques mentioned above make text to speech synthesis a complex and laborious task requiring expert domain knowledge and hard-coding information. Errors at any stage of the process can accumulate through the pipeline, resulting in inferior synthesis.

3.1.2 Neural network based speech synthesis

In the past few years, the power of Deep Neural Networks(DNNs) in learning complex nonlinear features from massive labeled datasets has been understood [23]–[25]. Breakthrough results in visual object detection have propelled the use of DNNs for every complex discriminative machine learning problem. The ability to stack neural network layers, and form complicated architectures depending on the problem, make neural networks extremely versatile statistical models. Complicated neural network architectures with millions of parameters are made possible because of backpropagation; a simple way to learn those parameters using enough labeled examples.

DNNs have been used in various ways to tackle the problem of speech synthesis. DNN models have been used to build the different components of SPSS systems. Deep-Voice is a text to speech system in which every part of the SPSS pipeline is replaced with a Neural Network Alternative [26]. In addition, Wavenet was introduced by Google, which replaced the Acoustic Model and Vocoder of a speech synthesizer with an autoregressive Neural Network model [27]. These advances improved the quality of synthesized speech; still, the systems required the design of multiple moving parts. This challenge was finally solved with Tacotron which is a neural network architecture that maps the input text directly to the output waveform [12]. Compared with the models discussed earlier, Tacotron has the following advantages:

- The model can be trained on a $< text, audio >$ pair with no need for a text backend, and an acoustic model, reducing the number of components in the system.
- A sentence-level alignment between text and audio is sufficient to train the model, making it possible to train it on large datasets.
- A sequence to sequence framework makes it possible to condition the output waveform on different characteristics like speaker, prosody, and

language.

All the above advantages make Tacotron an ideal choice for the task of generating Finnish accented English, the goal being to condition the output waveform on speakers and accents. The model architecture of tacotron is an Encoder-Decoder model with Attention. Sequence-to-sequence Neural Network models are discussed in the next section, followed by the tacotron system description.

3.2 Seq-to-Seq Learning With Neural Networks

3.2.1 Recurrent neural networks

To train DNNs, inputs and outputs must be represented as vectors of fixed dimensions. When it comes to sequences, the input and outputs will be of different lengths, making it difficult to use the DNNs as they are. Recurrent neural networks tackle this issue by making use of recurrent connections in the hidden layer. A simple recurrent neural network can be described as follows: For a sequence of inputs $(x_1, x_2, ...x_{T)}$ and a sequence of outputs $(y_1, y_2, ...y_T)$, the RNN maps inputs to outputs as follows,

$$h_t = f(W^{ht}x_t + W^{hh}h_{t-1}) \tag{3.1}$$

$$y_t = f(W^{hy}h_t) \tag{3.2}$$

In the above equation, h_t represents the hidden layer at time step t, W^{ht} represent the weights given to each of the inputs, and W^{hh} represents the weights given to the previous hidden-state values. While learning this model, we backpropagate through time to learn the optimal model parameters. Through these recurrent hidden connections, the model learns the long term dependencies in the data. Practically though, it fails to learn long term dependencies due to the vanishing gradient problem. Long Short Term Memory(LSTM) networks and Gated recurrent networks(GRU's) are recurrent neural networks with better architectures to handle vanishing gradient problem. If the alignments between input and output sequences are already known, RNNs can be used to model the system, but when the relationship between the inputs and outputs is complex or not known, they cannot be used.

3.2.2 Encoder-Decoder architecture

Neural networks with Encoder-Decoder architecture tackle sequence to sequence problems with complicated input-output relationships. The architecture is as follows: the input sequence is mapped into a fixed vector using an RNN, which encapsulates all the information needed to predict the output. The mapped vector is then fed into another RNN, which will in turn predict the output one time step at a time based on the fixed vector and the previously predicted output sequence. Mathematically this can be described as follows. Given a sequential input $(x_1, x_2, ...x_{T^x})$, the encoder maps the sequence into a fixed length vector s as seen below:

$$h_t = f(x_t, h_{t-1}) \tag{3.3}$$

$$s = g(h_1, h_2...h_{T^x}) \tag{3.4}$$

Intermittent hidden state representations are learnt for each time step, with an LSTM, bidirectional LSTM, or GRU model. This is represented in equation 3.3. The fixed length vector is obtained as a function of all the hidden states. This can be the hidden state at time step T or more complicated depending on the problem at hand. The decoder outputs the sequence $y_1, y_2...y_{T^y}$, conditioned on s and the previous outputs.

$$p(y) = \prod_{t=1}^{T^y} p(y_t|y_1, y_2...y_{t-1}, s) \tag{3.5}$$

3.2.3 Bahdanau attention

Even though the Encoder-Decoder architecture enables modeling complex sequence-to-sequence problems using neural networks, it has an inherent drawback: the path length of information flow is large, from input to output. In some sequence-to-sequence mapping examples, input at the first time step can affect the output at the last time step. In such cases, because of the large path length, the model fails to learn the dependencies. The fixed-length vector acts as an information bottleneck, making it difficult for the model to work effectively when the sequences are long and interdependent. The attention mechanism solves this problem. Rather than using one fixed-length vector to encode all information, the model keeps all the hidden states of the encoder at all the time steps, and the decoder attends to the relevant hidden state while outputting at any timestep. There are different ways to achieve this. Tacotron utilizes Bahdanau Attention, which is described below.

Encoder follows the same equation as equations 3.3 and 3.4. The output probabilities are calculated using an RNN as follows.

$$p(y_t|y_1, y_2...y_{t-1}, x) = f(y_{t-1}, p_t, s_t) \tag{3.6}$$

Where p_t is the hidden state of the decoder RNN computed as

$$p_t = g(s_{t_1}, y_{t-1}, s_t) \tag{3.7}$$

s_t is the fixed-length context vector, which is computed at every time step. It should ideally represent the input hidden state information, which is most relevant to predict the next output. It is calculated as the weighted sum of the encoder hidden state outputs.

$$s_t = \sum_{q=1}^{T^x} \beta_{tq} h_q \tag{3.8}$$

The weights β_{tq} is calculated as,

$$\beta_{tq} = \frac{exp(e_{tq})}{\sum_{q=1}^{T^x} exp(e_{tq})} \tag{3.9}$$

where,

$$e_{tq} = a(h_q, p_{t-1}) \tag{3.10}$$

Here, the weight of any input hidden states depends on e_{tq}, which is a function of the decoder hidden state in the previous time step. The goal of this mechanism is to make the network choose the hidden state, which will give the maximum information to predict the next output. Since the weights e_tq are designed as part of the network, it is learned through backpropagation. As s_t is calculated as an average over encoder hidden states, the alignment between input and output is referred to as soft alignment. Output at time t is dependent on h_q with probability β_{tq}

This mechanism makes the entire input sequence available for the decoder at every timestep, and the pathlengths to decode any output are significantly reduced. Since the encoder and decoder can be made deeper with different architectures, the attention mechanism is used in a wide variety of complex sequences to sequence tasks. Tacotron builds on the Encoder-Decoder architecture with attention, with a sophisticated encoder and decoder design. The architecture of the model is explained below.

3.3 Tacotron model architecture

The Tacotron model architecture was implemented as described in [12]. The architecture is illustrated in Figure 3.1.

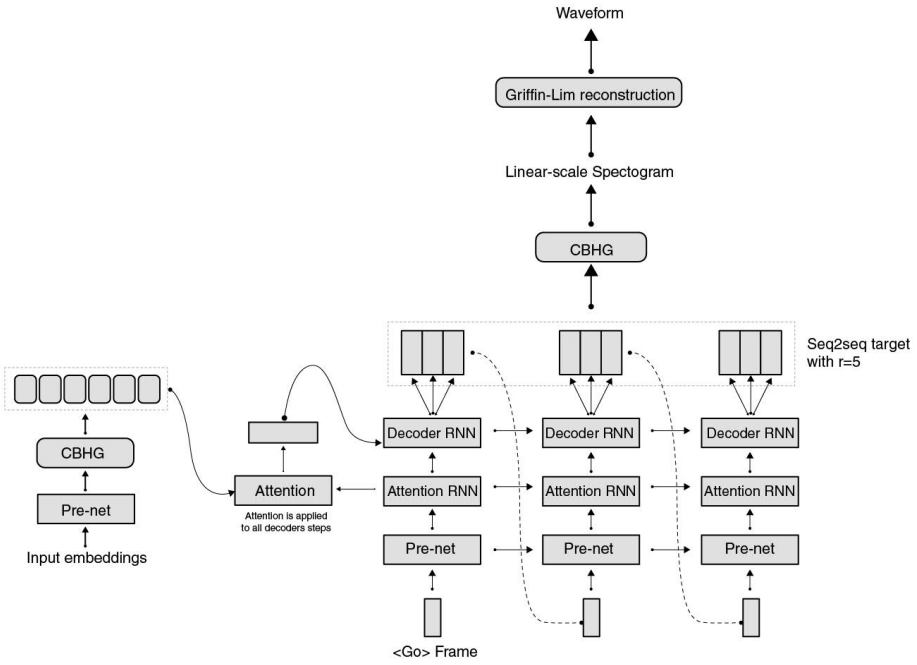

Figure 3.1: Tacotron model architecture

3.3.1 Encoder

The input phonetic transcripts are vectorized as one-hot encoded vectors. The embeddings then pass through two fully connected RELU layers with dropout. Following this, The data is transformed into encoder outputs by the CHBG module which is made up of three layers. The first layer consists of one-dimensional convolutional filters; the second layer is a highway network, and the third layer is a bidirectional gated recurrent net(GRU). The convolutional filters model the contextual features of the input text and the modeling is similar to n-gram language modeling. Highway networks model high-level features; the GRU net model the sequential information present in the text.

3.3.2 Decoder and attention

The target for the decoder is an 80-band Mel-scale spectrogram. Theoretically, even though the whole waveform can be directly mapped as the output, it is not preferred as this complicates the task of learning the alignment between the input characters and the output waveform. Once the Mel-scale spectrogram is recovered, it is straightforward to obtain the waveform in the time domain, through vocoders. The decoder is made to predict five frames of audio for every input time step. This helps to make predictions faster and reduces the computations required to learn alignment. The previous output frame is fed to the attention RNN through a Prenet. While training, the true frame is fed to the attention RNN, and during synthesis, the generated frame is fed to the attention RNN. Based on the principles described in the previous section, a context vector is calculated from the encoder outputs and fed to the decoder RNN. GRU with residual connections is used as the decoder RNN.

The linear scale spectrogram of the waveform is calculated from the Mel-scale spectrogram using a CHBG post-processing net. The time-domain waveform is calculated from the spectrogram using the Griffin-Lim algorithm, which performs the frequency inversion.

The synthesis can be conditioned on the speaker by providing the necessary information to the attention RNNs. The network does not output the stop word; instead, it is made to predict until a fixed time.

3.4 Experiments and results.

3.4.1 Speecon dataset

The Speecon (Speech-Driven Interfaces for Consumer Devices) data-set contains 192 hours of Finnish adult speech data and 12 hours of children's speech data. The data was collected to enable the development of speech applications in consumer devices, and was a collaborative project between corporate companies to collect speech corpus for over twenty languages, with the Finnish speech corpus being recorded by Nokia [28].

The corpus contains speech from 550 adults and 50 child speakers. Among the adult speakers, 276 are Male, and 273 are female. The data was collected using four separate microphones, kept at different distances from the speakers. The data was collected in 4 different environments; offices, home environments with background noise, in cars, and in public spaces. The choice of the

environment was based on the possible use cases of speech applications. The corpus contains 173600 utterances of both reading and spontaneous speech data. The share of spontaneous speech is small, with just 5500 utterances, most of which were usually short, with an average duration of 3 seconds. A histogram of utterance duration in the Speecon database is shown in the figure 3.2. As the target of the synthesizer was to generate short words, the training data matched the requirements of the task.

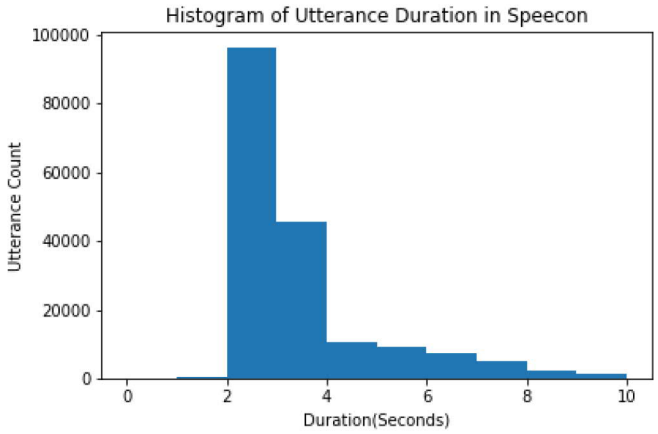

Figure 3.2: Histogram of Utterance Duration in Speecon Database

3.4.2 Training the model

Both input text and output waveform are preprocessed for training. The input text is represented phonetically from its entry in the dictionary. The output audio waveform is divided into 50ms frames with a 12.5 ms frameshift, each frame of which is converted to the frequency domain by 2048-point Fourier transform. The log magnitude spectrogram is obtained with Hann windowing. The model is trained with a batch size of 32. Since the model does not predict the end of synthesis through a stop token, all the inputs and outputs are padded to a max length of 12s. The choice of max length is arbitrary. The model is optimized using Adam optimizer with learning rate decay.

The model is trained on all of the Speecon data, for up to 500k time-steps. Training loss at different steps is plotted in the Figure 3.3. The model starts converging early, and the loss curve almost flattens after 100k steps. Similar

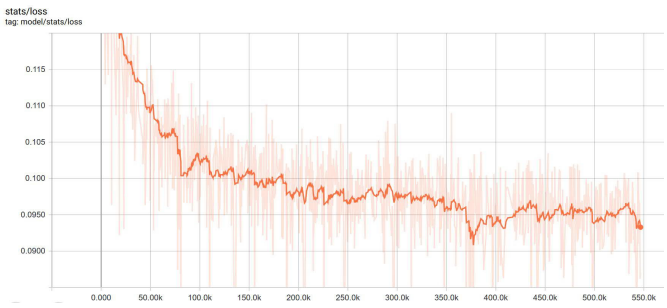

Figure 3.3: Training Loss vs training steps for Tacotron model

loss curves and convergence is observed when the model is conditioned on learned speaker embeddings.

Learning the alignment between the input and output is very important for the model's convergence. The attention weights at different stages of training give a clear indication of how the model is learning the alignment, while completely scattered attention weights indicate that the model has not learned the alignment. As the model is trained, the attention weights start to align in almost a continuous manner. A clear improvement in alignment can be seen as the training steps progress from 10k to 20k to 50k in Figure 3.4.

3.4.3 Evaluation of generated audio

The standard way to analyze the speech synthesis results is through Mean Opinion Scores(MOS). The synthesized audio is rated by human listeners on a scale of 1 to 5 on parameters such as naturalness and voice similarity. The average score measures how good the synthesized speech is on these parameters. Since the goal is to use generated audio to train a Phoneme Recognizer, MOS is not used for Tacotron results. The synthesized audio is analyzed qualitatively and is shared on the website https://sujithpadar.github.io/tacotron/ [29]. Refer the audio samples in the website corresponding to the results discussed below.

The parameters of the analysis were voice similarity and voice consistency. Voice similarity is evaluated by comparing the audio generated for ten different speakers, at random. It is noticeable that the model was able to discern the gender and variability in the voice of different speakers. Even though the voice in the generated audio and the reference are not identical, they sound similar. Voice consistency was evaluated by listening to audio samples generated from the same speaker. It can be seen that the voice for a

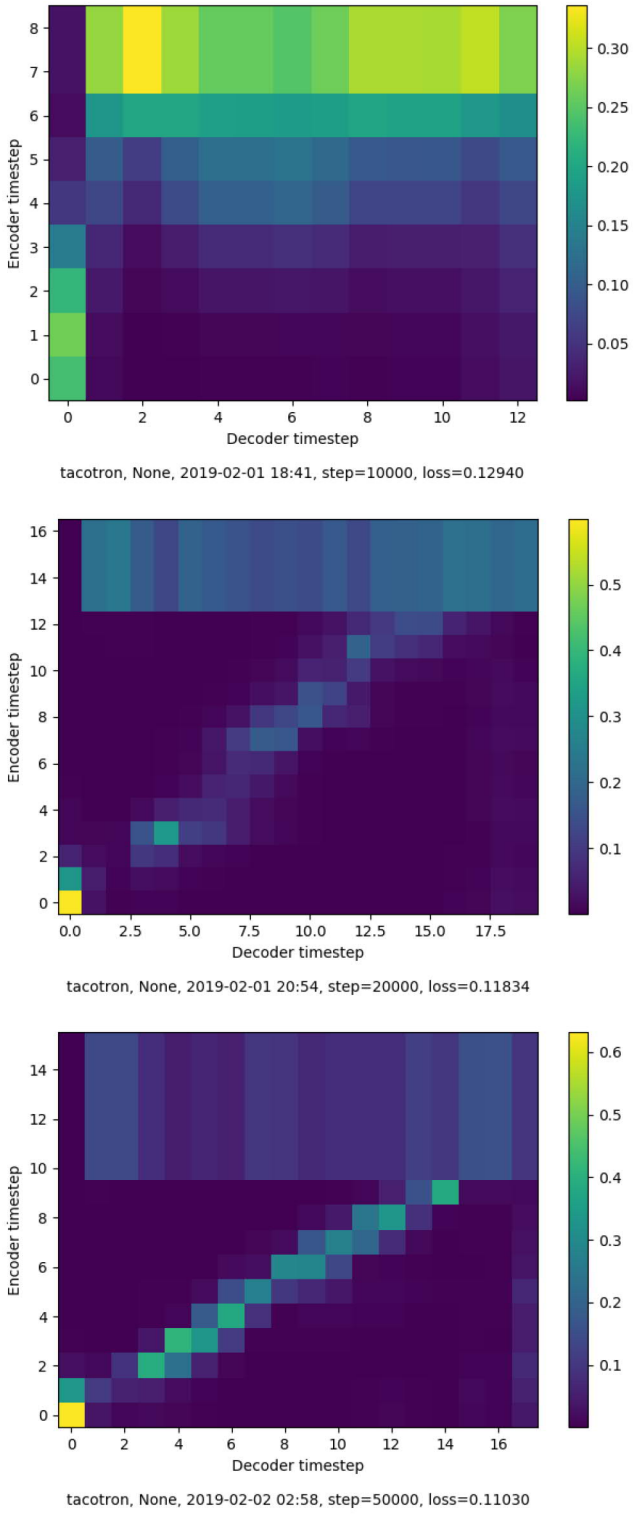

Figure 3.4: Tacotron model learning alignment as training proceeds from 10k steps to 20k, then 50k steps.

particular speaker remains consistent for different input prompts. However, there are a few examples where the voice diverges.

The quality of the output audio degrades as the length of the input text sequence increases. As the training data chiefly consists of small utterances with few words in each, it is natural for the model to follow the training data. This is not a problem for training the Phoneme recognizer for SIAK, as the target of the recognizer is short utterances.

3.4.4 Different input representations

Two different input representations were trialed for the text, to see if it affects the training and convergence. The two configurations are:

- Finnish phonetic representation for the data. Each phoneme is one-hot encoded.
- Free-flowing Finnish text representation, where different characters that make up the Finnish language are one-hot encoded.

It was determined that these two different configurations did not affect the convergence and resulted in similar synthesis results. The results were in line with what was reported in the original publication. [12]

3.4.5 Experiments with children data and adult data

Since the Speecon dataset consists of 192 hours of adult data and just 10 hours of children's speech, it was important to establish if it was possible to synthesize with just 10 hours of children's data. The attention mechanism failed to align the audio and text when only 10 hours of speeconkids speech was used. The shortness and non-variability (children repeat the same text prompt in the data-set.) of input data in children's data make it challenging for the Tacotron model.

3.4.6 Speaker embeddings experiments

It has been shown that the synthesized speech from Tacotron can be conditioned on parameters like the speaker's voice, speaking rate, pitch, and prosody [14], [30]. Voice cloning, which enables the generation of audio from a speaker sample not seen during training,[14] is of interest as it will allow the creation of cloned player audio samples in SIAK player voices. However, these results have been reported on Tacotron-2 and not Tacotron, while the

experiment attempted to replicate the voice cloning results using Tacotron. Tacotron-2 has a much simplified encoder-decoder architecture and uses location sensitive attention over Bahdanu attention.

Speaker embeddings for all Speecon speakers were obtained by using Aalto's Spherediar speaker diarization model[31]. The speaker embeddings are concatenated with encoder outputs. The Tacotron model converged comparatively more quickly with this setup. First, the outputs of this system were compared against model trained with one-hot encoded speaker representation (which does not allow speaker cloning). Both the models fared well in synthesizing audio based on the speakers seen in the training.

When new speaker embeddings that were unseen in training (SIAK speakers, and speeconkids test speakers) were used, while synthesizing, the model failed to generate coherent speech, instead outputting unintelligible utterances. It can therefore be inferred that Tacotron fails to clone voices, and it requires architectural refinements to achieve voice cloning as in Tacotron-2 [14].

3.4.7 Finnish-accented English generation

Knowing that the Tacotron produces good quality audio corresponding to Finnish words does not directly indicate whether it will succeed in generating audio corresponding to English words. These transcripts that are generated as described in section 2.6, are not naturally present in Finnish and might include phonetic distribution, which is not natively present in English. An analysis of phonetic distribution in generated transcript and native Finnish will give conclusive results in this regard, but this is skipped because of the time constraint of the work. These transcripts are given as inputs to Tacotron, and the generated outputs are analyzed. The audio resembles the Finnish accented English spoken by children. The subtle variations in the words were produced successfully by the model and it can be observed that about 10% of the generated audio was incomprehensible, and the generated audio does contain a significant amount of noise. However, this is easy to detect as the Tacotron will produce a long audio sequence which is dis-proportionate to its transcript sequence length when it fails to generate correctly. A total of 60 hours of speech was generated to be used by the recognition system of SIAK.

Chapter 4

Scoring Model

The previous chapter describes how the Finnish accented English speech is generated from SIAK game words. The experiment aims to determine if this synthetic speech will improve the phoneme recognizer used to score the SIAK game words and whether it will improve the correlation of computer generated scores for the SIAK data set with that of a human score. The SIAK game words are scored by building a regression model over the posterior predictions of the Phoneme Recognizer. The architecture of the phoneme recognizer and the experiments and results are discussed in this chapter.

4.1 Recognition Model

The Phoneme Recognizer used for scoring SIAK is a deep GRU neural network with a Connectionist Temporal Classification loss(CTC). The CTC loss is explained in the next section.

4.1.1 Connectionist Temporal Classification

Recurrent neural networks and encoder-decoder models which tackle the sequence-to-sequence problems are discussed in the previous chapter. CTC is an algorithm that resolves the alignment issue between input and output sequences for training a recurrent neural network model. Given an input sequence $(x_1, x_2, ...x_{T^x})$ and an output sequence $(y_1, y_2, ...y_{T^y})$ the aim is to find a mapping between the two sequences. For any X, CTC gives probability distribution over all possible Y's.

The algorithm has an interesting way of achieving this. At every time step t, the model outputs the probability for all possible outputs $p(y|X)$ and an empty token ϵ. ϵ signifies an empty token or the transition from one

output to another. For CTC to work, the possible number of outputs should be finite. The algorithm makes two assumptions regarding the alignment.

- The alignment is monotonic; that is, the current output does not depend on future input. For speech recognition, this can be safely assumed.
- The mapping between the input to output is many to one, i.e., many inputs can correspond to the same output, but a single input cannot correspond to more than one output. This assumption holds for speech recognition as text or phonemes are a compressed representation of speech, and the same phone will correspond to multiple frames of speech.

The alignment and loss are calculated from the probabilities at each time step $p(y|X)$. An output sequence is computed by merging the repeated outputs in the consecutive time steps and removing the ϵ. Multiple sequences of outputs can lead to the same output sequence Y. For example, the output sequences, $y_1, y_1, \epsilon, y_2, y_2$ and $y_1, \epsilon, y_2, y_2, y_2$ both correspond to $Y = y_1, y_2$.

The probability of the output sequence is computed as the product of probabilities at each time step $\prod_{t=1}^{T} p_t(y_t|X)$. Furthermore, the probability of a single output sequence Y is calculated by marginalizing over all the possible sequences, which will result in Y.

$$p(Y|X) = \sum_{A \in A_{X,Y}} \prod_{t=1}^{T} p_t(y_t|X) \tag{4.1}$$

For a training set D, model parameters are learnt to minimize the negative log likelihood,

$$L = \sum_{X,Y \in D} -logP(Y|X) \tag{4.2}$$

If calculated directly, these computations can be too expensive as the probability for all possible Y's needs to be calculated. However, this can be solved by efficient dynamic programming algorithms. At inference time, the most probable output sequence is calculated using the modified beam search algorithm.

4.1.2 Scoring mechanism

For SIAK scoring, a 5-layer GRU network is used with CTC loss as a phoneme recognizer. The model is trained using multi-lingual data, and Adam optimizer with momentum. The n-best list of output sequences is calculated for

each input utterance. For the SIAK game words, the model predicts output probabilities and the possible true pronunciation. The output probabilities are used to build a score regressor, which will try to model the human prediction for the same game words. Different regression methods are used to score the words, including Phonetically weighted Levinstein distance and random forest regression[9]. The next section describes the experiments that have been conducted and the results.

4.2 Experiments and Results

4.2.1 Training data simplification

Previously, a multilingual dataset listed in table 4.1 was used to train the phoneme recognizer, the outputs of which were used to score SIAK game words. A shared phoneme representation was developed by Reima Karhila to transcribe this multilingual speech [9]. This representation was developed by exploring the phonetic feature representation for the phoneme in different languages and mapping the nearest phonemes from different languages together.

Corpus	Duration (Hours)	Type of Data	Language
Wsj1(2 channels) Wsjcam0 (2 channels)	150 30.1	Adult	English
pfstar tidigits*	5.1 2.2	Children	
Speechdat-fi (8-bit) Speecon-fi (4 channels)	54.9 239	Adult	Finnish
Speeconkids-fi (4 channels)	37.5	Children	
spraakbanken-se	23.3	Adult	Swedish

Table 4.1: Training datasets for Phoneme Recognizer in the old setup

Speecon-fi forms the bulk of the adult Finnish speech used in training. It contains around 60 hours of Finnish speech. Speechdat-fi contains 54 hours of spontaneous telephonic speech. Speeconkids-fi contains speech from 50 children between the age of 8 and 12. The total duration of speech in Speeconkids is 37.5 hours.

Wall Street Journal speech datasets(wsj1, wsjcamo) are the English speech datasets which comprise the most significant part of the training dataset.

These contains 180 hours of reading and spontaneous speech from two separate channels. Children's speech is represented by the pfstar dataset, which is a collection of southern UK English children's speech, containing 5 hours of audio recordings.

Swedish speech is underrepresented in the training with just 23 hours of speech from the spraakbanken-se dataset. From each of these datasets, a well defined validation data set and test data set are defined.

The ratio of English data to Finnish data is almost 1:2 in this training setup. This imbalance is caused by using the 2 channels of wsj1, wsj0, and 4 channels of the Speecon dataset. Since the thesis aims to analyze the effects of Tacotron generated data on the model, the above training dataset was simplified. The following changes were made to the training dataset.

- Removing the Swedish dataset entirely from the training process.
- Using only a single channel data for wsj1,wsjcam0, Speecon, Speeconkids.
- Removing Speechdat-fi data from the training.

These simplifications resulted in the training dataset listed in table 4.2.

Corpus	Duration (Hours)	Type of Data	Language
Wsj1 (1 channel) Wsjcam0 (1 channel)	75 15.05	Adult	English
pfstar tidigits*	5.1 2.2	Children	
Speecon-fi (1 channel)	59.75	Adult	Finnish
Speeconkids-fi (1 channel)	9.375	Children	

Table 4.2: Simplified training dataset

By making the above simplification, a better balance between English and Finnish speech datasets is achieved. Although the amount of children's speech is still less compared with adult speech, the ratio of adult speech to children's speech is similar for both languages. To evaluate the effects of this data simplification, the CTC-GRU Model was trained on these two data configurations, keeping the same model hyperparameters. The performance of these models is compared by the evaluation of the Phoneme Error Rate(PER) over the held-out test set.

4.2.2 Effects of training data simplification

Simplification helped with English test set recognition performance.

The PER for English test sets dropped significantly by simplifying training data. The results are listed in table. 4.3.

	All data(PER)	Simplified Data(PER)
wsj1	0.28	0.22
wsjcam0	0.35	0.29
pfstar	0.47	0.43

Table 4.3: Model performance on English test sets, the two models are trained on all data and the simplified data configurations. The test sets are from channel 0 of wsj datasets.

Simplification resulted in slightly worse Finnish recognition performance.

The increase in the recognition accuracy for English datasets comes with a slight reduction in recognition accuracy for Finnish datasets. The results are tabulated in table 4.4. This decrease is not as significant as the increase in accuracy for English datasets.

	All data(PER)	Simplified Data(PER)
Speecon noisy(car)	0.19	0.23
Speecon noisy(public)	0.22	0.25
Speecon clean	0.20	0.22
Speeconkids	0.26	0.27

Table 4.4: Model performance on Finnish test sets, the two models are trained on all data and simplified data configurations, The test data is from channel 0 of Speecon.

It is to be noted that the above results analyze the performance of the model on the test sets corresponding to the data which was present in both the training configurations. It is interesting to see how the model performs on the test sets corresponding to data which was removed from the simplified training.

The GRU model struggles to generalize to datasets on which it is not trained. The GRU model performance on test sets corresponding to data not used in training in the simplified setup is listed in table 4.5.

	Data on which the GRU model was trained	
	All data(PER)	Simplified Data(PER)
Speecon noisy(car)	0.23	0.37
Speecon noisy(public)	0.28	0.37
Speecon clean	0.22	0.31
Speeconkids	0.26	0.29
speechdat	0.33	0.48
spraakbanken	0.58	0.69
wsjcam0	0.36	0.38

Table 4.5: Model performance on test sets corresponding to data not used for training in the simplified setup, channel 1

The model performance is negatively affected when a dataset is removed from training. PER for speechdat increased from 0.33 to 0.48. The speechdat dataset contain spontaneous telephonic speech. Even though Speecon does contain spontaneous speech, it is under-represented. The performance suffering indicates that it is difficult for the model to generalize to a different speaking style if it is not represented during training.

Model performance degrades significantly for the Speecon channel 1 test sets. The position of the recording device changes the dynamics of the audio enough to make it unrecognizable for a model trained on the same data but from a different channel. This indicates that the model generalizes moderately to the recording conditions.

4.2.3 Training data experiments with Tacotron generated data

6 hours of synthetic data corresponding to Finnish children's speech was added to the simplified training dataset. The audio mimics the voice of 50 children present in the Speeconkids dataset, but the audio corresponds to phoneme sequences that are not naturally present in Finnish. The phoneme sequences are generated from the English phonetic representation of words present in the SIAK game and by formulating other representations in Finnish that are statistically closer to it. Previous chapters describe the generation process of the data.

The GRU-CTC Model was trained with multiple configurations of the

simplified data along with 6 hours of Tacotron generated data, and the performance of the resulting models was compared. The different data configurations are:

- Only English datasets. [wsj1, wsjcam0,pfstar,tidigits]
- Only Finnish datasets. [Speecon-fi, Speeconkids]
- English and Finnish datasets. [wsj1, wsjcam0,pfstar,tidigits, Speecon-fi, Speeconkids]
- English and Finnish datasets + Tacotron generated data.

The configuration of the GRU-CTC model was kept the same while training, using all configurations of the data. The model performance was evaluated by calculating the PER over the held-out test set for each of the datasets on which the model was trained. The model converged when trained on each of the above data configurations. PER for test datasets was calculated from the resulting models. The following observations have been made based on the PER results, and the comparisons are restricted to the test sets of the corpus used in training.

4.2.4 Comparison between models trained on only native data and mixed data

The tables 4.6 and 4.7 highlight the effects of training the GRU model with multilingual data compared with only English and only Finnish data.

	Data on which the GRU model was trained	
	Only Finnish	Simplified
Speecon clean	0.26	0.22
Speecon noisy car	0.27	0.23
Speecon noisy public	0.27	0.25
Speeconkids clean	0.25	0.27

Table 4.6: Comparison of PER of test sets when the GRU Model was trained only on Finnish data, and the English and Finnish data

It could be assumed that the models that are trained on the native languages without data from the other would perform better than a mixed dataset as the number of phonemes to predict would be limited when trained

	Data on which the GRU model was trained	
	Only English	Simplified
wsjcam0	0.26	0.29
wsj1	0.16	0.22
pfstar	0.30	0.43

Table 4.7: Comparison of PER of test sets when the GRU Model was trained only on English data, and the English and Finnish data

on languages separately. However, this is not the case. Using English and Finnish datasets for training improved the Finnish recognition performance and worsened the English recognition performance; the PER of Speecon adult test sets decreased from 0.26 to 0.22 for clean data, 0.27 to 0.23 for noisy car data, and 0.27 to 0.25 for noisy public data. Worse still, combining the data negatively affects the phoneme recognition of the children's speech. Combining the data resulted in a worsening of performance in both languages, with Finnish Speeconkids test data seeing an increase in PER from 0.25 to 0.27 and, for pfstar test sets, a drastic increase of PER from 0.30 to 0.43. It is not clear what causes this performance degradation.

4.2.5 Effects of adding Tactron-generated Data

Since Tacotron generated speech is trained on the Speecon database, it is safe to assume that the distribution of speech in the generated data will have some resemblance to that of the Speecon dataset. However, generated data alters the phonetic distribution of the Finnish speech used in training, as the phonetic distribution for these English words is absent in native Finnish.

The addition of Tacotron generated data has shown a significant performance boost to the recognition performance of the Speeconkids test data. The improvements can be seen in table 4.8. The PER for the Speeconkids dataset dropped from 0.27 to 0.13. Similarly, performance gain was seen for the adult Speecon test sets. The PER for the English children's speech in pfstar reduced from 0.43 to 0.38, while the PER for adult English test sets worsened slightly, but insignificantly compared with the massive gains in children's recognition results.

There are a few different ways in which adding the synthetically generated data from Tacotron would have aided the training of the GRU model. Each possibility is discussed in detail below.

The added Tacotron generated data helps to mitigate data im-

	Data on which the GRU model was trained			
	Only Finnish	only English	simplified	simplified and Tacotron generated
Speecon clean	0.26	0.77	0.22	0.20
Speecon noisy(car)	0.27	0.78	0.23	0.20
Speecon noisy(public)	0.27	0.75	0.25	0.21
Speeconkids clean	0.25	0.65	0.27	0.13
wsjcam0	0.79	0.26	0.29	0.30
wsj1	0.80	0.16	0.22	0.24
pfstar	0.80	0.30	0.43	0.38
siak 90+	0.88	0.69	0.74	0.77
Tacotron generated	0.57	0.68	0.58	0.39

Table 4.8: PER Comparison of GRU Models when trained on Different data configuarations.

balance.

The Finnish training data consists of 10 hours of children's speech and 70 hours of adult speech. When 6 hours of synthetically generated children's speech is added to the training, it changes the proportion of child speech in the mix. To check if this is helping the improvement of the recognition results, an experiment was devised. In one configuration, the amount of Speeconkids data is sampled twice compared to adult datasets during training of the GRU model with a simplified data configuration. This configuration is compared with the training data configuration of simplified data and Tacotron-generated data. The table below compares the PER of these two models.

From table 4.9, it is clear that oversampling the Speeconkids speech does not have the same effect as using Tacotron generated speech. Oversampling also resulted in a PER reduction, but it was not as significant (0.27 to 0.23 for Speeconkids) as while using Tacotron generated data (0.27 to 0.13 for Speeconkids). Oversampling Speeconkids speech resulted in the deterioration of English children's phoneme recognition. These results are a clear indication that even though the Tacotron generated synthetic data helps to mitigate data imbalance, it also helps to improve the performance through other means.

	Data on which the GRU model was trained		
	Simplified	Simplified and Speeconkids sampled twice	Simplified and Tacotron generated speech
Speecon clean	0.22	0.23	0.20
Speecon noisy car	0.23	0.24	0.20
Speecon noisy public	0.25	0.24	0.21
Speeconkids clean	0.27	0.23	0.13
wsjcam0	0.29	0.30	0.30
wsj1	0.22	0.24	0.24
pfstar	0.43	0.50	0.38
Tacotron generated	0.58	0.55	0.39
siak 90+	0.74	0.75	0.77

Table 4.9: PER Comparison between oversampling and Tacotron synthtic speech generation

The added Tacotron generated data acts as a regularizer

Even though the Tacotron generated data is trained on the Speecon dataset and provides a close resemblance, it is not a perfect recreation of the original dataset. When a test set created from synthetic data was tested by the GRU model trained on only Finnish datasets, a PER of 0.57 was observed, while the PER for Speeconkids was 0.25. There are three sources of variability in the Tacotron generated data.

- Errors from the sequence to sequence generative model are added to the data. It is not possible to learn a perfect one to one mapping between phonemes and Mel-spectral filter banks.
- The Griffin-Lim frequency to time domain inversion adds artifacts and errors to the generated audio.
- The phonetic distribution of the generated speech is different compared with native Finnish.

The above factors differentiate the distribution of Tacotron generated data from that of Speeconkids. This variability has a regularizing effect and prevents the model from overfitting to training data, improving the generalization performance.

4.2.6 SIAK 90+ test data

Tacotron generated data improves the GRU model's performance on English and Finnish children's speech, but the PER for the test sets of Finnish and English speech datasets is not a direct indicator of how well the GRU model will perform when evaluating SIAK game words. Since there is no phonetic transcript for SIAK game words, it becomes rather difficult to gauge and choose a model from which the game words can be scored. Since the scoring regressor will be built on top of the phoneme recognition results, error from the regression model will be compounded on top of the GRU model error, making it difficult to identify the factors that would improve the system performance. To alleviate this problem, a SIAK 90 test set was created. The assumption here is that all the game words which have achieved a score of above 90 can be assigned to the correct phonetic transcription of the game word. There are 2697 such utterances in the SIAK database, which were selected to form the SIAK 90 test set.

	Data on which the GRU model was trained			
	Only Finnish	only English	simplified	simplified and Tacotron generated
siak 90+	0.88	0.69	0.74	0.77

Table 4.10: PER Comparison of GRU Models when trained on different data configurations on the SIAK 90 test set

All the trained models perform poorly in recognizing the SIAK 90 test set; a dramatic increase in PER is seen for all the models. In table 4.10, it is seen that for the SIAK 90 test set, the best PER is 0.69, which comes from the model which is trained only on English data. All the other models have a slightly poorer performance, indicative of the fact that the SIAK speech does not resemble any of the other training speech. So, the metrics to check if progress has been made to improve the SIAK results are lacking. Modifications in model architecture, or training data which improve the English and Finnish test data results, does not correspond directly with the improvement in the SIAK recognition results.

4.2.7 Regression results

Once the N-best estimation list for each SIAK game word is predicted by the Phoneme recognition model, it needs to be mapped to the human annotated score. Data-driven phonetically weighted Levenstein distance (DDPWLD) measure and Random forest regressor are used to map the N-best list estimation to the scores.

SIAK data is split into train (23488 samples), development (1305 samples), and test sets (4308 samples). For each sample 20 best estimations are saved from the different phoneme recognition models and Levenstein distance is calculated between the prediction and the target sequence. The weights of the distance measure are optimized to reduce the error using the training data. A random forest regressor is built taking the parameters of the Levenstein distance as inputs and the score as the output.

Models	Pearson correlation coefficient on test set	
	Data driven weighted Levenshtein distance	random forest regression
Simplified + Tacotron generated speech	0.379	0.47
Simplified	0.382	0.45
Only English	0.379	0.49
Only Finnish	0.378	0.42

Table 4.11: Comparsion of correlation between human annotated score and model generated score calculated from different models.

The correlation between the human annotated score and the regression outputs for different input model predictions is tabulated in table 4.11. The correlation is weak and is almost invarying for DDPWLD regression, but the results from the random forest regression follow the SIAK 90 phoneme recognition results closely. As in SIAK 90 results, predictions from the model trained only on English has better performance than other models. Using Tacotron generated data in training improves the performance compared to all the data, but the improvement is insignificant. This result reaffirms the observation that Finnish accented English is not equivalent to SIAK game data, and the generated speech reduces the data imbalance in Speecon dataset and not SIAK dataset.

Chapter 5

Discussion

At the beginning of the thesis, the goal was to establish definitive answers for whether in-domain data for the SIAK game be generated using TTS synthesis, and whether synthesized data would improve the performance of an acoustic model and scoring model used to score SIAK game words.

Through the engineering process, these questions were converted into more specific sub-questions:
Is it possible to generate multi-speaker Finnish speech using a Tacotron text to speech system by training it on the Speecon dataset?
Is it possible to generate speech from children's voices, which are underrepresented in the Speecon dataset?
Can children's voices be generated from the SIAK game?
Is it possible to generate Finnish accented English?

Through the subsequent experiments, it was established that a Tacotron text-to-speech system can be used for multi-speaker speech synthesis by specifying the speaker's identity along with the text-speech pair during the training. The speaker identity can be specified either using a one-hot encoded vector or a speaker embedding vector created from a separate model.

The research determined a way to generate Finnish accented English by intelligently mapping Finnish phonemes to English words using the phonological feature representation. In this work, even though the degree of accent cannot be controlled as has been achieved by some of the recent work from Google [32], this method is a simple way to achieve accented English, which does not require model architecture modifications. The degree of naturalness of the accent was analyzed qualitatively by listening to the audio, and mean opinion score tests were not been conducted to validate the claim.

The Tacotron model was trained on an imbalanced Speecon dataset,

which contained only 10 hours of children's speech for 100 hours of adult speech. Using multi-speaker training, the speech was synthesized with voices similar to both children and adults. This data imbalance did not cause any error in speech synthesis.

The generated dataset was considered as a proxy to the training dataset for SIAK. From the phoneme recognition results, it is now clear that the generated data can become a proxy to Speeconkids dataset and not to SIAK.

It was not possible to generate speech corresponding to SIAK, since SIAK data could not be used to train the Tacotron model, as it was not transcribed. Transfer learning was attempted with the help of speaker embeddings to generate speech as in SIAK. Even though the approach was reported to work for a similar Tacotron-2 text to speech models, it failed to succeed in the corresponding experiments with Tacotron-1. The simplifications made in the Tacotron-2 model architecture appear to be vital for the transfer learning approach to work.

After answering the first phases of questions, the resulting key question was whether a speech generated from a text-to-speech system improves phoneme recognition results and helps the SIAK scoring.

The Tacotron generated data has been shown to improve phoneme recognition for the Speeconkids dataset, and the overall performance of the recognitions system. It has been proven that by using text to speech synthesis, the problem of data imbalance can be mitigated through intelligent transfer learning, a conclusion which can be drawn from the PER results of the Speeconkids and pfstar test datasets.

Although the method was proven to reduce data imbalance for in Speecon dataset, the method failed to generalize for SIAK dataset. As the Tacotron generated data was conditioned on Speecon and not SIAK, the PER performance gain seen for Speecon dataset was not observed for SIAK. Contrarily PER performace was affected negatively as seen from the results of SIAK-90. Consequently, the correlation between the human annotated score and model generated score was affected negatively. These results strongly indicate the distributional differences between SIAK and Speecon datasets.

Looking at the experiments on the GRU model, it can be seen that it learns the spectral distribution of the individual datasets on which it is trained, but it fails to interpolate and generalize between them. When different datasets are mixed and the subsequent model is trained, the GRU model learns the distribution of each one of the datasets but fails to generalize beyond the data from which it is trained.

5.1 Future Work

The possibility of using text-to-speech systems to improve speech recognition presents several opportunities to use the same technique in other low-resource settings. The rapid improvement in text-to-speech systems with controllable pitch, accent, and noise levels in recent years has provided ample scope to utilize them for low-resource speech recognition.

The biggest bottleneck in improving the SIAK scoring is the unavailability of phonetic transcription of the speech. This prevents us from using the data in training the recognition models as well as text-to-speech synthesis models. Transcribing the game words will present different possibilities to improve the scoring system. Even if a small portion of the SIAK dataset is transcribed, the techniques mentioned in this thesis can be used to utilize that data to improve the recognition results and hence, the scoring system.

Bibliography

[1] K. Katzner and K. Miller, *The languages of the world*. Routledge, 2002.

[2] L. De Valoes, "Importance of language-why learning a second language is important", *Adjunct Faculty*, 2014.

[3] P. K. Kuhl, "Early language acquisition: Cracking the speech code", *Nature reviews neuroscience*, vol. 5, no. 11, p. 831, 2004.

[4] K. Hakuta, "A critical period for second language acquisition", *Critical thinking about critical periods*, pp. 193–205, 2001.

[5] S. M. Witt *et al.*, "Use of speech recognition in computer-assisted language learning", PhD thesis, University of Cambridge Cambridge, United Kingdom, 1999.

[6] H. Strik, K. Truong, F. De Wet, and C. Cucchiarini, "Comparing different approaches for automatic pronunciation error detection", *Speech communication*, vol. 51, no. 10, pp. 845–852, 2009.

[7] A. Lee and J. Glass, "Mispronunciation detection without nonnative training data", in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[8] S. Joshi, N. Deo, and P. Rao, "Vowel mispronunciation detection using dnn acoustic models with cross-lingual training", in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] R. Karhila, A.-R. Smolander, S. Ylinen, and M. Kurimo, "Transparent pronunciation scoring using articulatorily weighted phoneme edit distance", *arXiv preprint arXiv:1905.02639*, 2019.

[10] R. Karhila, S. Ylinen, S. Enarvi, K. J. Palomäki, A. Nikulin, O. Rantula, V. Viitanen, K. Dhinakaran, A.-R. Smolander, H. Kallio, *et al.*, "Siak-a game for foreign language pronunciation learning.", in *Interspeech*, 2017, pp. 3429–3430.

[11] J. Tao, S. Ghaffarzadegan, L. Chen, and K. Zechner, "Exploring deep learning architectures for automatically grading non-native spontaneous speech", in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 6140–6144.

[12] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, *et al.*, "Tacotron: Towards end-to-end speech synthesis", *arXiv preprint arXiv:1703.10135*, 2017.

[13] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions", in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 4779–4783.

[14] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu, *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis", in *Advances in neural information processing systems*, 2018, pp. 4480–4490.

[15] M. H. Christiansen and S. Kirby, *Language evolution*. OUP Oxford, 2003.

[16] I. P. Association, I. P. A. Staff, *et al.*, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[17] R. Wang, "Recognizing phonemes and their distinctive features in the brain", *Unpublished doctoral dissertation*, 2011.

[18] Y. R. Chao, "The non-uniqueness of phonemic solutions of phonetic systems", *Bulletin of the Institute of History and Language, Academia Sinica*, pp. 363–398, 1934.

[19] V. Fromkin, R. Rodman, and N. Hyams, *An introduction to language*. Cengage Learning, 2018.

[20] G. N. Clements, "The geometry of phonological features", *Phonology*, vol. 2, no. 1, pp. 225–252, 1985.

[21] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks", *Computer Speech & Language*, vol. 14, no. 4, pp. 333–353, 2000.

[22] P. Taylor, *Text-to-speech synthesis*. Cambridge university press, 2009.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[24] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification", *arXiv preprint arXiv:1202.2745*, 2012.

[25] Q. V. Le, "Building high-level features using large scale unsupervised learning", in *2013 IEEE international conference on acoustics, speech and signal processing*, IEEE, 2013, pp. 8595–8598.

[26] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, *et al.*, "Deep voice: Real-time neural text-to-speech", in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 195–204.

[27] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio", *arXiv preprint arXiv:1609.03499*, 2016.

[28] D. Iskra, B. Grosskopf, K. Marasek, H. Heuvel, F. Diehl, and A. Kiessling, "Speecon-speech databases for consumer devices: Database specification and validation", 2002.

[29] *Finnish multi-speaker tacotron results*, Dec. 2019. [Online]. Available: `https://sujithpadar.github.io/tacotron/`.

[30] D. Stanton, Y. Wang, and R. Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis", in *2018 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2018, pp. 595–602.

[31] T. Kaseva, A. Rouhe, and M. Kurimo, "Spherediar: An effective speaker diarization system for meeting data", in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2019, pp. 373–380.

[32] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning", *arXiv preprint arXiv:1907.04448*, 2019.