Low Resource Speech Synthesis

Aidan Pine

THE UNIVERSITY
of EDINBURGH

Master of Science

Speech & Language Processing

University of Edinburgh

2021

Abstract

This dissertation describes the motivation, development and evaluation of speech synthesis systems for three Indigenous languages spoken in Canada; Gitksan, SENĆOŦEN, and Kanien’kéha. I run a variety of experiments to determine the amount of data required to build a neural system of reasonable quality. I evaluate my models using a combination of objective measurements including log Mel spectral error, Mel cepstral distortion, and a custom built acoustic feature classifier, as well as subjective listening tests. I show that in low-resource situations, it is easier to train speech synthesis models with FastSpeech2 than with industry standard conditional autoregressive models with attention like Tacotron2, and that as little as 15 minutes of data produces reasonable quality synthesis. I also show that the common transfer-learning approach to neural low-resource speech synthesis is not necessary with a non-autoregressive model like FastSpeech2. Finally, I present a hybrid neural data augmentation technique using concatenative speech synthesis.

Keywords— speech synthesis, low-resource languages, language revitalization, Indigenous languages

Acknowledgements

This has been an incredibly challenging and rewarding year. I have learned and explored more about speech technology than I ever could have anticipated at the beginning of the year. I am extremely grateful for having had this opportunity to learn about such a fascinating area that combines so many of my interests. I would like to first and foremost thank the communities I have worked with that made this dissertation possible.

Nyawen’kó:wa/Niawen’kó:wa to Owennatekha Brian Maracle, Rohahíyo Brant, & Ronkwe’tiyóhstha Maracle in Six Nations, Akwiratékha’ Martin in Kahnawà:ke, Nathan Thanyehténhas Brinklow in Tyendinaga, and Satewas Harvey Gabriel in Kahnesetà:ke.

To PENÁĆ David Underwood, PENÁW̱EṈ Elliott, SX̱EDŦELISIYE Renée Sampson, & Tye Swallow in W̱SÁNEĆ: JÁN U HÍ,SW̱ḴE MEQ SÁN.

’Wii t’isi’m ha’miyaatxw ’nii’y loosi’m, Barbara Harris, Vincent Gogag, Hector Hill, g̱ant Louise Wilson. Thank you for sharing so much of your beautiful language with me. Thank you as well to Michael Schwan, Clarissa Forbes, Lisa Matthewson, and Henry Davis of the Gitksan Research Lab at UBC for their assistance.

I would also like to thank the Lekwungen, Esquimalt, Songhees peoples on whose land the bulk of my time during this last year was spent.

I owe a huge thanks to my parents Jim & Jan, and sister Maia for providing support in so many different ways this last year, including meals, hugs, air-hugs when physical distancing required it, and putting up with hearing me jabber on about phonological features for far too long.

To my colleagues at the NRC, Anna Kazantseva, Patrick Littell, Eric Joanis, Eddie Antonio Santos, Delaney Lothian, thank you for supporting and inspiring me to do this work. Thank you so much to Marc Tessier, Danny D’Amours and Samuel Larkin for helping me get up and running on the GPSC-C and Trixie. I would particularly like to thank my NRC supervisors Roland Kuhn and Cyril Goutte for encouraging me to pursue this goal and being so invested in my growth and professional development. As well, I would like to thank my friend, colleague, and mentor Mark Turin for writing a letter of support for me.

Finally, to my incredible instructors and mentors this year, Korin Richmond, Simon King, Catherine Lai, Dan Wells, Peter Bell, Sharon Goldwater, Martin Corley, and many others, thank you for sharing your knowledge with me, and for going many extra miles to make me feel connected to this cohort and university despite the distance. To the Speech & Language Processing graduating class of 2021 - we did it! Particular thanks to Eric DeMattos, Elisa Gambicchia, Sambit Paul, Noelia de la Cruz, Matt Maufe, Brent Ho, and Stephen Walters for your friendship, support and inspiration this year.

I dedicate this work to the late, dearly missed, Thelma Blackstock.
Sim ayeehl k'uuhl.

Table of Contents

1 Introduction **1**

2 Background **3**

2.1 Text-to-Speech Synthesis 3

2.1.1 Speech Synthesis for Indigenous Languages 4

2.2 Language Revitalization 5

2.3 Indigenous Languages 6

2.3.1 Kanien’kéha 7

2.3.2 Gitksan 8

2.3.3 SENĆOŦEN 9

3 Data **10**

3.1 Kanien’kéha Resources 11

3.1.1 Kawennón:nis 11

3.1.2 Bible Translations 12

3.2 Gitksan Resources 14

3.3 SENĆOŦEN Resources 14

3.4 Recording Methodology 15

4 Evaluation Methodologies **16**

4.1 Mel Cepstral Distortion 16

4.2 Listening Tests 17

4.3 Phonological Feature Classification 17

4.3.1 Classifier Description 18

5 Experiments **20**

5.1 Baseline 20

5.1.1 Neural Model 20

5.1.2 Concatenative Model 21
5.2 How much data do you need anyway? 23
5.2.1 Background & Motivation 23
5.2.2 Experimental Set-Up 24
5.2.3 Results 25
5.3 Transfer Learning 27
5.3.1 Background & Motivation 27
5.3.2 Experimental Set-Up 29
5.3.3 Results 31
5.4 Data Augmentation 33
5.4.1 Background & Motivation 33
5.4.2 Experimental Set-Up 35
5.4.3 Results 36

6 Conclusion **40**

A Appendix **42**
A.1 Speaker Adaptation 42
A.2 Compute, Accessibility, & Environmental Impact 43
A.2.1 Accessibility, Training & Inference Speed 43
A.2.2 CO2 Consumption 44
A.3 Tips 45
A.4 Extra Implementation Details 46
A.4.1 Database Selection for Concatenative TTS 46
A.4.2 Mel Cepstral Distortion 47
A.4.3 Phonological Feature Classifier 48
A.5 Glossary 48
A.6 Results 50

Bibliography **53**

Chapter 1

Introduction

Consider what is required in order for speech-based communication to work. A speaker decides to utter a word, contracts their diaphragm to pull air into their lungs, and upon exhaling, returns the air through their vocal tract. They then contort their vocal tract in highly specific ways to reach a series of articulatory targets that they have associated with a particular meaning. The flow of air past these orchestrated contortions causes pressure fluctuations at varying frequencies that, upon impinging on the listener's ear drums, are processed and understood to represent the same meaning the speaker intended - magic!

The idea of creating machines to simulate the speech process has origins as early as the 18th century when Hungarian inventor Wolfgang von Kempelen created his "speaking machine" to woo crowds. Speech synthesis has since made tremendous gains, and is now employed to solve real world problems. While von Kempelen's machine attempted to replicate the anatomy required for speech, modern techniques use computers to work with discrete representations of sound and the last decade of improvements to speech synthesis have grown in tandem with the progress of the field of neural network-based machine learning.

Simultaneously, the world is experiencing a crisis of linguistic diversity. It is estimated that over half of the world's approximately 7000 languages will cease to be spoken by the end of the 21st century (Krauss, 1992), and over half the world's population currently speaks one of only 13 languages. The processes that led to this degree of language loss are varied and complex, but many languages have been deliberately targeted by assimilationist policies (Pine & Turin, 2017). This is particularly true for Indigenous languages spoken in Canada. However, Indigenous people are motivated and committed to turning the tide of language loss and revitalizing their languages.

Many involved in language revitalization are turning to technology to help support the formidable task ahead of them (Littell et al., 2018). Speech synthesis could possibly be a valuable tool to supplement text-based technologies with audio and free up speaker time to focus on other tasks like in-person teaching. However, with the vast majority of speech synthesis research pertaining to an extremely small set of languages, and effectively no prior research on applying neural speech synthesis methods to Indigenous languages in Canada, there was no precedent to anticipate whether this approach would be successful and which hurdles would exist.

This dissertation is therefore focused on addressing a variety of outstanding questions about the feasibility of developing speech synthesis tools to supplement Indigenous language revitalization efforts in Canada. Many of the technical issues faced by developing speech synthesis systems for Indigenous languages can be coarsely generalized as problems related to the *low-resource* nature of the languages. That is, limited speakers means limited data, limited eligible participants for evaluation, and limited prior work.

I begin this dissertation by giving a brief background of speech synthesis (§2.1), language revitalization (§2.2), and the three Indigenous languages whose data is used to build the models(§2.3), followed by a brief description and analysis of the data (§3). I then describe the three different methods for evaluating my speech synthesis models including objective acoustic evaluation (§4.1), listening tests (§4.2), and a custom-trained phonological feature recognizer (§4.3). Finally, I set out to address the question of how much data is actually required to build a neural speech synthesis system (§5.2), whether transfer learning is able to reduce the data requirements of a given system (§5.3), and also present a novel data augmentation technique using concatenative speech synthesis (§5.4). Given that the dissertation involves multiple experiments and evaluation methodologies, there is no single dedicated discussion section; rather, discussions are included in each experiment's section.

Chapter 2

Background

2.1 Text-to-Speech Synthesis

Generally speaking, there are historically two main divisions in the approaches to speech synthesis: concatenative synthesis and statistical parametric speech synthesis (SPSS).

Concatenative speech synthesis describes the process of selecting pre-recorded units of speech and rearranging them into novel sequences. Often the points of concatenation or “joins” are smoothed over with some sort of corrective signal processing to mitigate mismatches in F0, energy, or other acoustic properties. This technique is effective and is capable of producing high quality speech, but has numerous downsides including a requirement that all recordings be created from a single speaker, poor handling of out-of-domain utterances, and an increasingly dated software infrastructure.

By contrast, statistical approaches to speech synthesis attempt to estimate the acoustic properties of speech. These methods then generate waveforms by passing the predicted acoustic properties or ‘parameters’ through a vocoder. Earlier techniques used Hidden Markov Models (HMM) to estimate phone durations and either Gaussian Mixture Models (GMM) (Zen et al., 2009) or multi-layer perceptrons (Ze et al., 2013) to estimate the acoustic properties of each phone. A vocoder was then used to generate a waveform based on the speech parameters output by the HMM.

The last few years have shown an explosion of research into purely neural network-based approaches to speech synthesis (Tan et al., 2021). These neural systems - similar to their HMM/GMM predecessors - typically consist of a network predicting the acoustic properties of a sequence of text and a vocoder capable of generating

Character	1-hot Encoding
a	[1,0,0]
b	[0,1,0]
c	[0,0,1]

Table 2.1: A toy example of a possible 1-hot encoding of the characters ‘a’, ‘b’, ‘c’

a waveform from these predicted acoustic properties. In most cases neural speech synthesis systems are trained with labelled data where the inputs are sequences of one-hot character or phone encodings (2.1) and the outputs from the TTS model are (log) Mel-spectral features (henceforth, LMSF) (Tan et al., 2021).

These output features are extracted by applying the discrete Fourier transform (DFT) to overlapping windows of the waveform at fixed widths. The resulting magnitude spectra have a filterbank applied where each filter is spaced according to the Mel scale which has higher resolution at critical bandwidths for speech perception. The natural logarithm of these filterbank outputs are referred to as log Mel-spectral features (LMSF, see Fig 2.1). The secondary network (i.e. the ‘vocoder’) is trained separately to transform log-compressed frequency domain Mel-spectral features to time domain waveforms.

2.1.1 Speech Synthesis for Indigenous Languages

There is extremely little published work on speech synthesis for Indigenous languages in Canada (and North America generally). A statistical parametric speech synthesizer using Simple4All was recently developed for Plains Cree (Harrigan et al., 2019). Although it was unpublished, two highschool students created a statistical parametric speech synthesizer for Kanien’kéha by adapting eSpeak (Duddington & Dunn, 2007). I know of no other attempts to create speech synthesis systems for Indigenous languages in Canada. Elsewhere in North America, some early work on concatenative systems for Navajo was discussed in a technical report (Whitman et al., 1997), as well as on Rarámuri (Urrea et al., 2009).

This dissertation therefore discusses the development of the first three neural speech synthesis systems for Indigenous languages spoken in Canada, as well as the first speech synthesis systems of any kind for SENĆOŦEN and Gitksan and indeed of any Salishan or Tsimshianic language.

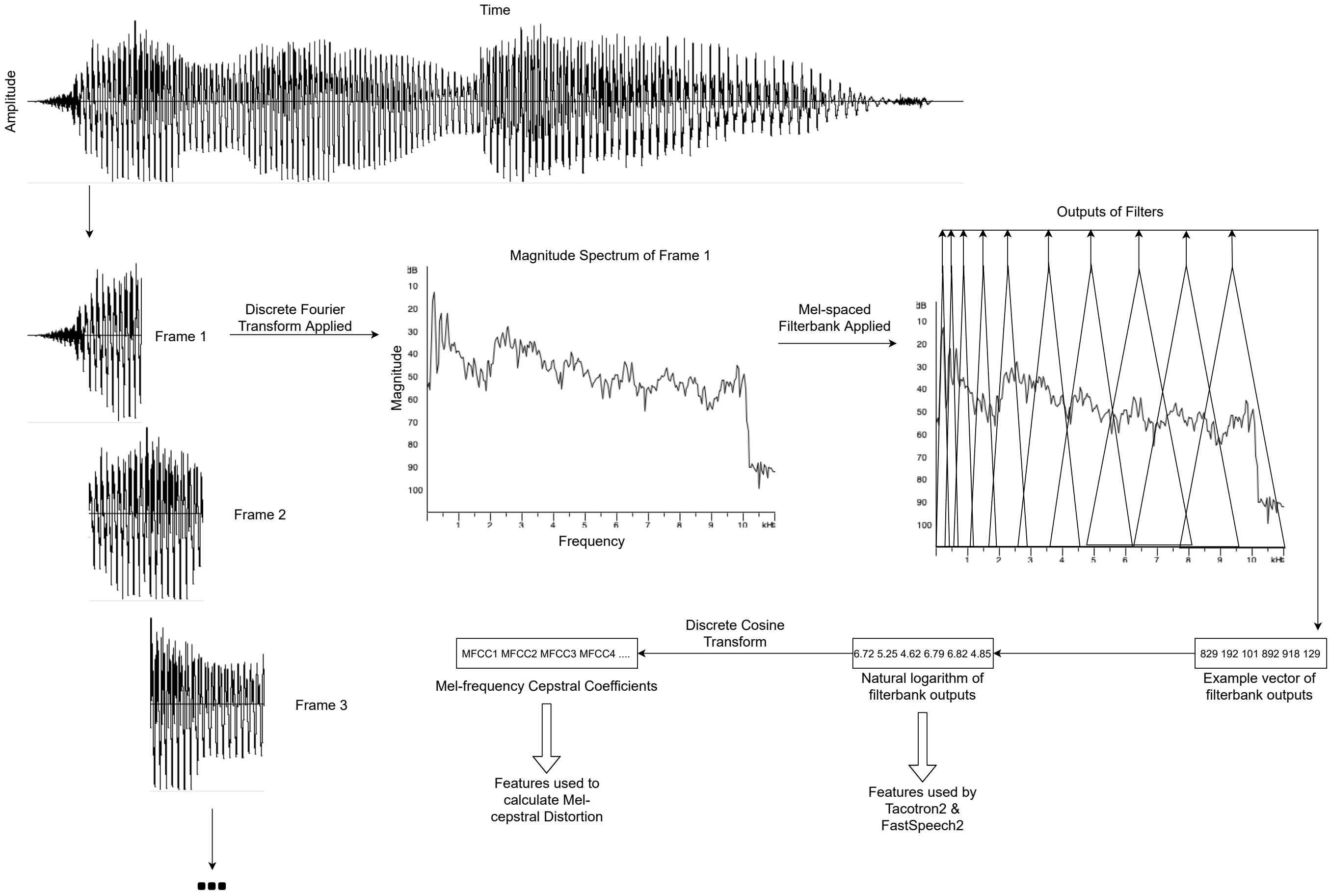

Time
Amplitude
Frame 1
Discrete Fourier Transform Applied
Magnitude Spectrum of Frame 1
dB
10
20
30
40
50
60
70
80
90
100
Magnitude
1 2 3 4 5 6 7 8 9 10 kHz
Frequency
Mel-spaced Filterbank Applied
Outputs of Filters
Frame 2
Frame 3
Discrete Cosine Transform
MFCC1 MFCC2 MFCC3 MFCC4 ....
Mel-frequency Cepstral Coefficients
6.72 5.25 4.62 6.79 6.82 4.85
Natural logarithm of filterbank outputs
829 192 101 892 918 129
Example vector of filterbank outputs
Features used to calculate Mel-cepstral Distortion
Features used by Tacotron2 & FastSpeech2

Figure 2.1: Diagram showing flow for feature extraction from waveform to LMSFs used by popular text-to-speech systems like Tacotron2 and FastSpeech2 and Mel Frequency Cepstral Coefficients (MFCCs).

2.2 Language Revitalization

It is no secret that the majority of the world's languages are in crisis, and in many cases this crisis is even more urgent than conservation biologists' dire predictions for flora and fauna (Sutherland, 2003). However, the 'doom and gloom' rhetoric that often follows endangered languages over-represents vulnerability and under-represents the enduring strength of Indigenous communities who have refused to stop speaking their languages despite over a century of colonial policies against their use (Pine & Turin, 2017).

Continuing to speak Indigenous languages is often seen as a political act of anti-colonial resistance. As such, the goals of any given language revitalization effort often extend far beyond memorizing verb paradigms to broader goals of nationhood and self-determination. The benefits of language revitalization programs can also have immediate and important impacts on a variety of factors including community health and wellness (Whalen et al., 2016; Oster et al., 2014; Marmion et al., 2014; Reyhner, 2010).

The residential school system in Canada tried systemically to sever intergenerational transmission of language and culture among Indigenous communities (Truth and Reconciliation Commission of Canada, 2015); language revitalization seeks to mend those severed relationships and continue community cultural practises. Hallett et al. (2007) showed that this ‘cultural continuity’ is a significant factor in determining the rate of youth suicide on reserves in BC, and they later show that Indigenous language use and revitalization programs are the most significant factors for reducing the rates of youth suicide on reserve, with the rate being lowered from epidemic levels to “effectively zero” in communities with at least 50% of the population reporting some communicative capabilities in the language.

From the Truth & Reconciliation Commission (TRC) report in 2015 which issued nine calls to action related to language, to 2019 being declared an International Year of Indigenous Languages by the UN, there is a growing consensus of the importance of linguistic diversity. The Canadian census reports that the number of speakers of Indigenous languages increased by 8% from 1996 to 2016 (Statistics Canada, 2016). These efforts have been successful despite a lack of support from digital technologies. While there seems to be an opportunity for technology to assist and support language revitalization efforts, these technologies must be developed in a way that does not further marginalize communities (Brinklow et al., 2019). See Section §A.2.1, for a discussion on the accessibility of the systems presented in this dissertation.

2.3 Indigenous Languages

There is remarkable linguistic diversity in Canada, with over 70 distinct Indigenous languages falling into 12 separate language families (Canadian Encyclopedia, 2020). The three Indigenous languages discussed in this dissertation were chosen because they are from three separate families and therefore represent a partial cross-section of the linguistic variation among Indigenous languages and because I have existing relationships with the communities. Due to the wide diversity among Indigenous languages, researchers should be cautious to assume that the results discussed in this dissertation are extendable to other Indigenous languages or that these three languages are interchangeable. Even using a crude metric like Jaccard similarity $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ to measure overlap between phoneme inventories as illustrated in Table 2.2, languages like English and French share more sounds with Gitksan, SENĆOŦEN, and Kanien'kéha than Gitksan and SENĆOŦEN share with Kanien'kéha.

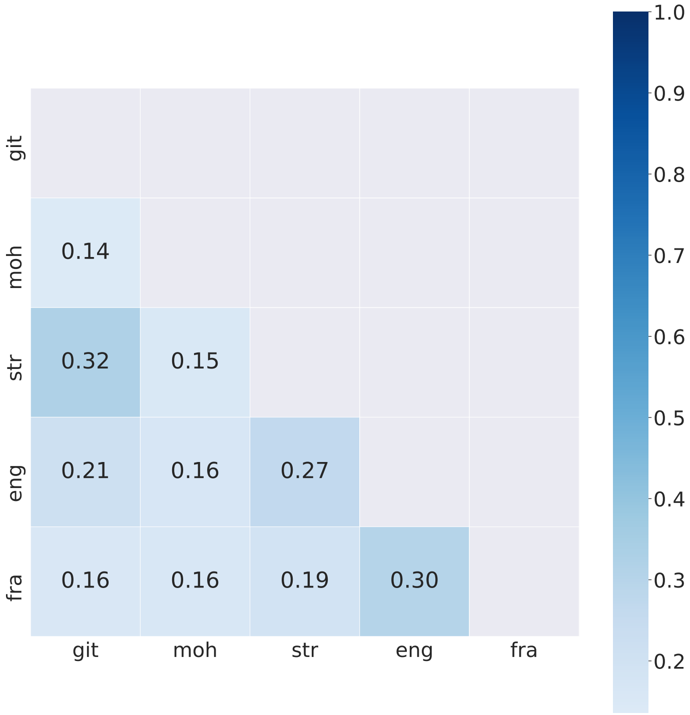


Figure 2.2: Heatmap of Jaccard Similarity between phoneme inventories of English (eng), French (fra), SENĆOŦEN (str), Kanien'kéha (moh) & Gitksan (git) showing for example that Gitksan and SENĆOŦEN have 32% overlap between phoneme inventories.

The following sections give brief descriptions of the three languages discussed in this dissertation.

2.3.1 Kanien'kéha

Kanien'kéha[1] (aka Mohawk) is an Iroquoian language spoken by roughly 2,350 people in southern Ontario, Quebec, and northern New York state (Statistics Canada, 2016). In 1979 the first immersion school of any Indigenous language in Canada was opened for Kanien'kéha, and many other very successful programs have been started since, including the Onkwawenna Kentyohkwa immersion school in 1999.

[1] As there are different variations of spelling, I use the spelling used in the communities of Kahnawà:ke and Kahnesetá:ke spelling throughout this dissertation

Kanien’kéha is quite different from the other two languages discussed in this dissertation. Relevant to speech synthesis, it has a small phoneme inventory with 6 vowels (capable of bearing four separate tones) and only 11 consonants. It is a highly polysynthetic language, with long words often having translations of entire sentences in English as seen below from Kazantseva et al. (2018):

(1) *tetsyonkyathahahkwahnónhne*
te-ts-yonky-at-hahahkw-hnón-hne
both-again-it.to.you.and.I-each.other-walk-purposive-have.gone
DUAL-REP-3SG.N/1.DU.INCL-SREFL-walk-PURP-PPFV

‘the two of us went for a walk’

2.3.2 Gitksan

Gitksan[2] is one of four languages belonging to the Tsimshianic language family spoken along the Skeena river and its surrounding tributaries in the area colonially known as northern British Columbia. Traditional Gitksan territory spans some 33,000 square kilometers and is home to almost 10,000 people, with approximately 10% of the population continuing to speak the language fluently (First Peoples’ Cultural Council, 2018).

The language is part of the Pacific Northwest ‘Sprachbund’ and shares many linguistic features with the other 33 languages spoken in the province, such as robust phoneme inventories with dozens of consonants. Of particular note for speech synthesis are the phonotactics that allow for long sequences of voiceless fricatives as seen below:

(2) *maaxwsxwhl* *xshla’wsxw*
maaxws-xw=hl xshla’wsxw
mæːxʷs-xʷ=ɬ xsɬæwˀsxʷ
snow.on.ground-VAL-CN shirt

‘the shirt is white’

[2] I use Lonnie Hindle and Bruce Rigsby’s spelling of the language, which, with the use of ‘k’ and ‘a’ is a blend of upriver (gigeenix) and downriver (gyets) dialects

2.3.3 SENĆOŦEN

The SENĆOŦEN language is the language spoken by the W̱SÁNEĆ people on the southern part of the island colonially known as Vancouver Island. It belongs to the Coastal branch of the Salish language family and, like Gitksan (§2.3.2), is part of the Pacific Northwest 'Sprachbund'.

It is distinguished by both a world-famous language revitalization program[3], and an orthography developed by the late SENĆOŦEN speaker and W̱SÁNEĆ elder PENÁĆ Dave Elliott. While the community of approximately 3,500 has fewer than 10 fluent speakers, there are hundreds of learners, many who have been enrolled in years of immersion education in the language (First Peoples' Cultural Council, 2018).

[3] `https://wsanecschoolboard.ca/sencoten-language/`

Chapter 3

Data

The term ‘low-resource language’ is an umbrella term which is not terribly discerning. The NLP community overwhelmingly focuses on English, and by comparison the vast majority of the worlds’ languages can be described as ‘low-resource’. The term typically refers to languages that do not possess the ideal amount of data required by a particular machine learning algorithm. However, the term can also be used to describe other limited resources including fewer linguistically descriptive materials, fewer digitized or computerized materials, a low number of speakers, or an economically disadvantaged speaker-population with limited free time (Maguer-esse et al., 2020; Singh, 2008).

Given the background described in §2.2, many Indigenous languages in Canada can be considered ‘low-resource’ in multiple senses. Two interrelated factors contributing to the low-resource nature of many Indigenous languages in Canada are a low number of speakers proportional to the population and a low amount of data. There are no corpora of transcribed audio for many Indigenous languages; fewer still have such data recorded in a studio context. Due to the limited number of speakers, creating these resources is non-trivial; there are limited amounts of text from which a speaker could read, and there are not many people literate in the language who would be able to transcribe. In addition, re-focusing speakers’ limited time to these tasks presents a significant opportunity cost - they are often already over-worked and over-burdened in under-funded and under-resourced language teaching projects.

Therefore, a significant part of this dissertation is focused on techniques which are effective with limited amounts of data. The following sections describe the origin and characteristics of the data used throughout this dissertation.

3.1 Kanien’kéha Resources

There are relatively many resources available for Kanien’kéha when compared with other Indigenous languages in Canada. As such, availability of data and resulting systems should be seen as a sort of topline for what can be expected for other languages (perhaps with the exception of Inuktitut and Cree).

3.1.1 Kawennón:nis

The first set of data comes from recordings made in connection with the Kawennón:nis (lit. it makes words) project. The Kawennón:nis project started as a joint effort between the National Research Council Canada and the Onkwawenna Kenyohkwa immersion school in Six Nations of the Grand River. The tool is an online verb conjugator built with a finite state transducer modelling a subset of the inflectional paradigms in the language.

Kawennón:nis is currently capable of producing 247,450 unique conjugations. The pronominal system is largely responsible for the highly productive set, as in transitive paradigms, agent/patient pairs are fused as shown below in examples from Kazantseva et al. (2018):

(3) ***Se**nòn:wes*
se-nonhwe’n-s
you.to.it-like-habitual
2.sg.agent-like-hab

‘**You** like it.’

(4) ***Sa**nòn:wes*
sa-nonhwe'n-s
it.to.you-like-habitual
2.sg.patient-like-hab

‘**It** likes you.’

(5) ***Take**nòn:wes*
take-nonhwe'n-s
you.to.me-like-habitual
2.sg/1.sg-like-hab

‘**You** like **me**.’

In user evaluations of Kawennón:nis, students often asked whether it was possible to add audio to the tool. With 247,450 unique conjugations, assuming a rate of 200 forms/hr for 4 hours per day, 5 days per week, this would take over a year of dedicated recording from one of the roughly 2,350 people who speak the language

- hardly an advisable use of time for speakers who might better spend their time in classrooms speaking with students. Considering Kawennón:nis is anticipated to have over 1,000,000 unique forms by the time language modelling work is finished, recording audio manually becomes not only ill-advised, but infeasible.

The question that then catalyzed my interest in speech synthesis was ‘what is the smallest amount of data needed in order to generate audio for all verb forms in Kawennón:nis’. The first set of data therefore comes from 852 recordings I initially made to build a concatenative speech synthesis system capable of domain coverage for Kawennón:nis. Utterances were recorded[1] by renowned Kanien’kéha speaker and educator Akwiratékha’ Martin. See §A.4.1 for discussion on how these utterances were selected.

3.1.2 Bible Translations

Since the late 1990s a team of five Kanien’kéha translators worked with the Canadian Bible Society to translate and record audio for parts of the Bible. Of the translation team, Satewas Harvey Gabriel of Kanehsatà:ke is the only living speaker on the recordings. Translation runs in Satewas’ family, with his great-grandfather also working on Bible translations in the 19th century. There are a total of 24 hours of audio recorded by Satewas and his collaborators. Later, a team of three speakers and learners including Nathan Thanyehténhas Brinklow created transcriptions in ELAN to align the text and audio at the utterance level.

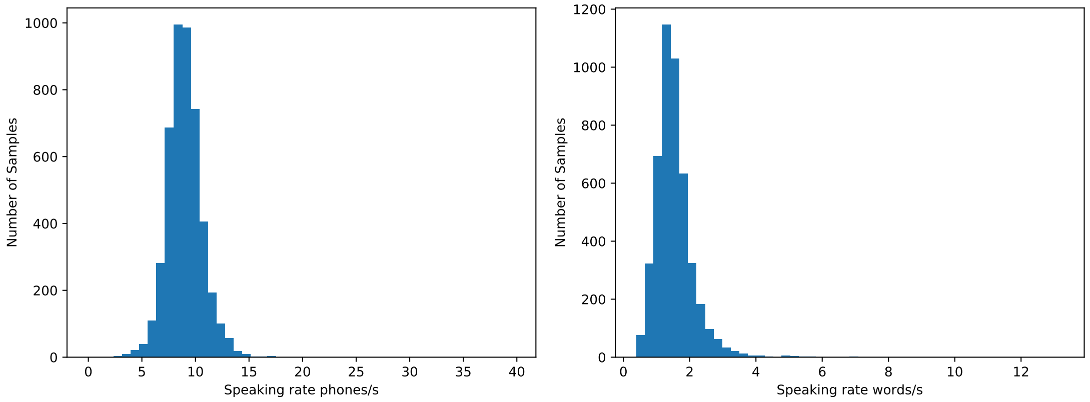
While some books, like the book of Daniel and Jonah were recorded solely by Satewas, others, like Genesis, were recorded by multiple speakers. Because the other speakers had passed away, it was only deemed appropriate to use recordings from Satewas, and so the utterances had to be labelled by speaker. In order to do that I trained a GMM-based speaker identification system following Kumar (2017) and ran it on the entire corpus.

The resulting corpus contained 6 hours of speech spoken by Satewas. I then analyzed the corpus for speaking rate (Figure 3.1), phone coverage (Figure 3.3), duration, and F0 features.

Removing outliers in duration (less than 0.4s or greater than 11s), and speaking rate (less than 4 phones per second or more than 15 phones per second), as well as recordings from the book of Corinthians which had an unknown phase effect present,

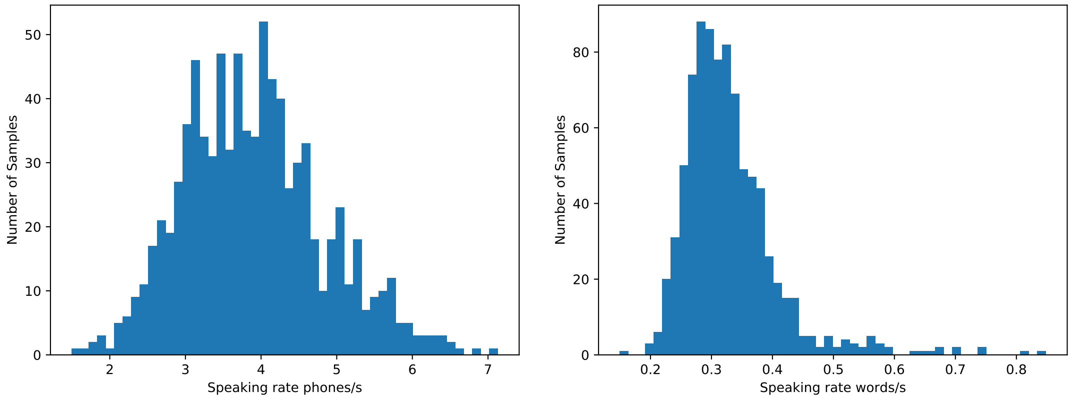
[1] See §3.4 for recording methodology

resulted in a corpus of 3.46 hours of speech. Utterances with Biblical words that contained characters not present in Kanien'kéha were removed (i.e. Euphrades). Other proper names and Biblical words were also removed.



(a) Speaking Rate (phone) (b) Speaking Rate (word)

Figure 3.1: Analysis of Speaking Rate for Bible Translation Kanien'kéha corpus



(a) Speaking Rate (phone) (b) Speaking Rate (word)

Figure 3.2: Analysis of Speaking Rate for Kawennón:nis Kanien'kéha corpus

Compared with the audio from Kawennón:nis as seen in Figures 3.2, the speaking rate is much faster. As mentioned, there is a distinct difference between the recordings; the Bible translation recordings were of continuous speech, whereas the Kawennón:nis recordings were of single words (often pronounced quite carefully).

As seen in Figure 3.3, some phones have a context-sensitive distribution which resulted in very low coverage in the corpus, as seen in both [f] and [ʃ] which occur only two times.

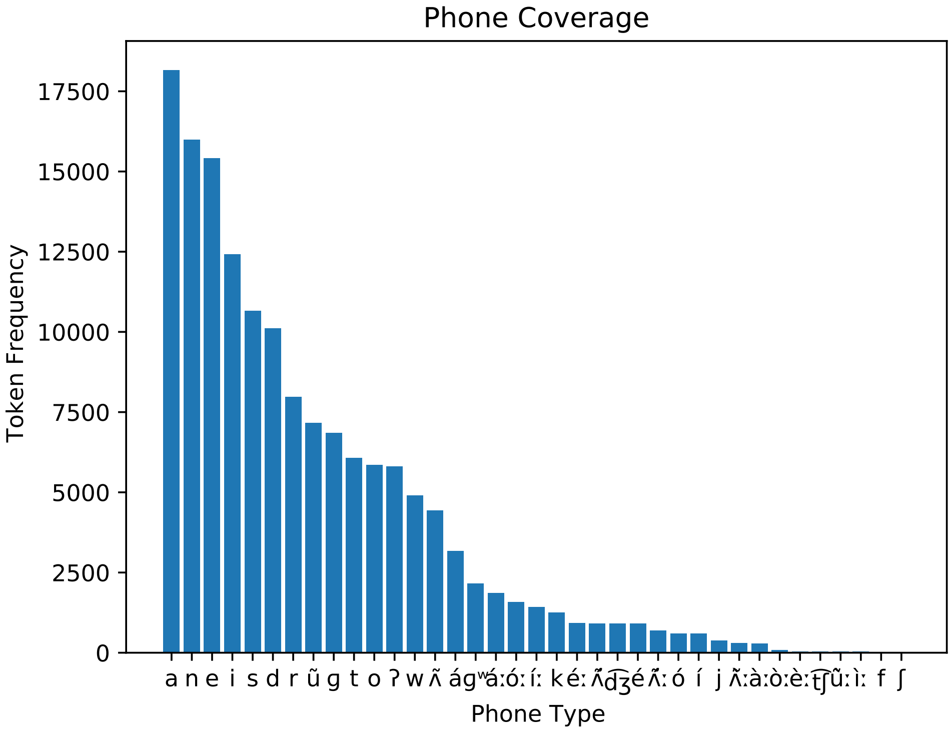


Figure 3.3: Phone coverage for Bible Translation Kanien’kéha corpus

3.2 Gitksan Resources

As there are no studio quality recordings of the Gitksan language publicly available, and as an intermediate speaker of the language, I decided to record a sample set myself. In total, I recorded 35.46 minutes of audio from reading isolated sentences from published and unpublished stories (Forbes et al., 2017).

3.3 SENĆOŦEN Resources

As there were no studio quality recordings of the SENĆOŦEN language publicly available, I recorded 25.92 minutes of the language from PENÁĆ David Underwood reading two stories originally spoken by elder Chris Paul. The recordings had a noticeable hum from the recording room’s fan/ventilation system. To remove this, all waveforms were converted to raw PCM and then passed through RNNoise, a recurrent neural network with gated recurrent units trained for the purpose of denoising audio (Valin, 2018).

3.4 Recording Methodology

Recordings made and described in this paper were made using a MacBook Pro running Speech Recorder (Centre for Speech Technology Research, 2015). A Zoom H4nPro recorder was used as an interface and an Audio Technica AT2020 large diaphragm condenser microphone was mounted on a shock mount and fitted with a pop filter. Recordings were done at 16-bit/48kHz and saved in a lossless raw wave format. Efforts were made to maintain consistent volume and 20cm distance from the microphone throughout the recording sessions.

Chapter 4

Evaluation Methodologies

One of the most significant challenges in researching speech synthesis for low-resource languages is the ability to evaluate the models. For some Indigenous languages in Canada, the total number of speakers of the language is less than the number typically required for statistical significance in a listening test.

I have adopted three separate evaluation methodologies in the hopes of triangulating an accurate summary for the experiments discussed in the following chapter (§5). These methodologies include reporting both log Mel spectral feature error (henceforth 'LMSF error') and Mel Cepstral Distortion (commonly 'MCD') (§4.1), canonical subjective listening tests with limited participants (§4.2), and a custom-trained phonological feature classifier (§4.3).

4.1 Mel Cepstral Distortion

There are a number of techniques for evaluating synthetic speech based purely on the acoustic signal. Mel Cepstral Distortion (MCD) quantifies the difference between two sequences of Mel cepstra and is a popular technique which has been shown to more closely reflect the intuitions of the results of listening tests than the comparison of plain spectral features (Kubichek, 1991, 1993). This is at least partly due to the fact that calculating distance between features that are warped to the Mel-scale ensures that differences that occur in the most perceptibly salient frequency range (roughly < 1000 Hz) are weighted more heavily than higher frequencies.

The loss function for the FastSpeech2 architecture used by most of the experiments in §5 is determined by combining five separate losses; the mean squared error $\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$ between the predicted values y and ground truth calculated values

x for each of the variance predictors (duration, 'pitch', energy), as well as the mean absolute error $\frac{1}{N}\sum_{i=1}^{N}|(x_i - y_i)|$ between the ground truth calculated spectral features x and the predicted spectral features y in both the decoder and residual postnet layers; see (§5.1.1) for a summary of FastSpeech2's architecture.

Since FastSpeech2's log spectral features are already warped to the Mel scale, its Mel spectral loss could be a sufficient evaluation metric. However, in order to separate spectral error from the correlated F0 (i.e. 'pitch'[1]) error, I calculate the MFCCs from the log Mel spectral features using the PyTorch optimized torch-dct library (Hu, 2021) and use MCD as a distance metric in addition to reporting the mean absolute log Mel spectral error from the residual postnet output. Please refer to §5.1.1 for an in-depth description of FastSpeech2's architecture (including the postnet layer) as well as §A.4.2 for extra implementation details of MCD.

4.2 Listening Tests

The standard approach for evaluating speech synthesis systems for naturalness and intelligibility is through listening tests. Due to the limited number of speakers for the languages discussed in this dissertation, it was infeasible to conduct listening tests with the large number of participants needed for statistical significance. Nevertheless, three surveys were designed and implemented for Gitksan, Kanien'kéha, and restricted data experiments with English. All surveys were posted for at least one week or until the maximum participants had contributed. The surveys were expected to take participants 30 minutes or less and were designed and implemented using Qualtrics. Participants were paid $10 for their time. Further information about each listening test is described in the relevant sections (§5).

4.3 Phonological Feature Classification

A final method for evaluation was developed using Time Delay Neural Networks (TDNN) following King & Taylor (2000) who used neural networks to detect phonological features. Phonological features have a long history in the study of phonology, and refer to the fact that any particular sound can be distinguished by a unique configuration of features related to the sound's place and manner of articulation (Chomsky

[1] Unfortunately there is a lot of fuzzy terminology that conflates F0 with its perceptual correlate 'pitch'. I use 'F0' except when specifically referencing the paper.

& Halle, 1991). As discussed in §5.3, some of the models developed for this dissertation encode the input text as sequences of phonological feature vectors. The goal of this evaluation method is to correctly classify frames of synthesized speech into their phonological feature representations. A more in-depth discussion of how this phonological feature encoding works is found in §5.3, and an example of the encoding of a few segments is found in §A.3.

4.3.1 Classifier Description

The classifier contains 5 TDNN layers, followed by two fully connected layers. The input to the network is a 15 x 80 matrix where 15 is the number of frames of a particular phone and 80 is the number of bins from the Mel-scaled filterbank. The output label is a 72 dimensional vector representing a 2-bit encoding of the phonological feature label for that phone. I use the same 2-bit encoding described by Zhu et al. (2021) in order to map the 36 dimensional phonological feature vector (which can have values of -1, 0, or 1) to a binary vector as seen in Table 4.1.

PF Value	2-bit Encoding
1	[1,0]
0	[0,0]
-1	[0,1]

Table 4.1: Phonological Feature (PF) Vector values and their corresponding 2-bit encoding for use in training a PF classifier for evaluation

The data was created by taking both the log Mel-spectral features and durations calculated during preprocessing in the FastSpeech2 pipeline. Then, for each utterance in the training set, each phone is separated into its own training example, where the skip size is set linearly to extract 15 frames. Phones for which there are fewer than 15 frames are padded.

The training data for the classifier used the combined data from Gitksan, SENĆOŦEN, and Kanien’kéha, as well as the English LJ corpus resulting in 160,140 training examples. The classifier was implemented in PyTorch and was trained for 10,000 steps using Adam optimization with a learning rate of 0.001 and a batch size of 256. As each sequence of frames results in a multiclass binary classification, the model was trained using a binary cross entropy loss function combined with the sigmoid function as seen in 4.1 (PyTorch, 2021).

$$l(x, y) = L = \{l_1, ..., l_N\}^T, l_n = -w_n[y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n))] \quad (4.1)$$

where N is batch size and σ is the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$

Because the sigmoid function is incorporated in the loss function, the final layer of the network was not followed by an activation; all other layers were followed by rectified linear unit (ReLU) activations. The implementation of the TDNN followed Luu (2021) and the model design decisions are summarized in Table A.2. Results are reported below in Table 4.2.

	Accuracy	Hamming-0	Hamming-3
Test Set	98.06%	85.90%	92.41%

Table 4.2: Phonlogical Feature Classifier Results. Accuracy is the mean accuracy of the classifier across each of the 36 phonological features. The 'Hamming-0' and 'Hamming-3' are the percentage of samples with a Hamming distance of 0 and 3 respectively, meaning the percentage of phones where 100% of the features, and at least 33/36 of the features were predicted correctly. Results are reported from running the classifier on a 16,000-sample held out test set.

As this method of evaluation was not the focus of this dissertation, it was implemented mostly as a proof of concept. With more time, a proper hyperparameter search could have been conducted and more sophisticated models like convolutional recurrent neural networks as discussed by Qamhan et al. (2021) could also have been implemented.

Chapter 5

Experiments

5.1 Baseline

5.1.1 Neural Model

There are a variety of contemporary ‘off-the-shelf’ neural models to choose from. While many researchers currently choose the Nvidia reference implementation of Tacotron2 (Shen et al., 2018), preliminary experiments I conducted showed that the attention mechanism did not learn properly with less than 10 hours of data using the default hyperparameters (§5.2).

Because none of the datasets for Kanien’kéha, SENĆOŦEN or Gitksan are even close to 10 hours in length, I needed to use a different architecture. Instead of Tacotron2, my baseline neural model is FastSpeech2 (Ren et al., 2021). I use the open source implementation with the default hyperparameters[1] unless otherwise specified (Chien, 2021). The implementation differs from the paper in a few key ways, firstly, it does not include the option to decode directly to waveform (FastSpeech2s), thus a vocoder was used for all experiments, namely, HiFi-GAN (Kong et al., 2020). Second, a residual “post-net” network styled after Tacotron2 was used. Finally, the ‘pitch’ values calculated for use in the pitch variance predictor were F0 values of the original signal - whereas later versions of the paper use pitch spectrograms from a continuous wavelet transform.

For the vocoder, I used the ‘Universal’ pre-trained HiFi-GAN model which was trained on the LibriSpeech, VCTK, and LJSpeech datasets. I experimented with finetuning the vocoder with Kanien’kéha, SENĆOŦEN and Gitksan data, but found

[1] See Appendix for full list

no discernible difference when comparing copy synthesized samples between the models.

I implemented a number of changes to the architecture as part of experiments discussed in §5. All changes to the system were implemented as optional changes that could be controlled through the use of the model configuration file and will be made publicly available in my fork of the implementation's repository[2]. In total, I implemented a multi-speaker architecture (§A.1), a multilingual architecture (§5.3), and an option to change the input from one-hot character embeddings to multi-hot phonological feature vectors (see §5.3 for discussion). As explained in §A.2.1, in order to reduce the number of model parameters, I reduced the number of layers in the decoder, removed the energy variance adaptor, and refactored all convolutional layers in the encoder, decoder and variance adaptors to perform depthwise separable convolutions. I also changed the size of the kernels in the encoder and decoder convolutional layers to match the model described in Luo et al. (2021). The changes to the original FastSpeech2 architecture implemented by me and Chien (2021) are summarized in Figure 5.1.

5.1.2 Concatenative Model

There are clear benefits to neural TTS systems, including improved generalization, and the ability to use multi-speaker data. However, in low-resource, limited domain contexts - such as single word synthesis for Kawennón:nis- it was not clear that a neural system would outperform a concatenative one. For that reason, and for the hybrid system described in §5.4, a concatenative system was built for both Kanien'kéha and Gitksan using Festival (Taylor et al., 1998) and MultiSyn (Clark et al., 2007) following the recipe found on Speech Zone (Simon King, 2021). Instructions for customizing Festival's phoneset and lexicon to be used with new languages were followed from the CMU Talking Clock tutorial (Black, 2017).

Festival and HTK only work with ASCII-encoded characters, but both languages here use characters in their writing system that are not ASCII-compliant. As a result, ad-hoc mappings from the orthography to an ASCII-compliant form were devised and published for both Gitksan and Kanien'kéha as a mapping in the *g2p* Python library (Pine & National Research Council Canada, 2021).

In order to create the lexicon for these concatenative systems, custom entries for

[2] https://github.com/roedoejet/FastSpeech2

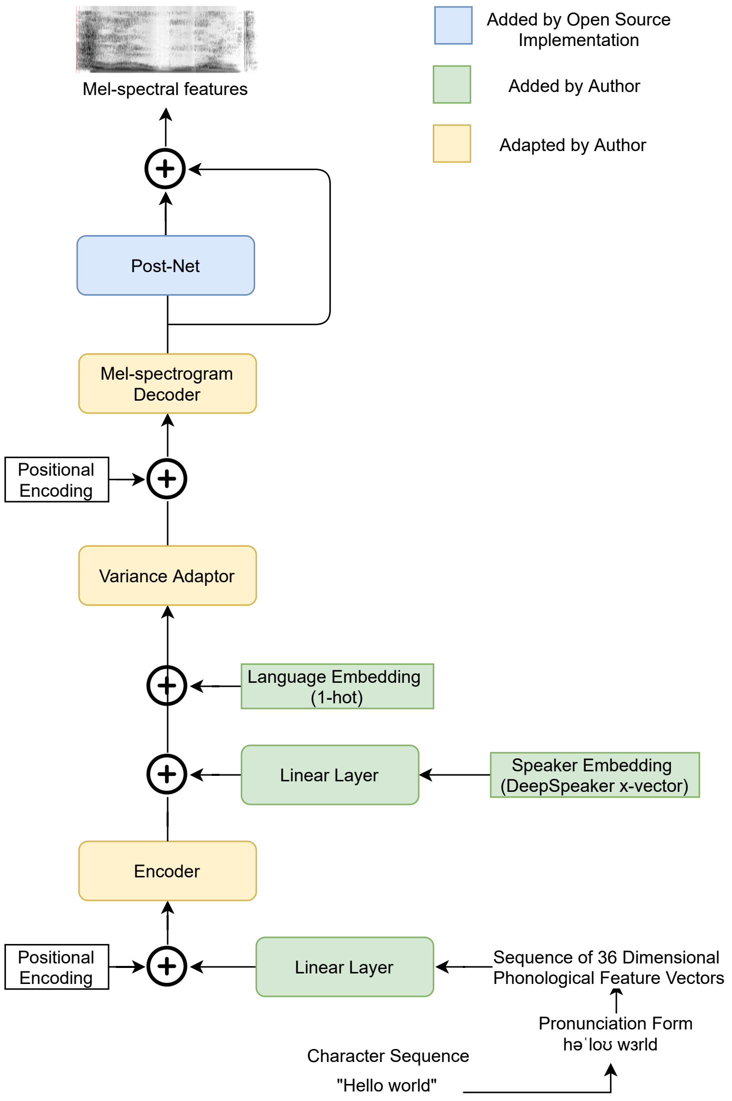


Figure 5.1: Adapted FastSpeech2 Architecture. A residual postnet module was added by the open source implementation's author. I further adapted the architecture to allow for zero-shot multispeaker inference, multilingual training, multi-hot phonological feature inputs, and refactored the encoder, decoder and variance adaptor layers to use fewer parameters. $\bigoplus$ represents a summing of tensors.

each word must be created with syllabification information. Part of speech for each lexical entry was left as 'nil', but syllabification information was added by adapting the SyllabiPy (Hench, 2019) zero-shot syllabification algorithm based on the sonority hierarchy of the target language phoneme inventory.

As part of the Kanien'kéha A/B listening test created in §5.3, I added 10 questions comparing single words synthesized from the concatenative baseline and the neural baseline trained with 3 hours of data. The results of this test, which unfortunately only recruited 6 participants, were that the neural model was preferred 72.22% of the time over the concatenative model. Further evaluation is needed confirm this preliminary preference for the neural baseline.

5.2 How much data do you need anyway?

5.2.1 Background & Motivation

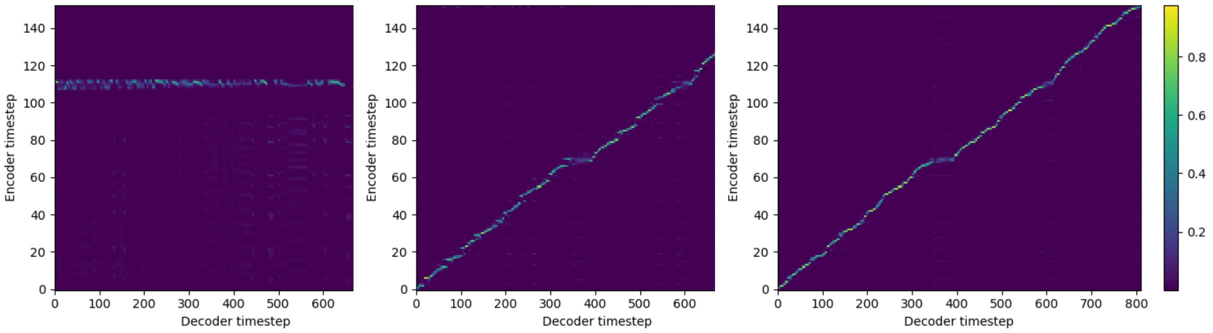
The first question to answer is whether the corpora ranging from 25 minutes to 3.46 hours described in §3 are sufficiently large for building neural speech synthesizers. Due to the immense popularity of Tacotron2, many people have assumed that the data requirements for training a Tacotron2 model are synonymous with the data requirements for training *any* neural speech synthesizer of similar quality. As a result, some researchers still choose to implement either concatenative or HMM/GMM based statistical parametric speech synthesis systems in low resource situations based on the assumption that a "sufficiently large corpus [for neural TTS] is unavailable" (James et al., 2020, p.298). However, conditional autoregressive (CAR) models like Tacotron2 should not be used as a benchmark for data requirements among all neural TTS methods. They are notoriously difficult to train, and, as I argue in this section, unnecessarily inflate the data requirements for training.

In traditional HMM/GMM-based statistical parametric speech synthesis the output is determined by both an acoustic model and a duration model. The trained acoustic model generates the most likely set of speech parameters given an input character, and the duration model generates the most likely number of speech frames (duration) of that input character. The result is a sequence of speech parameter frames $\mathbf{y} = \{y_1...y_T\}$ predicted from a sequence of text $\mathbf{x} = \{x_1...x_N\}$ where N is equal to the number of time steps in the input text and T is equal to the sum of the predicted durations of $\mathbf{x}$. This model assumes conditional independence between frames which is problematic given the highly correlated nature of speech frames due to context sensitive phonological processes like nasal place assimilation or uvular vowel lowering (Fortier, 2016). To address this faulty independence assumption, autoregressive models like Tacotron2 try to predict the sequence of speech parameters y_t from both the input sequence of text x and the previous speech parameters $y_1, ..., y_{t-1}$; in other words the model is autoregressive. In addition, duration is modelled as a latent variable z through an attention network, giving $P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} P(y_t|y_1, ..., y_{t-1}, z|x)$. However, the autoregressive part of the feature prediction network used by Tacotron2 uses ‘teacher-forcing’ meaning that during training the autoregressive frame y_{t-1} passed as input for predicting y is taken from the ground truth label and not the prediction network’s output from the previous frame $\hat{y}_{t-1}$. As discussed at length by Liu et al. (2019), such a system might learn to copy the teacher forcing input or disregard the text entirely, which could still optimize Tacotron2’s root mean square error function, but result in an untrained or degenerate attention network and thus be unable to properly generalize to new inputs at inference time when the teacher forcing input is unavailable.

There have been many proposals to improve training of the attention network, for example by guiding the attention or using a CTC loss function to respect the monotonic (i.e. order- but not time-synchronous) nature of the alignment between text inputs and speech outputs (Tachibana et al., 2018; Liu et al., 2019; Zheng et al., 2019; Gölge, 2020). As noted by Liu et al. (2019), increasing the so-called ‘reduction factor’ - which applies dropout to the autoregressive frames - will help the model to learn to rely more on the attention network than the teacher forcing inputs, but possibly at the risk of compromising the synthesis quality improvements that come as a result of using an autoregressive system. FastSpeech2, and similar systems like FastPitch, present an alternative to Tacotron2-type CAR systems with similar listening test results and with less sensitivity to hyperparameter tuning and misconfiguration. Instead of modelling duration through an attention mechanism, phone durations are first extracted from a forced-alignment step in the preprocessing stage. In the predecessor ‘FastSpeech’, the attention weights from a Tacotron2 system were used to provide phone durations and the main benefits of the model were argued to be its speed, controllability and less error-prone inference (Ren et al., 2019). However, from a data requirements perspective, this is a Catch-22, as there might not be a sufficient amount of data to train an initial Tacotron2 model. Luckily, trainable forced alignment software like the Montreal Forced Aligner used in the implementation of FastSpeech2 removes the lingering dependency on Tacotron2 and allows for accurate text/audio alignment with minimal data.

5.2.2 Experimental Set-Up

To investigate the effects of differing amounts of data on the attention network, I trained five Tacotron2 models with the ‘LJ’ speech corpus (Ito & Johnson, 2017). I used the reference implementation with the default hyperparameters except with batch size adjusted to 32 as the default batch size of 64 exceeded memory available on my GPU. I artificially constrained the data for the five models such that the first model had only the first hour of data from the corpus, the second model had that same first hour and another two hours (3 total), etcetera, so that the five models contained 1, 3, 5, 10, 24 (full) hours respectively. The models were trained for 100k steps and, as seen in Figure 5.2, at 5 hours of data, the attention mechanism does not learn properly resulting in degenerate outputs.



(a) 5 Hr LJ Corpus Subset (b) 10 Hr LJ Corpus Subset (c) Full LJ Corpus

Figure 5.2: Visualization of Tacotron2 Attention Network Weights extracted after 100k steps. The weights of the attention network should be diagonal and monotonic as seen in subfigures b and c. Subfigure a shows that the network trained on a 5 hour subset of the LJ corpus results in a degenerate attention network.

For comparison with Tacotron2, I trained seven models using FastSpeech2 with 15 minutes, 30 minutes, 1 hour, 3 hours, 5 hours, 10 hours, and 24 (full) hours of data from the LJ corpus using the baseline model described in §5.1.1.

I conducted a short listening test to compare the two Tacotron2 models that trained properly (10h, full) with the seven FastSpeech2 models. The test recruited 30 participants through Prolific and was designed to take 10-15 minutes. Participants were asked to use headphones and were presented with four separate MUSHRA questions where they were asked to rank the 9 voices + reference.

5.2.3 Results

While it only took 30 minutes to recruit 30 participants using Prolific, the quality of responses was quite varied. I rejected two outright as they seemingly did not listen to the stimuli and left the same rankings for every voice. Even still, there was a lot of variation in responses from the remaining participants as illustrated below in the box plot in Figure 5.3.

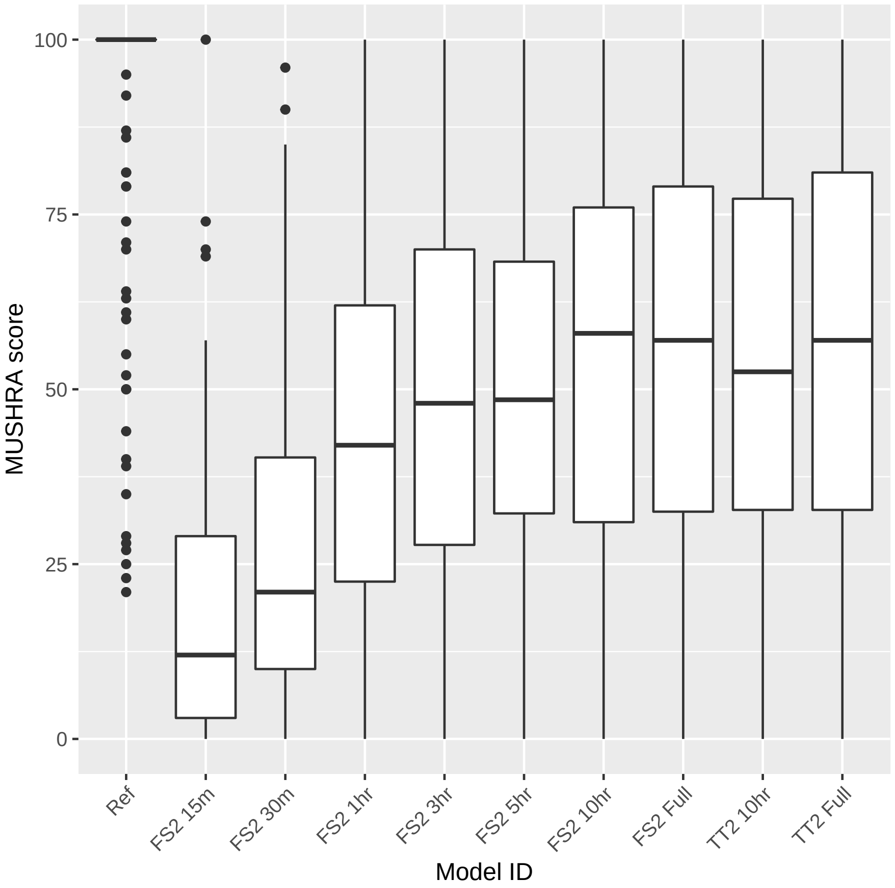


Figure 5.3: Box plot of survey data from MUSHRA questions comparing Tacotron2 and FastSpeech2 models with constrained amounts of training data

To test for significance, I performed Bonferroni-corrected Wilcoxon signed rank tests between each pair of voices. Formally, the Wilcoxon signed rank test makes the assumption that data is centered symmetrically around the median and makes the null hypothesis $H_0 = V_i = V_j$ and alternative hypothesis $H_1 = V_i \neq V_j$ for each pair of results V_i & V_j. The results of the pairwise test are summarized in the heat map of their p-values below in Figure 5.4.

In the results from the pairwise analysis, we can see that the distribution of MUSHRA results for the reference is significantly different (and higher) from the other voices. Similarly the results for the FastSpeech2 voices made with 15m and 30m of data are significantly different (and lower) from the other voices. The results from the remaining voices, while showing consistent improvements as more data is added in the raw MUSHRA scores seen in the box plot in Figure 5.3, are not significantly different from each other. This is a relevant and important finding for low resource speech synthesis because it shows that the results of a listening test for a

FastSpeech2 voice built with 3 hours of data are not significantly different from a Tacotron2 voice built with 24 hours of data. Similarly, the results of the listening test for a FastSpeech2 voice built with 1 hour of data are not significantly different from a Tacotron2 voice built with 10 hours of data. Additionally, while all the FastSpeech2 voices were intelligible, all Tacotron2 models trained with 5 hours or less data produced unintelligible speech.

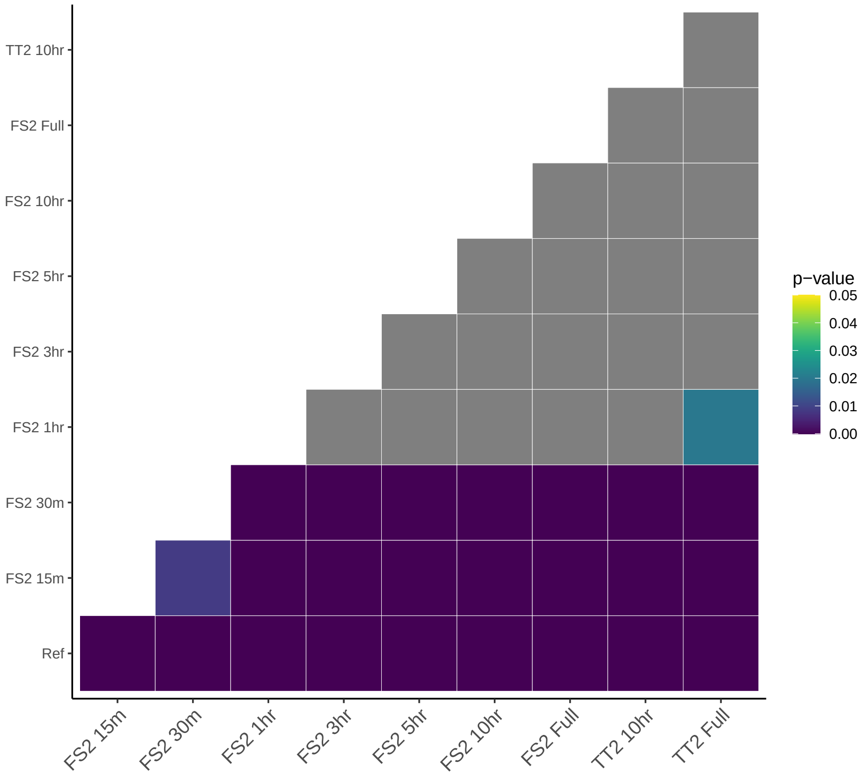


Figure 5.4: Pairwise Bonferroni-corrected Wilcoxon signed rank tests between each pair of voices. Cells correspond to the significance of the result of the pairwise test of model on the y-axis and model on the x-axis. Darker cells show stronger significance; grey cells did not show a significant difference in listening test results.

5.3 Transfer Learning

5.3.1 Background & Motivation

The findings from §5.2 showed that intelligible systems of reasonable quality can be created with very little data using FastSpeech2, which seems to be a largely overlooked fact as many low resource speech synthesis papers continue to adopt Tacotron2 as a baseline architecture. Most of the approaches for low-resource speech synthesis using Tacotron2 employ a method of training the network called ‘transfer learning’ (Tu et al., 2019; Wells & Richmond, 2021). Transfer learning is a term that refers to the process of training a neural network on a particular task and then finetuning the network on a separate but related task. This experiment sets out to determine whether transfer learning approaches can improve FastSpeech2 models as well. Preliminary results suggest that FastSpeech2 models trained from cold-starts perform better than their fine-tuned counterparts.

5.3.1.1 Transfer learning preliminaries: normalizing the input space

In transfer learning pipelines, it is crucially important that the network dimensions and number of parameters are the same between pretraining and finetuning tasks. In the standard implementations of neural speech synthesis systems like Tacotron2 or FastSpeech2, the input text is embedded as a sequence of one-hot vectors the size of the phone or character vocabulary. This means that the dimensions of the input to these systems are language-specific. As described in §2.3, the languages discussed in this dissertation have significant variability in the number and type of input characters, causing a problem for transfer learning between languages due to this input space mismatch. Tu et al. (2019) discuss methods addressing this difficulty including a ‘separate symbol space’ method which simply finetunes the source embeddings with the target symbols (note this would not work at all if the target space is larger than the source space), and a ‘unified symbol space’ method where the symbols from both source and target languages are combined and the source model is trained with that unified symbol space. The third method discussed develops a ‘Phonetic Transformation Network’ that learns a mapping between source and target inputs.

Further work by Wells & Richmond (2021) shows another approach which is most philosophically similar to the ‘unified symbol space’ method in Tu et al. (2019), but instead of using one-hot phone or character embeddings, the symbol space is unified by first converting each symbol in the text input sequence into a vector based on phonological features (Chomsky & Halle, 1991). Wells & Richmond (2021) use Tacotron2 and adapt their features slightly from Chomsky & Halle (1991)’s original feature set to show that modest improvements to the system can be found by using phonological features as inputs instead of one-hot embeddings, and that cross-lingual models can be finetuned with as little as 15 minutes of training data. In order to accommodate the change of inputs, their Tacotron2 system architecture is changed slightly to add a single fully-connected layer between the input phonological feature vectors and the first hidden layer of the rest of the network.

5.3.2 Experimental Set-Up

In order to determine the effectiveness of using transfer learning to decrease data requirements, I follow Wells & Richmond (2021) in training my models using phonological feature input vectors. However, instead of custom-created mappings from characters to features, I use the off-the-shelf PanPhon (Mortensen et al., 2016) library for converting IPA symbols to phonological feature vectors. In order to obtain the initial IPA symbol representation, I also created and published grapheme-to-phoneme mappings for each language using the *g2p* Python library. In addition to the 24 features used by PanPhon, I added 5 features for punctuation following Wells & Richmond (2021), as well as 7 features for representing tone following Wang (1967), as Kanien'kéha vowels can carry tone. See Table §A.3 for a sample of these vector representations. I also added the fully connected layer following Wells & Richmond (2021), as described in §5.1.1.

There are two experiments related to transfer learning that were developed. All models for both experiments are summarized in table 5.1.

5.3.2.1 Input Space Experiment

The first experiment is isolated to the difference in input embedding space. In order to test the effect of using phonological feature inputs instead of one-hot embeddings on varying levels of data, sixteen FastSpeech2 models were compared; eight using one-hot embeddings and eight using phonological feature embeddings, each with artificially constrained amounts of data ranging from 15 minutes to 3 hours as available for Gitksan, Kanien'kéha, and SENĆOŦEN, and all trained from cold starts for 200,000 steps.

5.3.2.2 Transfer Learning Experiment

The second experiment compares the effect of transfer learning compared with cold-start training. In order to test this, I trained another eight phonological feature-based models with the same data as described in the previous experiment, except instead of training from cold-starts, these models were initialized with the parameter weights from a pre-trained multilingual, multispeaker model after 300,000 steps and trained for an additional 25,000 steps. While the reference implementation of FastSpeech2

Model Name	Data	Amount	Inputs	Start
Str15mPhone	SENĆOŦEN (§3.3)	15m	1-hot	cold
StrFullPhone	SENĆOŦEN (§3.3)	full (26 min)	1-hot	cold
Git15mPhone	Gitksan (§3.2)	15m	1-hot	cold
GitFullPhone	Gitksan (§3.2)	full (35 min)	1-hot	cold
Moh15mPhone	Kanien'kéha (§3.1)	15m	1-hot	cold
Moh30mPhone	Kanien'kéha (§3.1)	30m	1-hot	cold
Moh1hrPhone	Kanien'kéha (§3.1)	1hr	1-hot	cold
Moh3hrPhone	Kanien'kéha (§3.1)	3hr	1-hot	cold
Str15mPF	SENĆOŦEN (§3.3)	15m	PF	cold
StrFullPF	SENĆOŦEN (§3.3)	full (26 min)	PF	cold
Git15mPF	Gitksan (§3.2)	15m	PF	cold
GitFullPF	Gitksan (§3.2)	full (35 min)	PF	cold
Moh15mPF	Kanien'kéha (§3.1)	15m	PF	cold
Moh30mPF	Kanien'kéha (§3.1)	30m	PF	cold
Moh1hrPF	Kanien'kéha (§3.1)	1hr	PF	cold
Moh3hrPF	Kanien'kéha (§3.1)	3hr	PF	cold
Str15mWarmPF	SENĆOŦEN (§3.3)	15m	PF	warm
StrFullWarmPF	SENĆOŦEN (§3.3)	full (26 min)	PF	warm
Git15mWarmPF	Gitksan (§3.2)	15m	PF	warm
Git30mWarmPF	Gitksan (§3.2)	full (35 min)	PF	warm
Moh15mWarmPF	Kanien'kéha (§3.1)	15m	PF	warm
Moh30mWarmPF	Kanien'kéha (§3.1)	30m	PF	warm
Moh1hrWarmPF	Kanien'kéha (§3.1)	1hr	PF	warm
Moh3hrWarmPF	Kanien'kéha (§3.1)	3hr	PF	warm

Table 5.1: Summary of Transfer Learning Experiment Models

used in the experiment included support for multi-speaker inputs by summing one-hot speaker embeddings with the output of the encoder, there was not support for multi-lingual inputs. As such I added another hyperparameter to the model configuration to allow for inputs to be labelled for language and for a one-hot language embedding to be summed to the output of the encoder as seen in §5.1.1 in Figure 5.1.

I trained my multi-lingual, multi-speaker model using data from the VCTK corpus[3], the Kawennón:nis recordings (§3.1.1), and the augmented synthetic data from the concatenative Gitksan system (§5.4).

[3] I trained an initial model with the LibriTTS Clean-100 corpus but found the quality to be poor, likely due to the acoustic variability of the corpus.

5.3.2.3 Evaluation Strategy

Objective evaluation is the main evaluation technique used for these experiments. However, an A/B listening test with 20 questions was provided to Kanien’kéha speakers and learners with ten questions to compare the best one-hot model (*Moh3hrPhone*) with the best phonological feature model (*Moh3hrPF*), as well as ten questions to compare the best cold-start phonological feature model (*Moh3hrPF*) with the best multilingual fine-tuned model (*Moh3hrWarmPF*).

5.3.3 Results

Due to the large number of models present in this experiment, I present summarized findings through statistical models. For full results from the objective evaluations please refer to A.4 in the Appendix.

The general and unsurprising trend across all experiments is that models with more data, tend to perform consistently better with more data as seen in Fig 5.5.

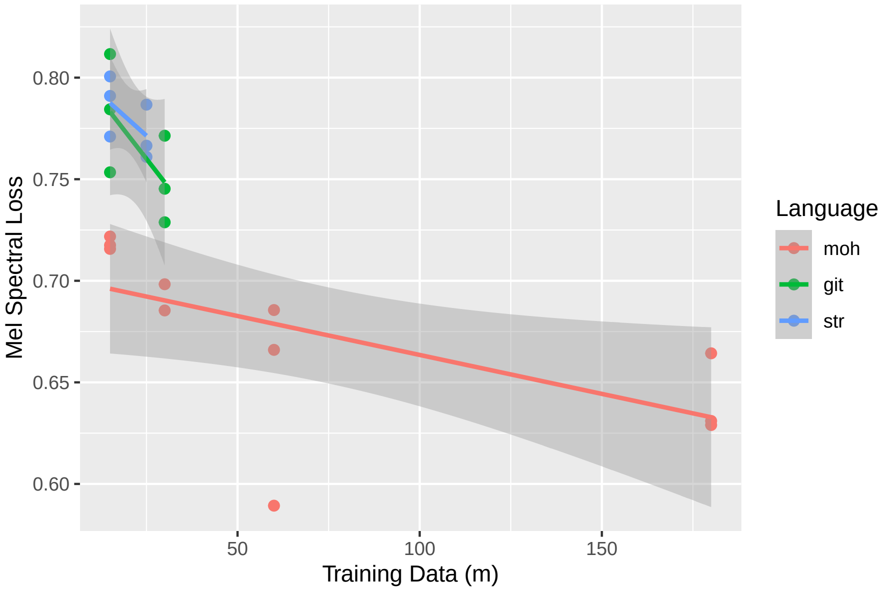
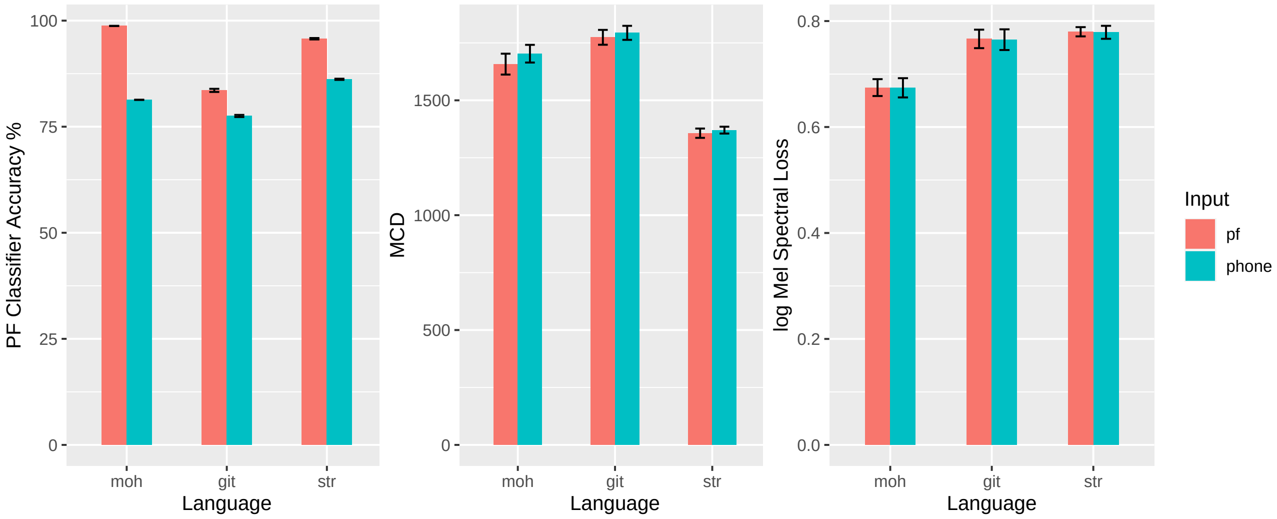


Figure 5.5: Plot of log Mel spectral feature loss (LMSFL) for differing levels of data between Kanien’kéha (moh), Gitksan (git), and SENĆOŦEN (str). The distribution of these results is not exactly linear and so the figure should only be understood as showing the rough trend of decreasing LMSFL as training data increases. Notably, all Kanien’kéha models perform better than all Gitksan and SENĆOŦEN models.

Results relevant to the input embedding space experiment (Fig 5.6) show that models trained with one-hot inputs had similar results to those trained with phonological feature inputs across all metrics except the phonological feature classifier accuracy (5.6a). There also seems to be a small improvement in MCD for models using phonological features, but this gain is not significant.



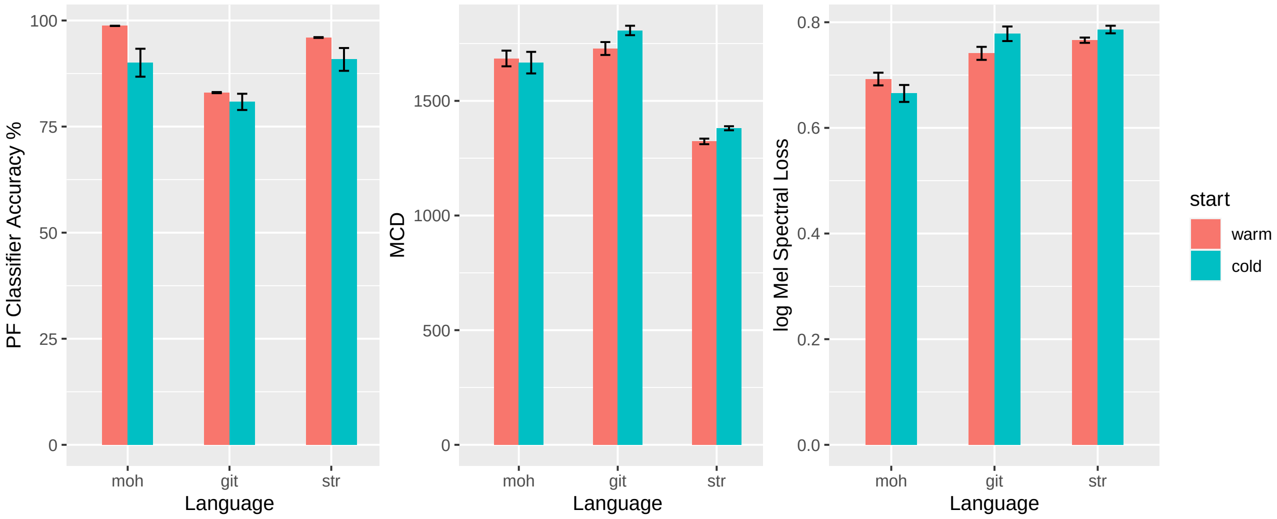
(a) PF Classifier Results (b) MCD Results (c) LMSFL Results

Figure 5.6: Bar Plot showing standard errors for evaluation metrics, split to show difference between Kanien’kéha (moh), Gitksan (git) and SENĆOŦEN (str) and whether the model was trained with one-hot or phonological feature vectors. Higher PF Classifier Accuracy results are better, and lower MCD and log Mel spectral feature loss (LMSFL) results are better.

In the listening test of A/B questions comparing phonological features and one-hots, 56.67% of responses preferred the models trained with phonological features, although the listening test was only able to recruit six participants. While more research with a larger group of participants in a listening test would be more conclusive, it would appear that there is not a meaningful difference between models trained from cold starts with one-hots compared with phonological features.

Objective results for the second experiment (see Fig 5.7) show that while Gitksan and SENĆOŦEN see minor improvements among models trained from a warm start (lower MCD, lower LMSFL, higher classifier accuracy), Kanien’kéha shows the opposite. The error bar for cold-start models is much higher in Figure 5.7a, but this is because the cold-start model combines phonological feature and 1-hot input models, whereas the warm start model was only trained with phonological features which performed much better as seen in Figure 5.6a.

Listening test results comparing the warm- and cold-start models showed a strong preference for cold start models; 81.67% of A/B question responses preferred the cold start model. Listening tests comparing these models were only done for Kanien’kéha.



(a) PF Classifier Results (b) MCD Results (c) LMSFL Results

Figure 5.7: Bar Plot showing standard errors for evaluation metrics, split to show difference between Kanien'kéha (moh), Gitksan (git) and SENĆOŦEN (str) and whether the model was finetuned on a pre-trained model (warm) or trained from a cold start. Higher PF Classifier Accuracy results are better, and lower MCD and LMSFL results are better.

While further testing would be needed to replicate this finding across other languages, it would appear that there is a strong speaker preference for cold-start trained models. This is an interesting finding because it questions the common practise of using transfer learning for low-resource speech synthesis. For conditional autoregressive models like Tacotron2, transfer learning might simply be required in order to ensure proper training of the attention network. However, if the difficulty of training the attention network is removed, as with FastSpeech2, it's not clear that transfer learning is an advisable strategy. It is also possible that the pretrained model was not adequately large however, so further research would need to be done to determine whether a more robust multilingual/multispeaker model could provide overall improvements over a cold-start trained model.

5.4 Data Augmentation

5.4.1 Background & Motivation

The cold-start trained FastSpeech2 models from §5.3 were sounding quite good for Kanien'kéha, but for languages like Gitksan and SENĆOŦEN which have more sounds and less data, the models had noticeable deficiencies. What was immediately apparent to me upon hearing the initial models was that there was a metallic noise that seemed to occur in some of the utterances, particularly during sequences of voiceless fricatives. Comparing synthesized samples with ‘copy-synthesized’ samples where the log Mel spectral features of the recorded audio are passed through the vocoder ruled out the issue being with the vocoder. And, as described in 2.3.2, while long sequences of voiceless fricatives are cross-linguistically rare, they are common in Gitksan and within the Pacific Northwest Sprachbund. I decided to observe the relative log Mel spectral error among voiceless fricatives in the synthetic Gitksan speech with other segments and indeed found a difference as seen in Figure 5.8.

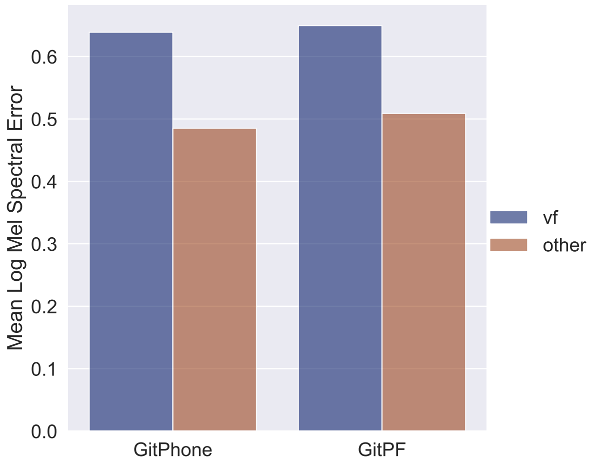


Figure 5.8: Comparison of mean log Mel spectral error between voiceless fricative (vf) segments and ‘other’ segments for both one-hot (GitPhone) and phonological feature (GitPF) based models

A traditional approach to improving this might involve transfer learning. However, because sequences of voiceless fricatives are cross linguistically rare, it’s not clear which language I would use to pretrain the model. Additionally, my previous experiment on transfer learning (§5.3) did not show any substantial improvements to the Gitksan models when finetuned on a multilingual, multi-speaker model.

Instead of transfer learning, I set out to try a data augmentation approach. Data augmentation is the practice of increasing the amount of data available to a machine learning system by creating synthetic or modified data. Data augmentation for speech synthesis has little discussion in the literature. Hwang et al. (2020) discuss a technique where they augment 5 hours of training data with an extra 174 hours of synthetic speech from an auto-regressive TTS system (Tacotron2) to improve the quality of FastSpeech2 outputs. More commonly, synthetic speech is used to augment data for ASR systems (Laptev et al., 2020; Bagchi et al., 2020; Jia et al., 2019).

The approach I take is a hybrid concatenative/neural approach. As described in §2.1, concatenative and neural speech synthesis paradigms are usually quite distinct, although some hybrid techniques have also been developed. Commonly however, as with Qian et al. (2013), these techniques involve using a neural system to generate the parametric targets of a synthetic utterance which are then used by the concatenative system to determine the appropriate units for concatenation. Instead, my approach is to use a concatenative system to generate targeted, augmented data for the neural system.

5.4.2 Experimental Set-Up

To augment the limited available data in a way that targeted voiceless fricatives, a concatenative model for Gitksan was created using the method described in §5.1.2 with the original Gitksan data. Then, a lexicon was created in Festival with a unique 'word' for each possible permutation of of voiceless fricatives with lengths two through six (the length of the set of voiceless fricatives), resulting in a vocabulary of entries like 'sx^{w}ɬx', 'x^{w}sɬ' or 'χxsɬh' as examples. This set was then filtered to exclude phontactically impermissible sequences, resulting in 495 additional synthetic utterances available to augment the original Gitksan data. My approach here stems from two key insights; the first is that the errors in the neural synthesis appear to be localized to voiceless fricatives. The second insight is that while concatenative systems produce errors for a variety of reasons, many are due to F0 mismatches in voiced sounds as seen in Figure 5.9. By contrast, the concatenation of voiceless fricatives tends to incur fewer errors as there is no periodicity in the signal that could result in a perceptible discontinuity.

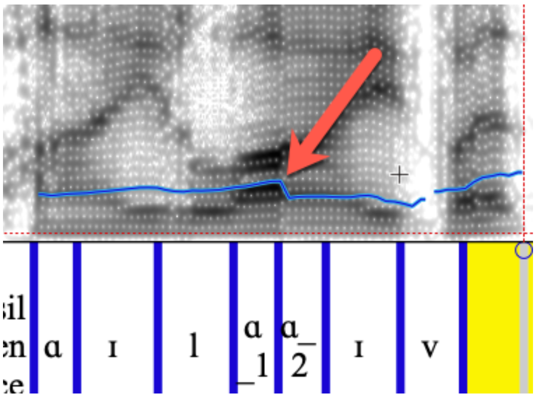


Figure 5.9: F0 discontinuity at join point between two voiced segments

To test the effect of this augmented data on the baseline model, I designed an experiment by creating four models using the neural baseline FastSpeech2 model summarized in Table 5.2.

Model Name	Data	Inputs
GitPhone	Gitksan (§3.2)	1-hot
GitAugPhone	Gitksan (§3.2)	1-hot
GitPF	Gitksan (§3.2)	PF
GitAugPF	Gitksan (§3.2)	PF

Table 5.2: Summary of Data Augmentation Models

I then evaluate the models according to their objective evaluation results and a listening test. The listening test included 40 questions, including 10 AB-pair tests comparing the augmented models with the un-augmented models. In addition, 5 MUSHRA questions and 25 MOS questions for randomly chosen sentences were also presented to 12 participants. Participants were recruited by emailing a link directly to speakers, through Facebook, and by sending a link to the Gitksan Research Lab at the University of British Columbia. Note that no participants reported to be first-language speakers, although all had some familiarity with the language.

5.4.3 Results

My personal impression is that the models made with augmented data were noticeably better than the baseline, particularly when trained with phonological feature inputs, and among utterances containing sequences of voiceless fricatives. However, the results from this experiment are somewhat contradictory, and my opinion was not borne out in all results. For example, as seen in Table 5.3, while the log Mel spectral feature loss dropped for both augmented models, the Mel cepstral distortion increased. I believe this could be due to the fact that the loss calculated by the 'pitch' (F0) variance predictor for the augmented models is drastically lower, which would be expected to be correlated with a drop of the log Mel spectral feature loss, but not the Mel cepstral distortion as the multiples of the fundamental frequency would more strongly affect the Mel spectral loss than the MCD calculated from decorrelated MFCCs. It is unclear why the augmented models affect the F0 loss so drastically.

The phonological feature classifier does not seem to support my impression that the augmented models improve either; the only real noticeable change is that models trained with phonological feature inputs seem to have a higher overall classifier accuracy than the models trained with one-hots.

	Mel loss	MCD	'Pitch' Loss	Hamming-0	Hamming-3	Classifier Acc.
GitPhone	0.746	**1763.21**	6.404	21.33%	63.77%	77.78%
GitAugPhone	**0.739**	1949.85	0.255	20.55%	63.86%	**76.81%**
GitPF	0.771	1784.88	6.475	20.68%	**63.32%**	84.55%
GitAugPF	0.744	1949.51	**0.251**	**20.54%**	64.44%	84.50%

Table 5.3: Summary of Objective Evaluation

To try to isolate the change to voiceless fricative segments, I calculated the MCD for both voiceless fricative segments and all other segments. As seen in Figure 5.10, the improvement seems to mostly occur in segments other than voiceless fricatives, and voiceless fricative segments actually incur a higher MCD in the augmented models.

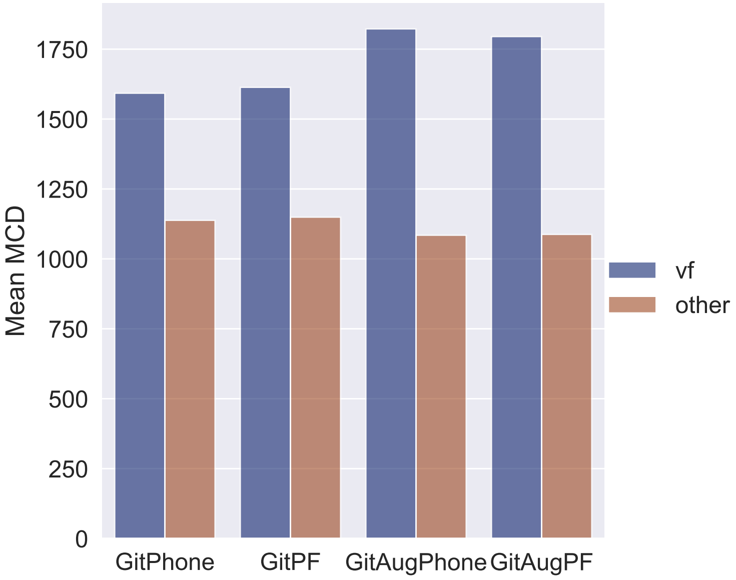


Figure 5.10: Mean MCD across batches separated by model as well as voiceless fricative 'vf' segments and 'other' segments.

The results from the listening test were varied as well. The MUSHRA results (Figure 5.11) did not show much of a difference between the voices, however, I received a comment from one of the participants that my reference pronunciation of 'haguxwsg̱alt'amdinsxw' (camera) had stress incorrectly assigned to the final syllable when it should have been on the penultimate syllable. The synthetic versions had the stress in the correct spot, and so the participant ranked one of those samples as the reference. This could explain some of the sub-100 scores for the reference samples as seen in Figure 5.11, and might suggest that participants were weighting the difference in voiceless fricative distortion lower than other differences in the samples.

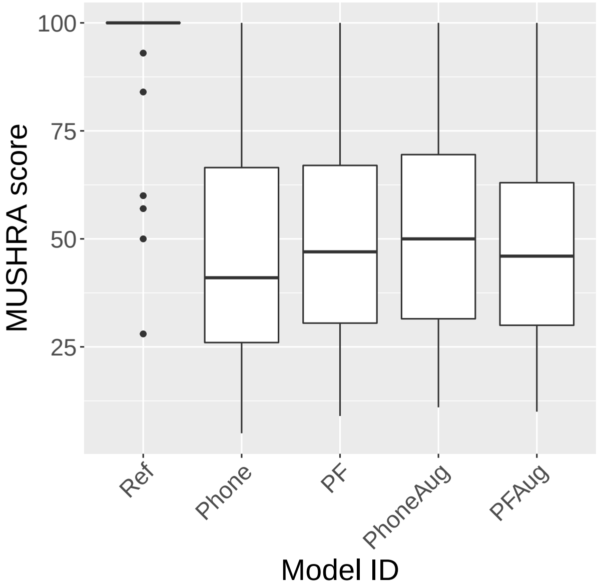
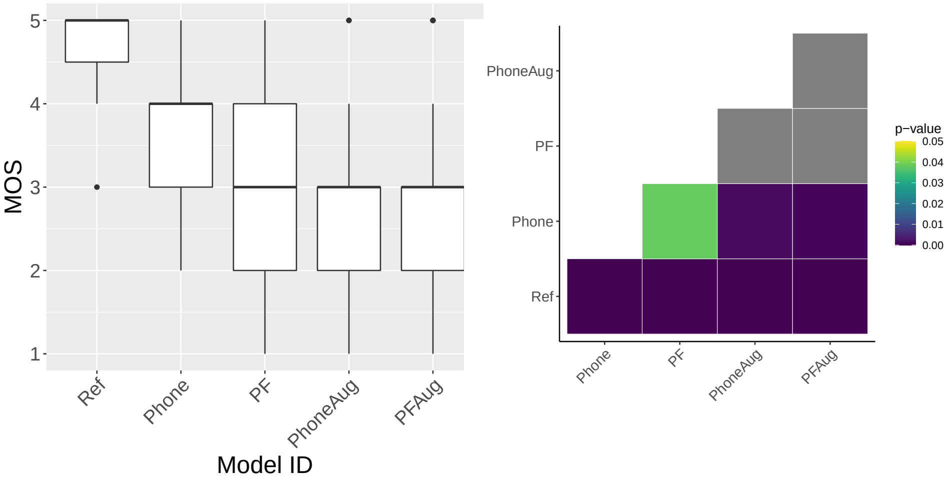


Figure 5.11: Results from the MUSHRA portion of the Gitksan listening test. A pairwise Bonferroni-corrected Wilcoxon signed rank test showed that the only significant difference between these distributions was between the reference (Ref) and other voices.

The results from the Mean Opinion Score questions also seemed to refute my personal preference for the augmented model, with the one-hot phone-based model performing significantly better than the other models as seen in Figure 5.12.



(a) Mean Opinion Score Results

(b) Bonferroni-corrected pairwise Wilcoxon signed rank test results

Figure 5.12: Results from the Mean Opinion Score Portion of the listening test

The results from the A/B tests, however, told another story where 100% of question responses preferred the augmented ‘GitAugPF’ samples over the ‘GitPF’ samples, which certainly seems to suggest a perceptible difference between the models, with a preference for the augmented version. By comparison only 62.5% of A/B responses preferred the augmented ‘GitAugPhone’ model over the ‘GitPhone’ model.

Further research and evaluation would be needed to verify the reliability of these findings and to observe whether this approach could be extended to target unique phonological patterns in other low-resource languages.

Chapter 6

Conclusion

In this dissertation, I have motivated and described the development of speech synthesis systems for Gitksan, Kanien'kéha, and SENĆOŦEN. I recorded, processed, and aligned corpora for these languages and developed and released grapheme-to-phoneme mappings for all three. A baseline FastSpeech2 architecture was adapted to include three new modules to support phonological feature inputs, multilingual training, and zero-shot multi-speaker training. The encoder, decoder, and variance adaptor modules were also refactored to reduce the number of parameters following Luo et al. (2021). I then built 42 speech synthesizers to run three main experiments, which were evaluated by three separate objective metrics and three separate listening tests.

Results showed that FastSpeech2 is able to produce comparable voices to Tacotron2 with far less data, Kanien'kéha listening test participants had a preference for the neural baseline over the concatenative baseline, as well as a mild preference for phonological feature based models and a strong preference for models trained from cold starts instead of fine-tuned on a pretrained multilingual, multi-speaker model. Gitksan listening test results were mixed, but seem to indicate that participants preferred models trained with augmented data, although objective evaluation metrics did not corroborate that finding.

In retrospect, I think the single most significant challenge of this dissertation has been determining a reliable and convincing method of evaluation. Between the lack of eligible listening test participants, and the unreliability of objective metrics in supporting the results of listening tests, further research on low-resource speech synthesis should prioritize evaluation. As with my first experiment, sometimes the difficulty of evaluation can be circumvented by creating experiments on a high resource language like English but with models trained on artificially constrained amounts of data. While this is enticing, I fear that there are other properties of low resource languages that could get overlooked.

At the outset of this dissertation, I did not know if neural speech synthesis models would produce intelligible speech for these three languages. I was particularly doubtful for Gitksan and SENĆOŦEN which both only had roughly 30 minutes of data. However, by carefully examining where the difficulty in training these neural systems originates, I feel this dissertation has concluded some valuable findings with respect to speech synthesis for Indigenous languages. Most important is the finding that even 15 minutes of data can produce intelligible speech given the proper neural architecture. This is exciting, because it opens the door to speech synthesis research for a large number of communities that may have otherwise discounted the possibility for their languages. The task of determining how to meaningfully integrate speech synthesis into language revitalization efforts is a goal of future work.

Appendix A

Appendix

A.1 Speaker Adaptation

The original FastSpeech2 description did not include multi-speaker support, but a multi-speaker system using one-hot speaker embeddings was included in the reference implementation. One-hot speaker embeddings are sufficient for adapting synthetic speech to one of the speakers seen in training, but the approach does not allow for speech to be adapted in a zero-shot or few-shot way to new speakers. As described in §3.1.2, sometimes it might not be culturally or politically appropriate to have the particular voice of a speaker used in a specific context. For example, if a speaker has passed away or if a speaker's voice is used to pronounce a dialect other than their own. In these situations, speaker adaptation is a goal for speech synthesis for language revitalization.

I adapted the FastSpeech2 preprocessing pipeline to also calculate a 512 dimensional vector for each speaker using a pre-trained Deep Speaker model (Li et al., 2017). For each speaker, each utterance was mapped to a hypersphere using Deep Speaker and the mean of the resulting vectors was stored. Then, in place of summing the one-hot speaker embedding to the output of the encoder, a fully-connected linear layer was added to project the stored speaker vector to a hidden dimension equal to the hidden dimension of the output of the encoder (256) and then summed together with the output of the encoder.

This seemed to work, although my impression was that the zero-shot models did not sound like the voices their speaker-vectors came from; rather they sounded mostly like the original speakers with some perceptible changes in pitch. Further research would be needed to investigate speaker adaptation for low resource languages.

A.2 Compute, Accessibility, & Environmental Impact

For reasons of environmental impact and accessibility, reducing the amount of computation required for both training and inference is important for any neural speech synthesis system, particularly so for Indigenous languages.

A.2.1 Accessibility, Training & Inference Speed

Reducing the number of parameters in the model should translate to increased efficiency of the model, and might make the model less prone to overfitting when training on limited amounts of data. Following Luo et al. (2021), I removed the energy variance adaptor and refactored the original convolutional layers in the encoder, decoder and remaining variance predictors to depthwise separable convolutional layers (a depthwise convolution followed by a pointwise convolution). I also changed the number of layers in the decoder from 6 to 4 and changed the size of the kernel in the encoder and decoder convolutional layers to match the LightSpeech model described in Luo et al. (2021). These changes reduced the number of parameters in the model from 35,076,161 to 11,591,233 without noticeable change in voice quality, in addition to reducing the size of the stored model from 417.06MB to 135.24MB and significantly improving inference and train times as summarized in Table A.1.

In addition to describing the environmental cost of popular NLP models, Strubell et al. (2019) also make a compelling argument for equitable access to compute. Put another way, systems which require less compute, are more accessible. While language revitalization efforts are *mostly* encouraging about integrating new technologies into curriculum, there is a growing awareness of the potential harms.

Beyond assessing the benefits and risks of introducing a new technology into language revitalization efforts, communities are concerned with the way the technology is researched and developed; as this process has the ability to empower or disempower language communities in equal measure (Alia, 2009; Brinklow et al., 2019). The current model for developing speech synthesis systems is not very equitable - models need to be run on GPUs by people with specialized training. For Indigenous communities to create speech synthesis tools for their languages, they should not need be required to hand over their language data to a large government or corporate organization. Despite the poor quality of the results in the transfer learning experiment (§5.3), a pre-training, fine-tuning pipeline is attractive for this reason. It means that communities could fine tune their own models on a laptop if a multilingual/multi-speaker model were pre-trained on GPUs at a larger institution. Reducing the computational requirements for training and inference of these models could help ensure language communities have greater control over the process of the development of these systems, less dependence on governmental organizations or corporations, and more sovereignty over their data (Keegan, 2019).

In Table A.1, I compare the training and inference time of the off-the-shelf FastSpeech2 system, as well as the adapted version described above.

		FastSpeech2	Adapted System
Training	GPU	90.52ms (σ 3.31)	60.04ms (σ 1.70)
	CPU	7561.50ms (σ 263.55)	2720.88ms (σ 92.99)
Inference	GPU	12.00ms (σ 0.30)	10.23ms (σ 0.78)
	CPU	138.73ms (σ 3.94)	59.50ms (σ 1.85)

Table A.1: Mean observed training and inference times for a single forward pass of baseline FastSpeech2 and adapted models

Results were timed by running the model for 300 repetitions and taking the mean. The GPU (Tesla V100-SXM2 16GB) was warmed up for 10 additional repetitions before timing started, and PyTorch's built-in GPU synchronization method was used to synchronize timing (which occurs on the CPU) with the training or inference running on the GPU. CPU tests were performed on an Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz with 4 cores and 16GB memory reserved.

A.2.2 CO2 Consumption

Strubell et al. (2019) argue that NLP researchers should have a responsibility to disclose the environmental footprint of their research, in order for the community to effectively evaluate any gains and to allow for a more equitable and reproducible field.

The Canadian General Purpose Science Cluster (GPSC) in Dorval, Quebec is where all experiments requiring a GPU were performed. Experiments were all run on single Tesla V100-SXM2 16GB GPUs. Strubell et al. (2019) provide the following equation for estimating CO_2 production:

$$p_t = \frac{1.58t(p_c + p_r + (g * p_g))}{1000} \tag{A.1}$$

where t is time, p_t is total power for training, p_c is average draw of CPU sockets, p_r is average DRAM memory draw, g is the number of GPUs used in training and p_g is the average draw from GPUs.

In my case, I estimate t to be equal to 1,541.98 after summing the time for experiments based on their log files, p_c is 75 watts, p_r is 6 watts, g is 1, and p_g is 250 watts, and the equation for grams of CO2 consumption is $CO_2 = 34.5p_t$ as the average carbon footprint of electricity distributed in Quebec is estimated at 34.5g CO2eq/kWh (Levasseur et al., 2021). This results in a total equivalent carbon consumption of 27,821.65 grams, roughly equivalent to driving a single passenger gas-powered vehicle for 110 kilometres according to the average rate of 404 grams/mile (EPA, 2019).

This is a comparatively low C02 consumption for over 1500 GPU hours, largely due to the low CO2/kWh output of Quebec electricity when compared with the 2019 USA average of 400g CO2eq/kWh (EPA, 2019). However, CO2 equivalents are just a proxy for environmental impact and should not be understood to comprehensively account for social and environmental impact. Hydro-electric dam projects in Quebec, like the ones powering the GPSC have a sordid and complex history in the province. Innu Nation Grand Chief Mary Ann Nui spoke to this when she commented that “over the past 50 years, vast areas of our ancestral lands were destroyed by the Churchill Falls hydroelectric project, people lost their land, their livelihoods, their travel routes, and their personal belongings when the area where the project is located was flooded. Our ancestral burial sites are under water, our way of life was disrupted forever. Innu of Labrador weren’t informed or consulted about that project” (Innu-Atikamekw-Anishnabeg Coalition, 2020).

A.3 Tips

Here is an informal list of tips for anyone attempting to reproduce the work described in this dissertation.

- Speech Recorder 1.0 (12) will erase any files in a folder that you export to without warning, so make sure to always export your files to an empty folder.

- When creating an ad-hoc character set be sure to not use digits in isolation, they are not handled properly by Festival without adjusting the source. In addition, HTK does not allow phones to begin with digits[1], and case-sensitive characters should also be avoided.

- One complication of using phonological features is that alignment needs to be done from scratch, even if alignments for a given corpus have been released publicly. I had access to VCTK alignments but had to align the data myself because the available alignments used standard ARPABET representations which treat diphthongs like EY /e͡ɪ/ as a single unit; however this sequence produces two feature vectors. Given the marginal improvement for phonological features found in §5.3.3 it is not clear that using features are worth this extra step.

A.4 Extra Implementation Details

There are some low-level implementation details that I felt were important to include, but might be seen as too low-level for the purposes of this dissertation. I therefore include them in this section of the Appendix.

A.4.1 Database Selection for Concatenative TTS

For a concatenative system, minimally each phone in the target domain must be present in the database in order to avoid missing phone errors. However, in order to capture context effects from adjacent phones, it is best to ensure wider coverage than a single phone or diphone. In order to select the fewest and best candidates from the nearly 247,450 unique forms from Kawennón:nis, a grapheme-to-phoneme engine was created for the language and used to turn each verb form into a sequence of phones. The recordings were then selected by recursively accumulating the utterance (u) that maximized the formula in A.2. The formula[2] is designed to score candidate utterances by the number of unique phoneme trigrams weighted by their token frequency in the target domain. When a candidate is selected, the phone tri-

[1] `https://github.com/prosodylab/Prosodylab-Aligner/issues/10`

[2] I could not find this written anywhere explicitly, although it is discussed in Simon King (2021) and by Black & Lenzo (2000)

grams that are found in the selected utterance are removed from the list of target phoneme trigrams P.

$$\arg\max_{u} f(u) = \sum_{i=1}^{|u|} \begin{cases} \frac{1}{count(u_{i-2},...,u_i)}, & \text{if } (u_{i-2}, ..., u_i) \in P \\ 0, & \text{otherwise} \end{cases} \tag{A.2}$$

Running this algorithm on the output set of unique conjugations resulted a data set containing 852 recordings of single words from Kawennón:nis. The reason rare trigrams are weighted more heavily is because of the Zipfian distribution of phoneme trigrams; if rare trigrams are targeted, more common trigrams will be accumulated along the way.

A.4.2 Mel Cepstral Distortion

My implementation of MCD differs slightly from published descriptions as found in Kubichek (1993); Mashimo et al. (2001); Kominek et al. (2008). Firstly, these papers do not agree on how many MFCCs to use; I only take the first thirteen MFCCs to decorrelate the features from multiples of F0 and I omit the 0th coefficient to disregard energy following Kominek et al. (2008). I then calculate MCD in the same way as the Mel spectral error: by taking the mean over all frames in a given batch (with padded frames removed) and then reporting the mean across batches. Kominek et al. (2008) also scale their result by $\alpha = \frac{10\sqrt{2}}{\ln 10} = 6.14185$ for historical reasons, which I do not do. The resulting formula for my implementation of MCD can be described in A.3.

$$MCD(v, \hat{v}) = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{T}\sum_{t=1}^{T}\sqrt{\sum_{d=1}^{D}(v_d(t) - \hat{v}_d(t))^2} \tag{A.3}$$

where B is the number of batches, T is the number of frames in the batch (after padded frames are removed), D is the number of coefficients and $v_d(t)$ and $\hat{v}_d(t)$ are the d^{th} coefficients from the DCT-II as applied to the log Mel spectral features at frame t for the label voice v and synthetic voice $\hat{v}$ respectively.

	Input Dim	Output Dim	Context Size	Dilation	Activation	Type
Layer 1	80	512	5	1	ReLU	TDNN
Layer 2	512	512	3	2	ReLU	TDNN
Layer 3	512	512	3	3	ReLU	TDNN
Layer 4	512	512	1	1	ReLU	TDNN
Layer 5	512	1500	1	1	ReLU	TDNN
Layer 6	1500	512	N/A	N/A	ReLU	FC
Layer 7	512	72	N/A	N/A	None	FC

Table A.2: Model Parameters of Phonological Feature Classifier

A.4.3 Phonological Feature Classifier

A.4.3.1 Network Architecture Summary

A.4.3.2 Evaluation

During evaluation, padded frames are removed and the 72 dimensional real-valued outputs of the model are first rounded to 0 or 1, then decoded from their 2-bit encoding into phonological feature vectors.

For speech synthesis models trained with phonological features as inputs, evaluation using this classifier can be implemented straightforwardly by using the ground truth inputs as the labels for the classifier. For models trained with one-hot embeddings, the input text must have dipthongs removed from both the inputs as dipthongs will result in two feature vector values. Then, all frames in the output corresponding to the input diphthongs must be removed after which the input text can be converted to feature vectors and used as labels for the classifier.

Models created with FastSpeech2 predict an explicit duration for each phone, whereas extracting phone durations from Tacotron2 models would require a heuristic for interpreting the weights of the attention network to reconstruct the number of frames related to each phone. It would be similarly possible to reconstruct the timestamps for each phone from a concatenative system; however this method of evaluation was only applied to models built using FastSpeech2.

A.5 Glossary

CN - connective, DUAL - dualic, PPFV - past perfective, PURP - purposive, REP - repetitive, SREFL - semi-reflexive, VAL - valency.

	s	á	χ	;
syl	-1	1	-1	0
son	-1	1	-1	0
cons	1	-1	1	0
cont	1	1	1	0
delrel	-1	-1	-1	0
lat	-1	-1	-1	0
nas	-1	-1	-1	0
strid	0	0	0	0
voi	-1	1	-1	0
sg	-1	-1	-1	0
cg	-1	-1	-1	0
ant	1	0	-1	0
cor	1	-1	-1	0
distr	-1	0	0	0
lab	-1	-1	-1	0
hi	-1	-1	-1	0
lo	-1	1	-1	0
back	-1	-1	1	0
round	-1	-1	-1	0
velaric	-1	-1	-1	0
tense	0	1	0	0
long	-1	-1	-1	0
hitone	0	0	0	0
hireg	0	0	0	0
exclamation	0	0	0	0
question	0	0	0	0
big break	0	0	0	1
small break	0	0	0	0
quote mark	0	0	0	0
contour	0	-1	0	0
high	0	1	0	0
central	0	-1	0	0
mid	0	-1	0	0
rising	0	-1	0	0
falling	0	-1	0	0
convex	0	-1	0	0

Table A.3: Example multi-hot encodings for four separate characters. The first 24 features are obtained from the PanPhon library (Mortensen et al., 2016), the next 5 features are related to punctuation and the last 7 features are for tone following Wang (1967).

A.6 Results

In this section I include some raw results §5.3 summarized in Table A.4, and also report qualitative responses to whether participants would feel comfortable with the voices they heard being used to supplement tools like digital dictionaries. The question was not asked to English listening test participants.

Model Name	Spec. Error	MCD	Hamming-0	Hamming-3	Classifier Acc.
Str15mPhone	0.791	1385.17	74.14%	88.08%	86.03%
StrFullPhone	0.767	1355.37	76.59%	89.39%	86.31%
Str15mPF	0.801	1397.22	72.99%	88.50%	95.30%
StrFullPF	0.164	1384.10	74.57%	89.54%	95.59%
Str15mWarmPF	0.771	1335.31	77.10%	90.28%	95.89%
StrFullWarmPF	0.761	1310.98	77.43%	90.72%	96.08%
Git15mPhone	0.784	1824.31	19.97%	61.47%	77.28%
GitFullPhone	0.745	1763.21	21.33%	63.77%	77.78%
Git15mPF	0.812	1855.64	18.99%	62.07%	83.67%
GitFullPF	0.771	1784.88	20.68%	63.32%	84.55%
Git15mWarmPF	0.753	1756.37	19.35%	60.37%	83.14%
Git30mWarmPF	0.729	1700.15	18.57%	59.97%	82.87%
Moh15mPhone	0.716	1788.94	39.11%	40.60%	81.31%
Moh30mPhone	0.685	1730.73	39.17%	40.69%	81.36%
Moh1hrPhone	0.666	1686.05	39.21%	40.68%	81.32%
Moh3hrPhone	0.629	1606.23	39.17%	40.66%	81.34%
Moh15mPF	0.718	1785.03	90.72%	94.18%	98.76%
Moh30mPF	0.688	1739.77	90.75%	94.75%	98.86%
Moh1hrPF	0.589	1386.89	86.09%	93.38%	98.62%
Moh3hrPF	0.631	1609.58	91.19%	94.77%	98.81%
Moh15mWarmPF	0.722	1768.29	89.78%	94.40%	98.69%
Moh30mWarmPF	0.698	1705.03	89.79%	94.50%	98.66%
Moh1hrWarmPF	0.686	1658.64	90.65%	94.96%	98.79%
Moh3hrWarmPF	0.664	1607.45	91.17%	95.06%	98.78%

Table A.4: Summary of Objective Evaluation Results from Transfer Learning Experiments

Question:

"Would you be comfortable with any of the voices you heard being played online, say for a digital dictionary or verb conjugator if no other recording existed?"

Kanien'kéha responses:

- Yes.
- yes
- Yes
- Out of the two voices I hear, the first was clearer to understand
- Yes, voices sounds really good!
- yes

Gitksan responses:

- yes
- Yes, but the ones that have the most whistling or buzzing would be annoying.
- maybe?? I think for a talking dictionary people do want to hear original pronunciations, but it could be a useful interim solution or a way to do short phrases!
- Yes
- Yes.
- Assuming there is a single control for the last section of the survey/test, then some of the synthesised voices actually sound really good and I would be comfortable hearing those in an online dictionary where audio didn't exist for a particular word or phrase.
- yes
- The ones with higher ratings for sure, some of the lower ratings were just about the sound quality because that hampered hearing the speech quality. So I may have confounded the results with that, but point remains that it is always good to try to avoid poor audio recordings for online dictionaries

- Maybe/yes
- only ones rated fair or above fair
- Absolutely yes
- yes, as long as they were identified as synthesized

Bibliography

Alia, V. (2009). *The New Media Nation: Indigenous Peoples and Global Communication*. Berghahn Books, ned - new edition, 1 ed.
URL https://www.jstor.org/stable/j.ctt9qd5x7

Bagchi, D., Wotherspoon, S., Jiang, Z., & Muthukumar, P. (2020). Speech Synthesis as Augmentation for Low-Resource ASR.
URL http://arxiv.org/abs/2012.13004

Black, A. (2017). 11-823: Conlanging: Building a Talking Clock.
URL http://tts.speech.cs.cmu.edu/11-823/hints/clock.html

Black, A., & Lenzo, K. (2000). Limited domain synthesis. *Interspeech*.

Brinklow, N. T., Littell, P., Lothian, D., Pine, A., & Souter, H. (2019). Indigenous Language Technologies & Language Reclamation in Canada. *Proceedings of the 1st International Conference on Language Technologies for All*, (pp. 402–406).

Canadian Encyclopedia (2020). Indigenous Languages in Canada | The Canadian Encyclopedia.
URL https://www.thecanadianencyclopedia.ca/en/article/aboriginal-people-languages

Centre for Speech Technology Research (2015). Speech recorder.
URL https://www.cstr.ed.ac.uk/research/projects/speechrecorder/

Chien, C.-M. (2021). ming024/FastSpeech2. Original-date: 2020-06-25T13:57:53Z.
URL https://github.com/ming024/FastSpeech2

Chomsky, N., & Halle, M. (1991). *The sound pattern of English*. MIT Press.
URL https://hdl.handle.net/2027/heb.08419

Clark, R. A., Richmond, K., & King, S. (2007). Multisyn: Open-domain unit selection for the festival speech synthesis system. *Speech Communication*, *49*(4), 317–330.

Duddington, J., & Dunn, R. (2007). eSpeak: Speech Synthesizer.
URL `http://espeak.sourceforge.net/`

EPA (2019). Emissions generation resource integrated database (egrid).

First Peoples' Cultural Council (2018). Report on the status of b.c.
URL `https://fpcc.ca/resource/fpcc-report-of-the-status-of-b-c-first-nations-languages-2018/`

Forbes, C., Davis, H., Schwan, M., & Gitksan Research Lab (2017). Three gitksan texts. *45*, 47–89.
URL `https://lingpapers.sites.olt.ubc.ca/files/2017/08/Gitlab`

Fortier, K. (2016). Shifty Vowels: Variation in Dialectal Lowering in Gitksan. *Canadian Acoustics*, *44*(3).
URL `https://jcaa.caa-aca.ca/index.php/jcaa/article/view/2984`

Gölge, E. (2020). Solving Attention Problems of TTS models with Double Decoder Consistency.
URL `http://erogol.com/solving-attention-problems-of-tts-models-with-double-decoder-consistency/`

Hallett, D., Chandler, M., & Lalonde, C. (2007). Aboriginal language knowledge and youth suicide. Cognitive Development, 22, 393-399. *Cognitive Development*, *22*, 392–399.

Harrigan, A., Arppe, A., & Mills, T. (2019). A Preliminary Plains Cree Speech Synthesizer. In *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1* (*Papers*), (pp. 64–73). Honolulu: Association for Computational Linguistics.
URL `https://aclanthology.org/W19-6009`

Hench, C. (2019). Syllabipy universal syllabification algorithm.
URL `https://github.com/henchc/syllabipy`

Hu, Z. (2021). DCT (Discrete Cosine Transform) for PyTorch.
URL https://github.com/zh217/torch-dct

Hwang, M.-J., Yamamoto, R., Song, E., & Kim, J.-M. (2020). TTS-by-TTS: TTS-driven Data Augmentation for Fast and High-Quality Speech Synthesis.
URL http://arxiv.org/abs/2010.13421

Innu-Atikamekw-Anishnabeg Coalition (2020). Export of Canadian Hydropower to the United States - First Nations in Québec and Labrador Unite to Oppose Hydro-Québec Project.
URL https://www.newswire.ca/news-releases/export-of-canadian-hydropower-to-the-united-states-first-nations-in-quebec-and-labrador-unite-to-oppose-hydro-quebec-project-845431188.html

Ito, K., & Johnson, L. (2017). The LJ speech dataset. https://keithito.com/LJ-Speech-Dataset/.

James, J., Shields, I., Berriman, R., Keegan, P., & Watson, C. (2020). Developing resources for te reo māori text to speech synthesis system. In P. Sojka, I. Kopeček, K. Pala, & A. Horák (Eds.) *Text, Speech, and Dialogue*, (pp. 294–302).

Jia, Y., Johnson, M., Macherey, W., Weiss, R. J., Cao, Y., Chiu, C.-C., Ari, N., Laurenzo, S., & Wu, Y. (2019). Leveraging Weakly Supervised Data to Improve End-to-End Speech-to-Text Translation.
URL http://arxiv.org/abs/1811.02050

Kazantseva, A., Maracle, O. B., Maracle, R. J., & Pine, A. (2018). Kawennón:nis: the wordmaker for Kanyen'kéha. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, (pp. 53–64). Santa Fe, New Mexico, USA: Association for Computational Linguistics.
URL https://aclanthology.org/W18-4806

Keegan, T. T. (2019). Issues with Māori sovereignty over Māori language data.
URL https://www.youtube.com/watch?v=fodGN4kaEcI

King, S., & Taylor, P. (2000). Detection of phonological features in continuous speech using neural networks. *Computer Speech & Language*, *14*(4), 333–353.
URL https://www.sciencedirect.com/science/article/pii/S0885230800901487

Kominek, J., Schultz, T., & Black, A. (2008). Synthesizer Voice Quality of New Languages Calibrated with Mean Mel Cepstral Distortion. In *Proceedings of SLTU-2008*, (pp. 63–68).

Kong, J., Kim, J., & Bae, J. (2020). HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis.
URL `http://arxiv.org/abs/2010.05646`

Krauss, M. (1992). The world's languages in crisis. *Language*, *68*(1), 4–10. Publisher: Linguistic Society of America.
URL `https://muse.jhu.edu/article/452858#sub01`

Kubichek, R. (1993). Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1, (pp. 125–128 vol.1).

Kubichek, R. F. (1991). Standards and technology issues in objective voice quality assessment. *Digital Signal Processing*, *1*(2), 38–44.
URL `https://www.sciencedirect.com/science/article/pii/1051200491900942`

Kumar, A. (2017). Spoken Speaker Identification based on Gaussian Mixture Models : Python Implementation.
URL `https://appliedmachinelearning.blog/2017/11/14/spoken-speaker-identification-based-on-gaussian-mixture-models-python-implementation/`

Laptev, A., Korostik, R., Svischev, A., Andrusenko, A., Medennikov, I., & Rybin, S. (2020). You Do Not Need More Data: Improving End-To-End Speech Recognition by Text-To-Speech Data Augmentation. *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics* (*CISP-BMEI*), (pp. 439–444).
URL `http://arxiv.org/abs/2005.07157`

Levasseur, A., Mercier-Blais, S., Prairie, Y. T., Tremblay, A., & Turpin, C. (2021). Improving the accuracy of electricity carbon footprint: Estimation of hydroelectric reservoir greenhouse gas emissions. *Renewable and Sustainable Energy Reviews*, *136*, 110433.

URL `https://www.sciencedirect.com/science/article/pii/S1364032120307206`

Li, C., Ma, X., Jiang, B., Li, X., Zhang, X., Liu, X., Cao, Y., Kannan, A., & Zhu, Z. (2017). Deep speaker: an end-to-end neural speaker embedding system.

Littell, P., Kazantseva, A., Kuhn, R., Pine, A., Arppe, A., Cox, C., & Junker, M.-O. (2018). Indigenous language technologies in Canada: Assessment, challenges, and successes. In *Proceedings of the 27th International Conference on Computational Linguistics*, (pp. 2620–2632). Santa Fe, New Mexico, USA: Association for Computational Linguistics.
URL `https://aclanthology.org/C18-1222`

Liu, P., Wu, X., Kang, S., Li, G., Su, D., & Yu, D. (2019). Maximizing Mutual Information for Tacotron. *arXiv:1909.01145* [*cs, eess*]. ArXiv: 1909.01145.
URL `http://arxiv.org/abs/1909.01145`

Luo, R., Tan, X., Wang, R., Qin, T., Li, J., Zhao, S., Chen, E., & Liu, T.-Y. (2021). LightSpeech: Lightweight and Fast Text to Speech with Neural Architecture Search.
URL `http://arxiv.org/abs/2102.04040`

Luu, C. (2021). cvqluu/TDNN. Original-date: 2019-03-13T18:58:14Z.
URL `https://github.com/cvqluu/TDNN`

Magueresse, A., Carles, V., & Heetderks, E. (2020). Low-resource Languages: A Review of Past Work and Future Challenges.
URL `http://arxiv.org/abs/2006.07264`

Marmion, D., Obata, K., & Troy, J. F. (2014). *Community, identity, wellbeing: the report of the Second National Indigenous Languages Survey*. Australian Institute of Aboriginal and Torres Strait Islander Studies.

Mashimo, M., Toda, T., Shikano, K., & Campbell, N. (2001). Evaluation of cross-language voice conversion based on gmm and straight. In *INTERSPEECH*.

Mortensen, D. R., Littell, P., Bharadwaj, A., Goyal, K., Dyer, C., & Levin, L. S. (2016). Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, (pp. 3475–3484). ACL.

Oster, R., Grier, A., Lightning, R., Mayan, M., & Toth, E. (2014). Cultural continuity, traditional indigenous language, and diabetes in alberta first nations: a mixed methods study. *International journal for equity in health*, *13*, 92.

Pine, A., & National Research Council Canada (2021). G2P: grapheme-to-phoneme transductions that preserve input and output indices.
URL `https://github.com/roedoejet/g2p`

Pine, A., & Turin, M. (2017). Language Revitalization. ISBN: 9780199384655.
URL `https://oxfordre.com/linguistics/view/10.1093/acrefore/9780199384655.001.0001/acrefore-9780199384655-e-8`

PyTorch (2021). BCEWithLogitsLoss — PyTorch 1.9.0 documentation.
URL `https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html`

Qamhan, M. A., Alotaibi, Y. A., Seddiq, Y. M., Meftah, A. H., & Selouani, S. A. (2021). Sequence-to-Sequence Acoustic-to-Phonetic Conversion Using Spectrograms and Deep Learning. *IEEE Access*, *9*, 80209–80220. Conference Name: IEEE Access.

Qian, Y., Soong, F. K., & Yan, Z. (2013). A Unified Trajectory Tiling Approach to High Quality Speech Rendering. *IEEE Transactions on Audio, Speech, and Language Processing*, *21*(2), 280–290. Conference Name: IEEE Transactions on Audio, Speech, and Language Processing.

Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., & Liu, T.-Y. (2021). FastSpeech 2: Fast and High-Quality End-to-End Text to Speech.
URL `https://arxiv.org/abs/2006.04558`

Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z., & Liu, T.-Y. (2019). FastSpeech: Fast, Robust and Controllable Text to Speech.
URL `http://arxiv.org/abs/1905.09263`

Reyhner, J. (2010). Indigenous language immersion schools for strong indigenous identities. *Heritage Language Journal*, (pp. 299–313).

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R. J., Saurous, R. A., Agiomyrgiannakis, Y., & Wu,

Y. (2018). Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions.
URL `http://arxiv.org/abs/1712.05884`

Simon King (2021). Speech zone.
URL `https://speech.zone/`

Singh, A. K. (2008). Natural Language Processing for Less Privileged Languages: Where do we come from? Where are we going? In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.
URL `https://aclanthology.org/I08-3004`

Statistics Canada (2016). Census of population.
URL `https://www12.statcan.gc.ca/census-recensement/2016/dp-pd/index-eng.cfm`

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (pp. 3645–3650). Florence, Italy: Association for Computational Linguistics.
URL `https://aclanthology.org/P19-1355`

Sutherland, W. J. (2003). Parallel extinction risk and global distribution of languages and species. *Nature*, *423*(6937), 276–279. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 6937 Primary_atype: Research Publisher: Nature Publishing Group.
URL `https://www.nature.com/articles/nature01607`

Tachibana, H., Uenoyama, K., & Aihara, S. (2018). Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
URL `http://dx.doi.org/10.1109/ICASSP.2018.8461829`

Tan, X., Qin, T., Soong, F., & Liu, T.-Y. (2021). A Survey on Neural Speech Synthesis.
URL `http://arxiv.org/abs/2106.15561`

Taylor, P., Black, A. W., & Caley, R. (1998). The architecture of the festival speech synthesis system. In *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*.

Truth and Reconciliation Commission of Canada (2015). *Canada's Residential Schools: Reconciliation: The Final Report of the Truth and Reconciliation Commission of Canada, Volume 6*. McGill-Queen's University Press.
URL `http://www.jstor.org/stable/j.ctt19qghck`

Tu, T., Chen, Y.-J., Yeh, C.-c., & Lee, H.-y. (2019). End-to-end Text-to-speech for Low-resource Languages by Cross-Lingual Transfer Learning.
URL `http://arxiv.org/abs/1904.06508`

Urrea, A. M., Camacho, J. A. H., & Garćıa, M. A. (2009). Towards the Speech Synthesis of Raramuri: A Unit Selection Approach based on Unsupervised Extraction of Suffix Sequences.
URL `https://www.semanticscholar.org/paper/Towards-the-Speech-Synthesis-of-Raramuri%3A-A-Unit-on-Urrea-Camacho/fe43914e0f484a81436c73939dda5b24290806e6`

Valin, J.-M. (2018). A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement. In *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, (pp. 1–5). Vancouver, BC: IEEE.
URL `https://ieeexplore.ieee.org/document/8547084/`

Wang, W. S.-Y. (1967). Phonological Features of Tone. *International Journal of American Linguistics*, *33*(2), 93–105.
URL `http://www.jstor.org/stable/1263953`

Wells, D., & Richmond, K. (2021). Cross-lingual Transfer of Phonological Features for Low-resource Speech Synthesis. In *Proc. 11th ISCA Speech Synthesis Workshop*.

Whalen, D., Moss, M., & Baldwin, D. (2016). Healing through language: Positive physical health effects of indigenous language use. *F1000Research*, *5*, 852.

Whitman, R., Sproat, R., & Shih, C. (1997). *A Navajo Language Text-to-Speech Synthesizer*. AT&T Bell Laboratories.

Ze, H., Senior, A., & Schuster, M. (2013). Statistical parametric speech synthesis using deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, (pp. 7962–7966). Vancouver, BC, Canada: IEEE.
URL `http://ieeexplore.ieee.org/document/6639215/`

Zen, H., Tokuda, K., & Black, A. W. (2009). Statistical parametric speech synthesis. *Speech Communication*, *51*(11), 1039–1064.

Zheng, Y., Wang, X., He, L., Pan, S., Soong, F. K., Wen, Z., & Tao, J. (2019). Forward-Backward Decoding for Regularizing End-to-End TTS.
URL `https://arxiv.org/abs/1907.09006`

Zhu, C., An, K., Zheng, H., & Ou, Z. (2021). Multilingual and crosslingual speech recognition using phonological-vector based phone embeddings.
URL `http://arxiv.org/abs/2107.05038`