Applying Automatic Speech Recognition to Indigenous Language Documentation: A Case Study with Hul'q'umi'num'

By

Xin He Jiang

B.A., University of Alberta, 2020

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF ARTS

in the School of Languages, Linguistics, and Cultures

We acknowledge and respect the Lək̓ʷəŋən (Songhees and Xʷsepsəm/Esquimalt) Peoples on whose territory the university stands, and the Lək̓ʷəŋən and W̱SÁNEĆ Peoples whose historical relationships with the land continue to this day.

Supervisory Committee

Applying Automatic Speech Recognition to Indigenous Language Documentation: A Case Study with Hul'q'umi'num'

By Xin He Jiang

B.A., University of Alberta, 2020

Supervisory Committee

Dr. Sonya Bird, School of Languages, Linguistics, and Cultures, University of Victoria

Co-Supervisor

Dr. Suzanne Urbanczyk, School of Languages, Linguistics, and Cultures, University of Victoria

Co-Supervisor

Dr. Christopher Cox, School of Linguistics and Language Studies, Carleton University

Supervisory Committee Member

Abstract

The process of documenting Indigenous languages can create a large amount of audio recordings that are difficult to convert into a written form. Speeding up the transcription process using automatic speech recognition could help the Hul'q'umi'num' Language & Culture Society to create pedagogical materials and make their recordings more accessible. In this project, I trained a language model known as XLS-R on Hul'q'umi'num' audio recordings to determine how accurately it can transcribe Hul'q'umi'num', whether particular linguistic and orthographic features are more difficult for XLS-R to transcribe, and what amount of time and computational resources the training takes. The model reached a CER of 11.1% and WER of 50% using 26 minutes of continuous speech. Most phonemes could be transcribed with high accuracy but the model showed difficulties with segmenting words, differentiating glottalized consonants from plain consonants, determining vowel length, and predicting the placement of glottal stops.

Table of Contents

List of Tables

List of Figures

List of Examples

Acknowledgements

I would like to thank my co-supervisors Dr. Sonya Bird and Dr. Suzanne Urbanczyk for their immeasurable contributions to this work. This research could not have been completed if they had not shared their knowledge and experience with me. Dr. Urbanczyk and Dr. Bird show others so much kindness, empathy, and consideration that I strive to better myself to honour the example that they have set.

I would also like to thank Dr. Christopher Cox, who recommended the language model for this project and provided so much support for the computational aspect of this project.

I would like to show my appreciation for Dr. Donna Gerdts and the HLCS for their interest in my work and thank them for allowing me to use their recordings and transcriptions. I extend my thanks across time to the late Dr. Sti'tum'at Ruby Peter for her recordings as well as her incredible contributions to Hul'q'umi'num' research and education.

I am also grateful for the School of Languages, Linguistics, and Cultures at the University of Victoria and the members of the Hul'q'umi'num' Working Group for sharing their thoughts on my work and making the writing process more bearable.

Finally, I would like to thank my family and my partner, Kyle. I could not have done this without their love and support. All the words in every language would not be enough to communicate my appreciation.

Chapter 1

Introduction

The goal of this thesis project is to train an automatic transcriber in the hopes of making the transcription process faster and making recordings in the Hul'q'umi'num' language more accessible. Transcription, when used by linguists, generally refers to the act of converting spoken language to a written form. In order to transcribe a recording, a linguist or fluent speaker will typically listen to it several times while writing and checking their hypotheses about what they are hearing, then ask for a second opinion if they remain unsure. Depending on many factors like familiarity with the language, it may take a linguist "30 minutes to 2 hours to transcribe and translate a minute of speech" (Adams et al., 2018, p. 3356). Transcription is so time-consuming, when compared to other steps in *natural language processing* (NLP), that it has been referred to as "the transcription bottleneck" (Bird, 2020, p. 713). Manual transcription has benefits, such as being a useful exercise for language learning. However, experienced transcribers may wish to speed up the process using *automatic speech recognition* (ASR), especially in cases where there is limited capacity for the work and/or it needs to be done on a tight timeline.

The Hul'q'umi'num' Language & Culture Society (HLCS) —which I worked with for this project—has archived recordings that they would like transcribed to create a database of stories. However, there are few available skilled transcribers and their time is in high demand. If ASR could be used to create transcriptions that are accurate enough that several stories could be edited in the amount of time that transcribing a single story from scratch would take, many recordings could be made accessible fairly quickly for linguistic analysis, the creation of pedagogical tools, or for those who would like to read a story for any other purpose. This thesis investigates whether *XLS-R*, one of the most promising ASR systems, is effective for transcribing

the Hul’q’umi’num’ language. Additionally, this project aims to describe what is necessary to train XLS-R to transcribe Hul’q’umi’num’ in amount of human-transcribed recordings, computational requirements, and time so that the HLCS and anyone interested in using ASR for language documentation can determine whether training a language model would be feasible or beneficial. The errors made by XLS-R were also examined thoroughly to find the factors impacting XLS-R’s transcription accuracy, particularly the influence of linguistic and orthographic features of Hul’q’umi’num’.

Chapter 2

Literature Review

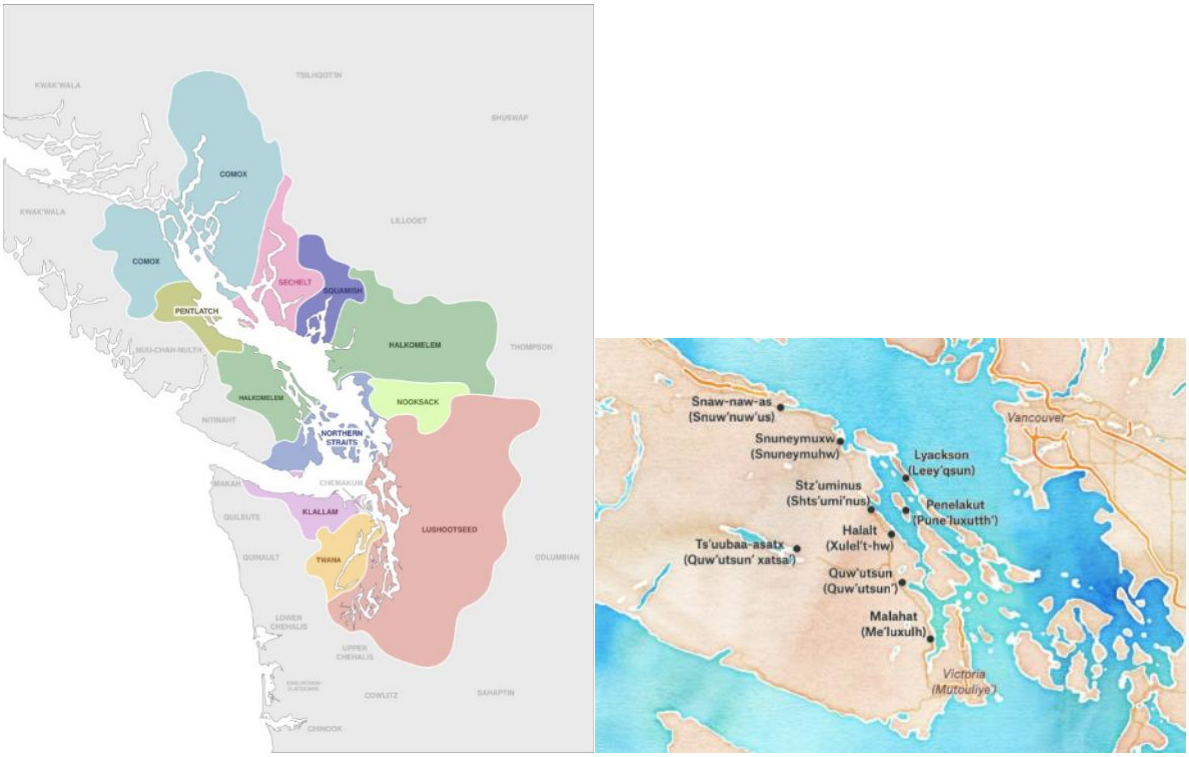
This literature review begins with basic information about Hul'q'umi'num', such as the number of speakers. This chapter also contains information regarding who I am working with in the Hul'q'umi'num'-speaking community for this project. I then provide a brief description of the history of ASR systems, explain why the automatic transcriber XLS-R was chosen for this project, show previous research evaluating XLS-R, and discuss research on Hul'q'umi'num' and how this project fits in this context. The final section of this chapter outlines the research questions and goals of this project.

2.1.1 Hul'q'umi'num' Language Background

Hul'q'umi'num', also known as Island Halkomelem, is one of three dialects of Halkomelem, a Central Salish language. Hul'q'umi'num' is "spoken by the people whose territory extends along the Western Salish Sea, from Me'luxulh (Malahat) to Snuw'nuw'us (Nanoose) on Vancouver Island, BC, Canada, and the neighbouring islands in the Strait of Georgia" (Bird et al., 2023, pp. 298–299), while hen'q'əmin'əm' (Downriver Halkomelem) is spoken at the mouth of the Fraser River in the Lower Mainland and Halq'eméylem (Upriver Halkomelem) is spoken further up the river in the Fraser Valley, to Hope. Maps of the distribution of Coast Salish languages and the Hul'q'umi'num'-speaking community are show in Figure 2.1.

Figure 2.1

Coast Salish Linguistic Distribution and Hul'q'umi'num' Community Map



Note. (Left) Map of Coast Salish linguistic distribution in the early to mid 1800s, by Noahedits (2019), licensed under CC BY-SA 4.0. (Right) Hul'q'umi'num' community map from Bird et al. (2023, p. 299) created by Helen Zhang and modified by Rae Anne Claxton and Maida Percival, licensed under CC BY-NC 4.0.

The most recent report by Gessner et al. (2023) stated that the combined varieties of Halkomelem had 105 fluent speakers, 617 total speakers, and 1,901 language learners. Hul'q'umi'num' is estimated to have "approximately twenty to thirty L1 speakers, over 200 fluent L2 speakers, and over 1,000 learners of all ages" (Bird et al., 2023, p. 299). Much work is currently underway to document and revitalize this language. The HLCS is an organization that supports language work in different Hul'q'umi'num' communities to meet their mandate of "[ensuring] the survival of traditional knowledge and values" (HLCS, n.d.). For this project, I

have been in contact with this organization through Dr. Donna Gerdts. Dr. Gerdts is a professor at Simon Fraser University, who has been working with speakers of Hul'q'umi'num' since the 1970s. She is on the HLCS governing board and generously acts as a liaison between the HLCS and outsider researchers, including at UVic. She expressed to me that the HLCS is interested in creating a database of transcribed stories for the creation of pedagogical materials using the assistance of ASR but would like to have evidence that the generated transcriptions will be accurate enough to justify the computational resources and time spent training such a tool. More detailed information about my communications with the HLCS via Dr. Gerdts is available in Section 3.2.

The Hul'q'umi'num' orthography, also known as the Hul'q'umi'num' practical orthography or alphabet, uses characters that can be easily typed with standard English keyboards. This orthography is shown in Table 2.1 next to the characters for the same sounds in the American Phonetic Alphabet, which are enclosed in square brackets. Apostrophes mark glottalization in glottalized resonants as well as ejective stops and affricates. Rounded consonants are marked by <w> and long vowels are indicated with double letters.

Table 2.1

Hul'q'umi'num' Consonant Chart and Vowel Chart

	Labial	Dental	Alveolar	Lateral	Palatal	Velar	Rounded velar	Uvular	Rounded uvular	Glottal
Stop										
Plain	p / [p]		t / [t]			k / [k]	kw / [kʷ]	q / [q]	qw / [qʷ]	
Glottalized	p' / [ṗ]		t' / [t̓]				kw' / [k̓ʷ]	q' / [q̓]	qw' / [q̓ʷ]	' / [ʔ]
Affricate										
Plain		tth / [tᶿ]	ts / [c]		ch / [č]					
Glottalized		tth' / [tᶿ̓]	ts' / [c̓]	tl' / [ƛ̓]	ch' / [č̓]					
Fricative		th / [θ]	s / [s]	lh / [ɬ]	sh / [š]	h / [x]	hw / [xʷ]	x / [x̌]	xw / [x̌ʷ]	
Resonant										
Plain	m / [m]		n / [n]	l / [l]	y / [y]		w / [w]			
Glottalized	m' / [m̓]		n' / [n̓]	l' / [l̓]	y' / [y̓]		w' / [w̓]			

	Front		Mid		Back	
High	**short**	i / [i]	u / [ə]		**short**	ou / [u]
	long	ii / [i:]			**long**	oo / [u:]
Mid	**short**	e / [e]				
	long	ee / [e:]				
Low			**short**	a / [a]		
			long	aa / [a:]		

Note. From *Sounds of Hul'q'umi'num': the Hul'q'umi'num' phonetic inventory*, by the Hul'q'umi'num' Language & Culture Society, 2022 (https://sqwal.hwulmuhwqun.ca/wp-content/uploads/2022/05/Hulquminum-phonetic-inventory.pdf). Copyright 2016 by the HLCS.

2.2.1 The History of Automatic Transcribers

In order to understand why XLS-R holds so much potential as a tool for language documentation, the disadvantages of previous ASR systems must be discussed. Previous ASR

systems used *hidden Markov models* (HMMs), which have large resource requirements for training. According to Besacier et al. (2014, p. 88), "the building process of ASR systems [based on HMMs required] transcribed speech recordings from many speakers, pronunciation dictionaries which cover the full vocabulary of at least the training corpus, and massive amounts of text data" to go along with hundreds of hours of audio. While some language communities have hundreds of hours of recorded speech, HMM-based systems require fully transcribed audio, which presents a significant barrier to usage. HMM-based systems are also less accessible for people without powerful computers or access to a server.

Since many language communities lack the resources to train HMM-based ASR systems, other techniques like phoneme recognition have been explored. One phonemic recognition tool—Persephone—was developed by Adams et al. (2018). Instead of recognizing words, Persephone would attempt to recognize individual sounds. This removed the need for an existing lexicon and drastically reduced the required amount of transcribed audio. The rate at which errors are made by an ASR is typically measured using *phone error rate* (PER). Persephone was able to achieve "a lower than 30% PER with 30 minutes of training data" (Adams et al., 2018, p. 3360) and the PER would decrease logarithmically with increased training data. While this result is reasonable, having fewer errors would reduce the workload involved in manually correcting the transcriptions.

Sparse transcription (Bird, 2020) was created to overcome the resource, accuracy, and workflow issues of previous ASR systems, including Persephone, by leveraging the combined powers of the machine, human transcriber, and native speaker. Sparse transcription uses multiple

language models[1] to automatically segment the waveform, display automatically recognized sounds, show suggested words from the lexicon while the linguist is typing, add to the lexicon each time a new word is transcribed, and improve the capabilities of the models as the lexicon grows. Additionally, transcriptions would be sent to a native speaker for approval to make the process collaborative. In theory, sparse transcription requires no pre-transcribed recordings. Although pretraining could be done to improve the accuracy of sparse transcription, this is not required. The current main issue with sparse transcription is that it has not been fully implemented. For example, Lane et al. (2021) describe a word-spotting model, which is used to detect new instances of words already in the lexicon, but this has not actually been implemented in the current version of sparse transcription. In addition, the setup instructions are limited, and it is "no longer actively maintained" (Lane, 2023).

While ASR systems were being developed especially for languages with less transcribed audio, the field of NLP was rapidly changing due to several major innovations. The first[2] of these innovations that will be discussed here is the *Transformer*, proposed by Vaswani et al. (2017), which is an architecture created with a unique mechanism called *attention*. Attention allows a model to focus on specific portions of the input that are the most important to process. This is accomplished by adding information from the surrounding segments into the representation of each *token* (i.e. unit of data) as the model is learning. Alammar & Grootendorst use "The dog chased the squirrel because it" (2024, Chapter 3) as an example prompt to explain this idea. In

[1] The term "language model" or "model" refers to the portion of an ASR system that most people think of as being the "artificial intelligence." In actuality, the model is a structured and summarized collection of data that adjusts based on the input it is given. The process in which the model is made to adjust to produce more desirable outputs is typically called "training" or "learning." The structure of the model as well as other components that support the function of the model are referred to as the architecture. I use "ASR system" in this thesis to refer to the model, architecture, and approach as a whole.

[2] This list is ordered for ease of explanation, not chronological order.

this example, the final word is currently being processed. For a model trained using a dataset of text, using words as tokens, each word would be assigned a level of attention weight. If the model is completely untrained, there should be uniform attention given to each word in the example. If the model is trained on some sentences where “the dog” is used with the pronoun “she,” the model may recognize that the pronoun “it” is refers to “the squirrel” and adjust its weights so that the token “squirrel” has a higher attention weight. Attention allows for long sequences to be used as input without increasing the computational load much. The second and third notable innovations are *self-supervised learning* (SSL) and *pretraining*, which go hand-in-hand with each other. SSL as a term was popularized by LeCun (2019), but existed for years prior as “unsupervised learning.” It refers to when machine learning is done on unlabelled data. An ASR system using SSL would use untranscribed audio as input and find relationships between just the sounds. This differs from *supervised learning*, for which the ASR system would use transcribed audio to find relationships between the audio and transcriptions. Pretraining originated from Hinton et al. (2006) but took many years to be widely used in machine learning. Hinton et al.’s approach split the training of a language model into two phases: pretraining and *fine-tuning*. In the pretraining phase, the model undergoes SSL for rapid learning of a variety of patterns. In the fine-tuning phase, supervised learning trains the model on a particular task, using the patterns learned during pretraining to decrease the supervised learning that has to be done. The innovations of SSL and pretraining drastically reduced the amount of transcribed audio required for ASR. For most ASR systems that are actively used at the time of writing this thesis, including XLS-R, all three concepts are incorporated.

2.2.2 The Relevance of XLS-R

The ASR system used for this project is commonly known as *XLS-R-wav2vec2*, or simply XLS-R (Conneau et al., 2020). The initialism XLS comes from XLM, which stands for "cross-lingual language model" (Lample & Conneau, 2019, p. 1), with the "M" changed to "S" to represent speech. The "R" stands for RoBERTa (Liu et al., 2019), a method for pretraining *BERT* (Devlin et al., 2019), which is a language model based on the Transformer architecture. The "wav2vec2" portion of the name refers to Wav2vec 2.0, which is an ASR system by Baevski et al. (2020) that adapts BERT for usage on speech. Altogether, the name XLS-R means that the system uses Wav2vec 2.0 architecture with self-supervised pretraining on untranscribed speech in different languages and supervised fine-tuning on transcribed speech in the target[3] language. The corpora used to pretrain the XLS-R model used for this project contains 56,000 hours of publicly available unlabelled speech in 53 languages. By using a large multilingual dataset, Conneau et al. trained XLS-R to recognize sound patterns in any language so that minimal data is needed to learn to transcribe one particular language.

XLS-R and similar contemporary ASR systems fix many of the issues that have made automatic transcription difficult to implement for languages with limited transcribed audio. The Transformer architecture reduces computational requirements while self-supervised pretraining creates language models that require very little transcribed speech. The massive amount of untranscribed speech that language models use for SSL has led to these models being referred to as *Large Language Models* (LLMs). XLS-R can transcribe languages in its pretraining corpora with around 10% PER and languages not in its pretraining corpora with an average of 18% PER (Conneau et al., 2020). Although there are many ASR systems that outperform XLS-R in

[3] Target is used in machine learning to refer to the training goal, or what the model is being tasked to do.

accuracy—like MAmba-based DEcoder-ONly approach (Masuyama et al., 2024), Voice-text Language Model (VoxtLM) (Maiti et al., 2023), and XEUS (W. Chen et al., 2024)—most competitive languages models are not pretrained on as much data in as many languages as XLS-R, limiting their application to other languages. Li (2024) and Li et al. (2025) have demonstrated that popular LLMs like ChatGPT have better performance in English, with overall superior performance in languages with high amounts of publicly available training data—such as German, French, and Russian—than languages with less—like Burmese, Kazakh, and Hawaiian.

Another advantage of using XLS-R is that it has been made more accessible than other ASR systems because of the resources that were created for it. The code necessary for training XLS-R has been made available by researchers like von Platen (2021) and Coto-Solano et al. (2022)[4]. They have also made the code accessible through clearly annotating the code, providing additional instructions that are easy to understand, and using Google Colab to reduce limitations related to computational needs. Extensions have also been created to make XLS-R easily usable with *ELAN* (Nijmegen: Max Planck Institute for Psycholinguistics, 2025), a program that is commonly used to annotate sound files. In particular, XLS-R-ELAN (Cox, 2023) was made to allow ELAN users to generate transcriptions directly in ELAN. This extension potentially makes XLS-R more accessible for people wanting to use ASR to support Indigenous language documentation since ELAN is familiar to many researchers, including the HLCS. Rodríguez & Cox (2023) found that using XLS-R-ELAN with their fine-tuned XLS-R model reduced the work of annotating new recordings in ELAN.

[4] Although the code is not directly given in the work cited, Dr. Coto-Solano has provided it at workshops like CoLang (https://www.colang2024.org/) and upon request.

In short, XLS-R is a promising option for using automatic speech recognition to speed up transcription work. Its architecture reduces the amount of transcribed audio in the target language needed for training to a minimum. The massive number of languages in XLS-R's pretraining corpora show its potential to transcribe languages that ASR systems are typically not trained for. XLS-R can be set up by people with no knowledge of computer science and easily dovetailed into the annotation process, thanks to the work of dedicated researchers like Dr. Cox and Dr. Coto-Solano. These features have led XLS-R being tested on several languages, including some Indigenous languages.

2.2.3 Previous Evaluations of XLS-R

Coto-Solano et al. (2022) reported that XLS-R could transcribe Cook Islands Māori with a *character error rate* (CER[5]) of about 6.1%, showing great performance. Similarly, Rodríguez & Cox (2023) reported that using an XLS-R model fine-tuned on about four hours of Tsuut'ina audio resulted in transcriptions that were generally usable (< 20% CER) on the first pass. In comparison, Chen et al. (2023) tested XLS-R on the Central and South American languages Quechua, Bribri, Guarani, Kotiria, Wa'ikhana, and Totonac. They found a wide range of error rates, with Totonac being the most accurately transcribed with CERs ranging from 20.6% to 27.7%. Wa'ikhana was the least accurately transcribed with 55.1% to 62.2% CERs. Chen et al. did not discuss what may have caused this disparity in CER but because the amount of fine-tuning data was controlled, the disparity could be attributed to the differences between these Central and South American languages or the degree of similarity between these languages and those in the XLS-R pretraining corpora. The overall high error rates found by Chen et al. could

[5] CER is used instead of PER in some articles. The distinction between the two measures is whether the model is being trained to transcribe phonemes or not.

be explained by audio quality, considering that radio recordings requiring extensive post-processing were used. Additionally, recordings in different Quechua dialects (Chanca and Collao) were not separated, which introduced more opportunities for errors for the Quechua models.

Despite the diversity of languages in its training corpora, XLS-R may struggle with some languages with fewer hours of recorded speech. For example, Spanish had 168 hours of speech data in one corpus for pretraining and reached 2.9% PER with one hour of fine-tuning data (Conneau et al., 2020). Swedish had three hours of speech data in the same corpus and had the much higher 12.2% PER with the same amount of fine-tuning speech. Overall, the training corpora for XLS-R contain massive amounts of Indo-European languages, causing those languages to be transcribed more accurately.

However, linguistic and orthographic differences can also cause differences in transcription accuracy. According to Gale et al., "While it has been long known that neural language models are able to capture implicit information about phonology from orthography (Elman, 1990; Prince and Smolensky, 1997), the extent to which this occurs will depend on the degree to which the model's unit of representation maps to the writing system in question's representation of phonology" (2023, p. 212), suggesting that languages with mostly one-to-one correspondences between letter/digraphs and sounds—like Spanish—may be transcribed more accurately than those that are less orthographically transparent. Loweimi et al. (2023) created a detailed analysis of language model errors at the phonetic level that showed that there are factors causing individual phonemes to have different error rates—but their analysis only used English phonemes and phone recognition systems. Hul'q'umi'num' is a language with relatively high morphological complexity that is mostly orthographically transparent and has many phonemes that English does not. Such

morphological complexity was expected to impact the *word error rate* (WER) of the fine-tuned XLS-R model. WER is a commonly used metric for assessing language models, but it may be less useful for languages that are on either side of the extremes of morphological complexity (D K et al., 2025), referring to languages in which words consist of a single morpheme representing a basic unit of meaning (e.g., Mandarin Chinese) and languages in which single words can have many morphemes such as Hul'q'umi'num'. The expectation that the WER of the XLS-R model fine-tuned for this project would be higher than similar models trained on less morphologically complex languages was supported by Otmakhova et al. (2022), who found that the accuracy of BERT-based models could be affected by the target language's morphological complexity, head directionality, agreement marking, gender marking, and whether it has fixed or free word order. In Babu et al.'s (2021) evaluations of larger XLS-R models, more morphologically complex languages had higher WERs than less morphologically complex languages (e.g., the best Swahili WER was 21% compared to 12% for Italian). However, these error rates cannot be directly compared as the more morphologically complex languages comprised much smaller portions of the pretraining data. The fine-tuning data that the more morphologically complex languages received were also much shorter and noisier. Similarly, Conneau et al.'s (2020) results were also difficult to compare due to the differences in pre-training and fine-tuning data between languages.

Previous research into XLS-R and other ASR systems show that multiple factors can affect the models' transcription accuracy, demonstrating that great outcomes are achievable and more research into linguistic and orthographic factors affecting transcription accuracy can be done. While the morphological complexity of Hul'q'umi'num' suggests that the WER of the fine-tuned XLS-R should be high, Hul'q'umi'num' is also quite orthographically transparent compared to other languages, which means that the CER for this project was expected to be low.

2.2.4 Related Hul'q'umi'num' Research

One way that this proposed project may help with documentation and revitalization efforts of the HLCS is through analyzing the errors made by XLS-R. Because ASR systems will transcribe using mostly information from the sound signal, repeated errors tend to occur due to either some discrepancy between what is said and written in the training data, or the language having multiple allophones for a given phoneme. One example of this discrepancy would be glottalization. Hul'q'umi'num' transcribers will often transcribe glottalized resonants and glottal stops based on phonological and morphological knowledge in places where no glottalization is pronounced (S. Bird, personal communication, May 1, 2024). An example of multiple allophones for a given phoneme in Hul'q'umi'num' would be /p/, /t/, and /q/ being unaspirated word-initially and medially, but aspirated word-finally (Kava, 1969). Both of these patterns will likely result in XLS-R transcribing these sounds with lower accuracy than sounds that are more consistent. Allophony is generally assumed to have a large effect on error rates in computational linguistics and has been described in studies such as Adams et al. (2018). By analyzing the errors made by XLS-R, I will potentially provide further evidence for Kava's descriptions of allophonic variation, discover patterns in Hul'q'umi'num' that have not been previously documented, or find discrepancies between how linguists have transcribed Hul'q'umi'num' and how speakers actually produce the language. The results of analyzing the errors made by XLS-R will be shared with the HLCS; potentially these can be used as part of creating pedagogical resources and approaches, to help learners understand the details of fluent pronunciation.

2.4 Research Questions & Goals

The goal of this project is to present to the HLCS what the output of XLS-R is and what the training requires in terms of resources, so that they can decide whether it is worth using.

This project aims to answer the following questions:

1. How well does XLS-R perform on Hul'q'umi'num', and therefore how useful might it be in the transcription process?
2. Can the errors made by XLS-R when transcribing Hul'q'umi'num' teach us anything about Hul'q'umi'num' phonology and the discrepancy between phonemic (and orthography) forms and phonetic realization?
3. How useful would XLS-R be for people working to document other languages, when considering the resources and requirements to use it?
4. More generally, what factors could be causing XLS-R to transcribe languages that it has not encountered in pretraining with higher or lower accuracy?

Chapter 3

Methods

The following sections will lay out the methods used for this project in chronological order. I begin with self-location by stating my background and how it affects my worldview and biases. This is followed by an outline of my communications with the HLCS and the values that I have aimed to centre in this project to ensure that their recordings and transcripts are used respectfully. I then list all the materials for this project then provide details about the data and code used for fine-tuning. The next sections cover the process for fine-tuning XLS-R by first focusing on data preparation and other aspects of the XLS-R setup then detailing what occurs during and after fine-tuning. Finally, I explain how the outputs of XLS-R, namely the transcriptions and error rates, were categorized and analyzed.

3.1 Situating Myself & My Work

I am a non-Indigenous Chinese Settler Canadian. When I was four years old, my family immigrated from Northeastern China to Western Canada. I grew up in Burnaby, B.C., which is on the eastern side of the unceded lands of the xʷməθkʷəy̓əm, Sḵwx̱wú7mesh, səlilwətaɬ, and kʷikʷəƛ̓əm Peoples. At 18 years old, I moved to pursue my undergraduate degree at the University of Alberta, which is on the territories of the Néhiyaw, Niitsítapi, Métis, Nakoda, Dene, Haudenosaunee and Anishinaabe Peoples. I currently live and study on the unceded lands of the Lək̓ʷəŋən and W̱SÁNEĆ Peoples. The Settler aspect of my identity requires some unpacking, as is the case for many non-Indigenous people of colour. Lowman and Barker "position Indigenous and Settler as identities "always in a relationship"" (2015, Chapter 1), referring to the complex ways that factors like race and class affect the way that people experience settler colonial structures and systems. Wong accurately summarizes the space within

the Indigenous/Settler relationship occupied by Asian Canadians as "racialized subjects who have inherited the violence of colonization" (2008, Section Introduction); while Chinese immigrants are excluded from some social structures and systems—a historical example being the Chinese Immigration Act of 1923—the history of Native-Asian relations contains both camaraderie and Asian complicity in colonization. Although my family was not a part of this early wave of Chinese immigration, these early interactions have rippling effects influencing Indigenous-Asian relations today. I believe that it is the responsibility of myself and others who have benefited from immigrating to Canada to acknowledge that we are able to live on the lands of Indigenous Peoples as a result of colonization, be conscious of the biases that we may hold not just as individuals but as communities with complex histories in Canada, and work towards having respectful relationships with Indigenous Peoples.

When my family arrived in Canada, my parents were advised by a doctor to raise me speaking, reading, and writing exclusively in English, for fear that using my heritage language would hinder my ability to assimilate. As a result, I did not develop the skills to communicate any complex or meaningful ideas with my family until I took on the task of learning Chinese as an adult. While experiencing the difficulty of learning a language as an adult, I also felt the immense privilege of having endless language-learning resources available in my goal language. When experiencing my first linguistics courses at the University of Alberta, I became interested in researching tools for aiding language documentation and revitalization because of Dr. Antti Arppe, my supervisor at the time, and others working at the Alberta Language Technology Lab to make language technology available for Indigenous languages. This experience, along with a course on language documentation and revitalization in which we got to practice our skills with a speaker of Stoney Nakoda, taught me the responsibilities that linguists have to approach

interactions with L1 speakers with a collaborative mentality that prioritizes how they would like their language and language materials to be shaped.

My first introduction to Hul'q'umi'num' was through Dr. Gerdts's Hul'q'umi'num' Narrative Discourse class that she was teaching at Simon Fraser University. By attending this class, I had the pleasure of working with Hul'q'umi'num' speakers and learners who showed a lot of enthusiasm for increasing the amount and accessibility of language learning resources. This experience, along with Dr. Gerdts expressing interest in using automatic speech recognition for transcription, served as my motivation for this project.

3.2 Community Engagement

In Leonard's (2018) words, "language documentation practices can reinforce colonial power hierarchies and norms in ways that work against the needs and values of Indigenous language communities" (p. 1). Following Leonard's approaches to Indigenous research, this project has centered how the HLCS wants the documentation to exist in the future, focused on having a collaborative relationship, respected the responsibility that comes with knowledge and its dissemination, and engaged with community needs and institutions.

The research topic and questions were created based on the HLCS's priorities. Initially, they expressed that although they are interested in developing a large corpus of transcribed texts, they do not wish to invest time and energy into training a language model without knowing the quality of the transcriptions it will produce or how long it can be used for, given that I will be graduating after this project. I used their concerns about the resource costs and outcomes as the basis for my research questions. I also proposed methods that relied on using only data (transcribed stories with accompanying audio) that is already publicly available and was specifically shared with me for use, to require no work from Hul'q'umi'num' transcribers or

speakers. In addition, I plan to make the model accessible for the HLCS as part of this project. One of the reasons that the HLCS is interested in having a large corpus of transcribed texts is to create pedagogical materials, such as exercises for Hul'q'umi'num' learners to fix the errors in the transcriptions generated by XLS-R. I have expressed my interest in assisting with the creation of these pedagogical materials but will also provide instructions for using the fine-tuned model[6] so that the HLCS can fulfill such goals without me. I gave a description of the project to the HLCS in accessible language to ensure that they understood the project fully and they approved it. I reported the results to Dr. Gerdts with a description of the model's accuracy and examples of transcriptions.

This research project will demonstrate whether it is feasible to train a language model that will be useful for transcribing Hul'q'umi'num', without an unreasonable burden on HLCS members. If this model is deemed to be useful, Dr. Coto-Solano has created written instructions for using XLS-R that will allow HLCS members to use the model fine-tuned for this project or fine-tune their own version. I also recorded a video demonstration of how to use the language model to ensure that the model remains accessible in the future. Hopefully, XLS-R will perform well enough that it can be used on any recordings that the HLCS wants to have transcribed, supporting the HLCS in their goal of having a database of transcribed Hul'q'umi'num' stories.

3.3 Data and Materials

The materials used for this research include a computer, Google web services (including Drive, Sheets, and Colab Notebooks), XLS-R, ELAN, XLS-R-ELAN, Praat, and transcribed Hul'q'umi'num' audio for training and testing. The HLCS provided three WAV files[7] for this

[6] Shown in Appendix A

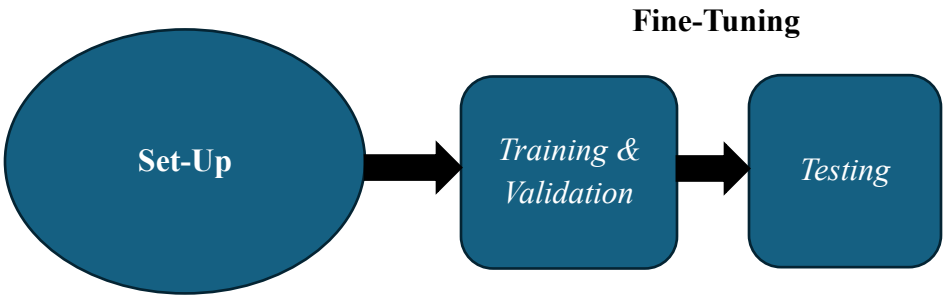
[7] MP3 and MP4 files can be used instead if WAV files are not available.

project with a total length of 53 minutes. This amounted to 26 minutes of speech. All of the recordings were stories told by one speaker, the late Dr. Sti'tum'at Ruby Peter. The three stories are titled *yu 'um'mush tthu t'ut'um'*, "Little Wren Goes Hunting;" *s-hwuhwa'us 'i' lhu q'ullhanumutsun*, "Thunderbird and Orca;" and *q'ise'q 'i' tthu munmaanta'qw*, "Q'ise'q and the Stoneheads." These recordings had already been transcribed by Dr. Peter and Dr. Gerdts,[8] and the transcriptions imported and *time-aligned* into ELAN in the form of EAF files by Webb (2022). Time-aligned, in this case, means that time boundaries were added to match the transcriptions to their corresponding sections of audio. Without this time-alignment, whether done by hand or by algorithm, ASR systems would not be able to map words or characters to sections of audio, resulting in lower accuracy. For this project, the audio was time-aligned based on breath groups, meaning that the boundaries were placed in the pauses before and after continuous speech. Breath groups may be single words, multiple words, or portions of words if the speaker took a breath in the middle of a word.

The steps taken for training XLS-R are shown in Figure 3.1. A more detailed explanation of the set-up step is given at the start of Section 3.3.

Figure 3.1

Flowchart of Steps in Training XLS-R



[8] With English translations that were not included as training data.

The code used for preparing the data before fine-tuning was written by Dr. Rolando Coto-Solano for a workshop on NLP for language documentation and revitalization at CoLang 2024 (Coto-Solano, 2024) and used with his permission. The code used for the fine-tuning was originally written by von Platen (2021) and adjusted by Dr. Coto-Solano. The code was written in the programming language Python. While many others have fine-tuned their own XLS-R models and some have uploaded their code, Dr. Coto-Solano has gone above and beyond to make his code accessible. Although an average computer could be used for training when there is less than an hour of speech, Dr. Coto-Solano's code utilizes Google Drive and Google Colab Notebooks, which are free and easy to use for people who have older computers and are unfamiliar with using servers.[9] He also wrote instructions for using his code in language that a general audience can understand.

After the conclusion of fine-tuning and error analysis, the ELAN extension XLS-R-ELAN (Cox, 2023) was used to generate transcriptions using the language model in a way that was easy to view and try for HLCS members. This was necessary to create an accessible guide for using the model after the conclusion of this project. The "Little Wren Goes Hunting" audio file was used in this guide so that no new audio files were necessary.

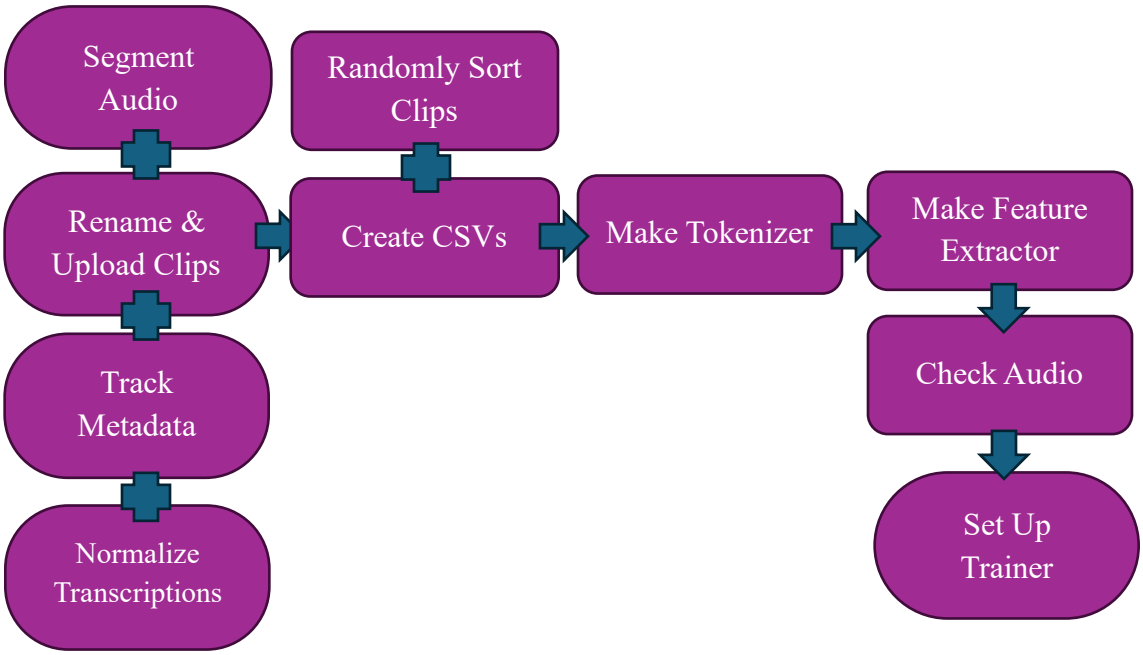
3.4 Data Organization, Preprocessing, and Set-Up for Fine-Tuning

The multiples tasks required before fine-tuning are shown in Figure 3.2. The + signs show that the connecting steps occur at the same time or in rapid succession. Every step described in this section was done automatically using Dr. Coto-Solano's code. No changes were made, with the exception of file names and locations.

[9] Although Google services are online, uploading files to Google Drive does not automatically make the files publicly available. For people with further concerns about Google's data security, the code is easily adaptable to be run on a home computer or private servers.

Figure 3.2

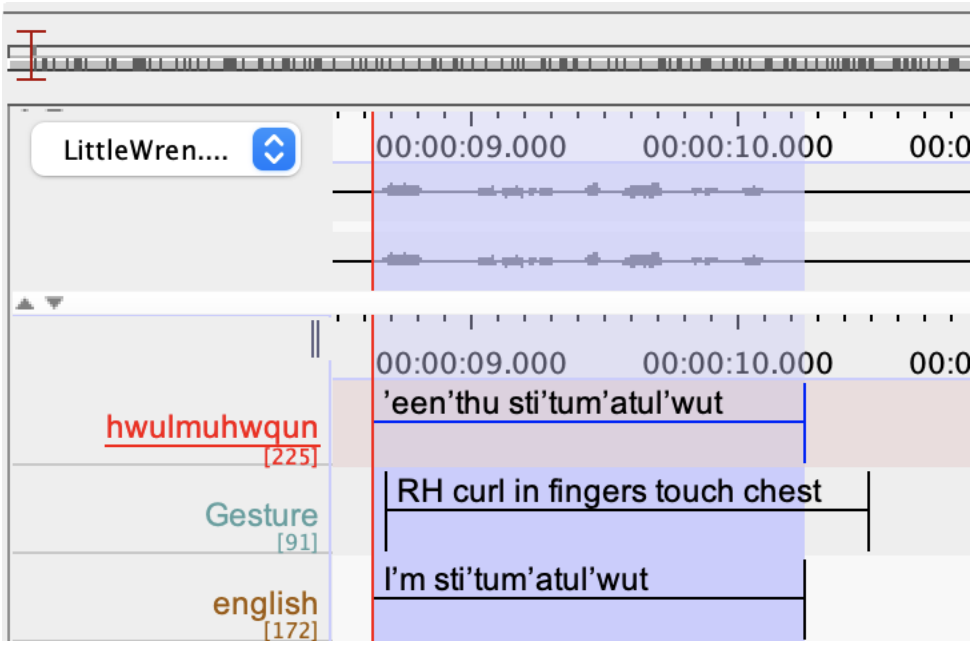
Flowchart of Set-Up Steps



To prepare for fine-tuning XLS-R, the large WAV files had to be cut into short clips, renamed, and the metadata (e.g., time stamps) encoded, all based on the time-alignments in the EAF files. For example, the first annotated transcription for the story “Little Wren Goes Hunting” is shown in Figure 3.3. It is time-aligned with a breath group at around nine to ten seconds in the audio. The English translation is shown on the bottom tier.

Figure 3.3

Example of ELAN Annotation



Note. A "Gesture" tier exists for Webb's (2022) gestural research, but it is not used in this project.

For each annotation, the EAF file contains metadata such as the start time, end time, and transcription. The code compiles a list of start and end times for the annotations. With these lists, FFmpeg (FFmpeg Developers, 2025)—a project containing libraries of code and programs for interacting with various forms of media including audio and video—was used to segment the large WAV files at the specified times, assign new file names to each of the segmented WAV files, and upload all of the segmented files to a Google Drive folder separate from one housing the large file. A Google Sheets spreadsheet was automatically populated with the filenames of the segmented files—along with other metadata including speaker name, file size, duration, and matching transcription. The transcriptions were divided into two columns. One contained the original transcriptions and the other had all characters made lowercase and special characters

removed.[10] This change to the transcriptions is known as normalization and it is done to improve the accuracy of language models, because punctuation typically causes minimal acoustic change and is unpredictable.

The next step was sorting the short WAV files for different parts of fine-tuning. The percentages of files used for training, validating, and testing were set at 80, 10, and 10, respectively. In training, the files are used to adjust the weights and biases of the model to better fit the data. Training is sometimes referred to as “learning” using the analogy of humans and involves adjusting schemas when presented with new information. Validations are evaluations of the model, in which the model’s accuracy is tested on data that it has not used in training to determine if the model has been memorizing individual samples or generalizing. Validation is also done to adjust the hyperparameters of the model, which are characteristics of the model that are set prior to training. Testing is the final evaluation, taking place after the model has completed its rounds of training and validation with no further adjustment done to the model. In machine learning, the 80:20 split for training to validation and testing is considered "balanced and widely accepted” (Sivakumar et al., 2024, p. 7). Although splits of 90:10 or even higher proportions of data allocated to training is standard when low amounts of data are available, the amount of data was sufficient for the 80:20 split because less data is needed to fine-tune a pretrained model like XLS-R compared to fully training a model, as discussed in Section 2.2.2. Files were assigned to each group by creating a list of numbers, shuffling[11] the list, assigning

[10] Apostrophes and spaces were not removed. Apostrophes are used to indicate glottalization in Hul’q’umi’num’. Spaces were replaced with bars (|) for visibility and kept so that the model could learn to predict word boundaries. Double quotation marks were supposed to be removed in this step, but they remained for unknown reasons. The em dash was mistakenly left in as well. An additional character was created for unknown characters that the model may encounter.

[11] Shuffling was done using Python’s shuffle function in the random module, which uses pseudorandom number generation.

each file a now-randomized number, then distributing the 80% of files with the lowest number to training, the next 10% to validation, then the remaining to testing. For each phoneme in Hul'q'umi'num', its frequency in the data allocated for testing and its frequency across all the fine-tuning data are shown in Tables 3.1 and 3.2.

Table 3.1

Phoneme Frequencies - Consonants

	Phoneme	# in Testing Data	# in all Data
Stops	p	6	43
	p’	1	14
	t	93	910
	t’	9	141
	k	1	1
	kw	32	255
	kw’	14	115
	q	19	149
	q’	7	130
	qw	2	49
	qw’	5	36
	’	159	1153
Affricates	tth	17	258
	tth’	3	21
	ts	25	203
	ts’	1	67
	tl’	13	116
	ch	6	19
	ch’	0	0
Fricatives	th	27	251
	s	111	1066
	lh	41	469
	sh	14	173
	h	21	109
	hw	21	238
	x	14	121
	xw	3	44
Resonants	m	33	324
	m’	26	301
	n	105	813
	n’	10	130
	l	30	220
	l’	17	182
	y	20	226
	y’	10	119
	w	20	114
	w’	41	163

Table 3.2

Phoneme Frequencies - Vowels

Phoneme	# in Testing Data	# in all Data
i	91	873
ii	7	34
e	77	560
ee	3	52
u	278	2921
a	56	379
aa	6	40
oo	0	4
ou	0	0

Three comma-separated values (CSV) format files were then created; one for each part of training. CSV is a simple file format that contains tabular data in plain text. This format is often used in programming because it can be easily manipulated with code. The CSV files contain the locations, file sizes, and transcriptions of each short WAV file. These CSV files organize the data that will be given to the trainer.

The next step was making a *feature extractor* and *tokenizer*. The feature extractor maps speech signals to features, which are representations of the important components of the sound, as identified by the model. A tokenizer maps these features to the orthography of the transcriptions. The feature extractor used in von Platen's code is called "Wav2Vec2FeatureExtractor" and the tokenizer is called "Wav2vec2CTCTokenizer" (Wolf et al., 2020). Both were included in the documentation for the pretrained XLS-R model used for this project (Conneau et al., 2020). The tokenizer required a list of all distinct characters in the Hul'q'umi'num' orthography. This list was automatically generated using a function written by von Platen. Default settings[12] were used for all other parameters of the tokenizer. For the feature

[12] Default settings and values refer to values that are written in the code by Wolf et al. (2020) to be used automatically when other values are not provided by the user.

extractor, default values were used for all parameters except return_attention_mask. An attention mask is technically an optional part of the Transformer architecture, but "XLS-R models [*sic*] checkpoints should always use the attention_mask" (von Platen, 2021, n.p.) because they improve generalization by hiding portions of the input that the model is not meant to see could cause it to seem more accurate than it is when it can only access its standard input size.

The final step of data preparation was ensuring that the audio was in the correct format. The audio was loaded in using torchaudio (PyTorch Foundation, 2025) then arranged into a 1-dimensional array, which is a common data structure that XLS-R expects the audio in. Audio resampling was also done to ensure that all files had 16kHz sampling rate. As part of the process, a few samples of audio were individually examined for whether they sounded clear, matched their transcription, and were formatted as a 1-dimensional array.

There are four steps to setting up a trainer, which is the final portion of preparation for fine-tuning (Sun, 2020):

1. Defining the data collator
2. Setting the evaluation metric
3. Loading a pretrained checkpoint
4. Defining the training configuration

One purpose of *data collators* is to package training data into smaller groups. The number of samples is referred to as the *batch size*, which was set at 16, and one pass through all the samples in a batch is called a *step*. After the model processes one step, it learns by adjusting the numbers that it uses in its calculations. The data collator also pads the inputs and outputs. *Padding* is when extra values—typically zeroes—are added to ensure that something is the correct size. XLS-R has much longer input than output lengths because speech is fairly complex

to capture with numeric representations so padding should be done to only the longest sample in a batch for efficiency. WER was set as the evaluation metric, but CER was also calculated. WER is necessary to calculate to compare the results of this project with other trained language models that can reach low CERs while CER is necessary to calculate because the relative morphological complexity of Hul'q'umi'num' will likely mean that the WER does not accurately reflect the quality of the transcriptions. The pretrained checkpoint Wav2Vec2-large-XLSR-53 (Conneau et al., 2020) was then loaded. Checkpoints are versions of the model saved at particular points in training so they can be used when their error rates are ideal, distributed to other people for their own uses, or fine-tuned without fear of losing prior versions. At this point, the audio files have been segmented and checked; a Google Sheet of metadata and CSV files for XLS-R have been created, the audio has been checked, and the trainer is readied.

3.5 Fine-Tuning XLS-R

At the fine-tuning stage, I watched to ensure that the *loss* and error rates were decreasing—which shows that the process is happening correctly because the model is improving—then ran the code to evaluate CER and WER, record the transcriptions, and save a few checkpoints.[13] Loss is a measurement that shows how much the model's predictions differ from its targets. One complete pass through all the training data is called an *epoch*. The fine-tuning of XLS-R took 72 epochs. Using human learning as an analogy again, this can be thought of like going through an entire deck of flash cards 72 times. The training loss, validation loss, and WER were shown for every 200 steps. After training, at every 400 steps, the median CER and WER were calculated. With a batch size of 16, this can be visualized as going through flash

[13] Only the best checkpoint needs to be saved but I saved a few in case they could be of any interest.

cards in sets of 16 cards, finding out how many mistakes you made after every 200 sets of 16 cards, then finding out how accurate you were on average after 400 sets of 16 cards. Typically, only the checkpoint with the lowest WER or CER is saved, and the others are left to avoid taking up unnecessary space in storage. For this project, the checkpoint with the best WER and CER was checkpoint 1200. A list of transcriptions made by the model, paired with their original transcriptions and file names, was also generated and saved. The computational resources used for training were displayed automatically by Colab Notebooks. I recorded these figures to show the HLCS the requirements for fine-tuning XLS-R.

3.6 Error Analysis

To analyze the errors made by XLS-R, the errors were categorized based on how the language *model's transcription* (MT) differed from the original *human transcription* (HT; error type), as well as which transcription more closely matched its spectrogram (error origin). Five error types were coded: mismatch, deletion, insertion, segmentation, and other. An example for each error type is shown in Table 3.3 with English translations in quotations. When a sound in the HT is transcribed as a different sound in the MT, this is labelled a mismatch error. When a sound in the HT is not transcribed in the MT, this is a deletion error. When a sound that is not in the HT is transcribed in the MT, this is an insertion error. When a word boundary is placed incorrectly in the MT, resulting in one word being separated into multiple words or multiple words being combined into one word, this is a segmentation error. When an error cannot be categorized in any of these ways, it is considered an "other error". The subcategories for this type of error included untranscribed speech, codeswitching, and rhetorical lengthening.[14] The error

[14] Transcriptions of speech that could cause such errors can be marked in a column in the Google Sheets metadata table to exclude them from training. This step was skipped for this project to see how XLS-R transcribes these cases.

type categorization notes the differences between the sets of transcriptions but does not make any judgements regarding correctness nor similarity to the acoustic signal.

Table 3.3

Error Types and Examples

Error Type	Human Transcription (HT)	Machine Transcription (MT)
Mismatch	s**u**s "and"	s**i**s
Deletion	hwu**t**[15]	hwu
Insertion	kwus "he, she, it, that he, when he, as he, etc."	kw**s**us
Segmentation	muk**w' s**tem "everything, anything"	mu**kw's**temu
Other (rhetorical lengthening)	m**u.u.u**kw' "all"	m**u** kw'

Since XLS-R was trained on the Hul'q'umi'num' orthography for this project, the error analysis will compare phonemes in the Hul'q'umi'num' phonemic inventory, as opposed to individual orthographic characters. Decisions for error categorization take into account the minimum number of changes required as well as the preservation of phonemes. For example, the transcription of glottalized consonants as their plain counterparts—like <m'>[16] as [m]—is

[15] Unknown translation, root of <s-hwut> "thrush: Swainson's thrush"

[16] From this point forward, I will adopt the following notation for clarity: angle brackets (<>) for HT, square brackets ([]) for MT, and slashes (//) for referring to phonemes outside of their usage in particular transcriptions. Double quotes ("") will continue to be used for English translations as well as glosses. All English translations and glosses in this thesis, with the exception of lines from the stories translated by Dr. Gerdts and Dr. Peters shown in Appendix B, were found in "Hul'q'umi'num' Words: An English-to-Hul'q'umin'um' and Hul'q'umin'um'-to-English Dictionary" (Gerdts et al., 1997) and the "Hul'q'umi'num' 175 Little Words" document (HLCS, 2021).

considered a substitution error rather than a deletion error because this interpretation preserves <m’> as one phoneme being transcribed as a plain phoneme, rather than two phonemes (a resonant and glottal stop) with one phoneme being deleted. Analyzing the data this way makes spotting patterns much easier, but it also introduces one limiting factor. Some Hul’q’umi’num’ words, like /swaaw’lus/ “young men” have two consonants with a glottalization marker between them (/w’l/). In these cases, it is unclear which sound should be considered glottalized (/w’/ vs. /’l/); the preceding consonant was treated as the glottalized one because post-glottalization is more common than pre-glottalization for Hul’q’umi’num’ glottalized resonants (Percival et al., 2025).

All errors were examined in Praat (Boersma & Weenink, 2024) to determine if the spectrogram more closely matched the HT or MT to determine a possible cause for the difference in transcriptions. This additional label is referred to here as the *error origin*. If the spectrogram matched the HT, the error would be considered to have *originated from the model* (OM). If the HT does not match the spectrogram as closely as the MT, the error *origin is the human transcription* (OH). If neither transcription matches the spectrogram well, the *origin is unknown* (OU). Although the HT is referred to as a potential origin of errors in this project, this does not mean that the human transcriber was wrong or inaccurate. The HT prioritizes readability over faithfulness to the audio, making it more suitable for many purposes, such as helping language learners identify what morphemes they are hearing in the narratives. This is the standard for Hul’q’umi’num’ transcription and the target of the language model’s training. Thus, all errors are considered mistakes made by the language model, despite the error origin label. Error origin

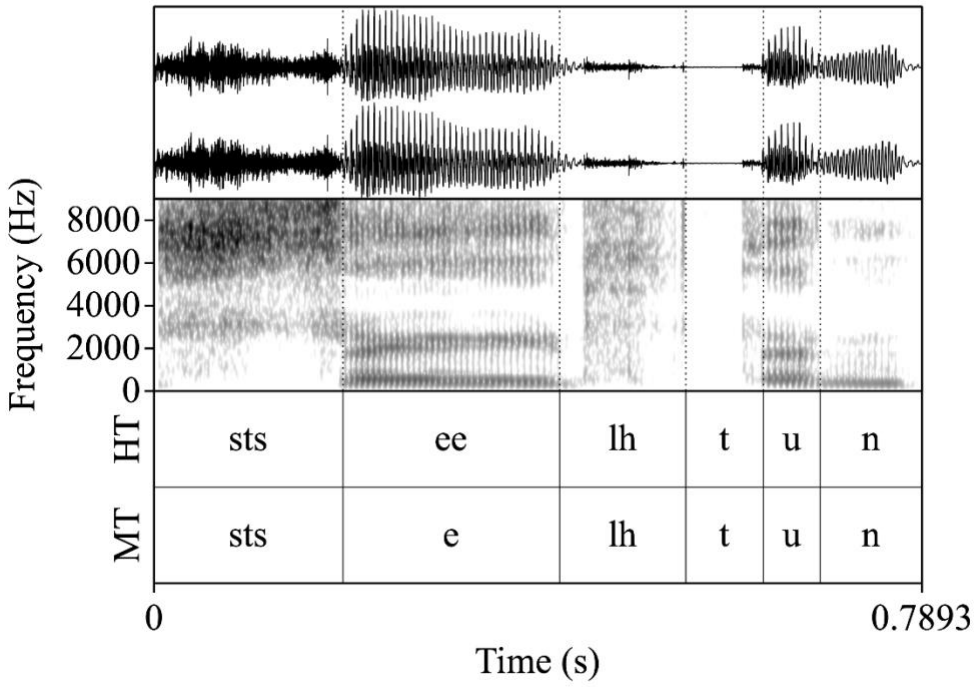
---

Glosses will follow the Leipzig Glossing Rules (Bernard Comrie et al., 2015) and use the standard abbreviations with the addition of LNK for linkers.

labels were not used for other errors because there is no HT available for untranscribed speech and there was no expectation for the model to transcribe codeswitching or rhetorical lengthening in the same manner as the HT, the latter due to the removal of punctuation during data preparation. An example to demonstrate the process of determining error origin is shown in Figure 3.4. This spectrogram of the word <stseelhtun> “salmon” comes from the story “Thunderbird and Orca.” The HT and MT disagree on the length of the first vowel. When I examined the duration of the vowel, I found that it was approximately 200ms, which is within the expected range for long vowels in Hul’q’umi’num’. Thus, this error was classified as OM.

Figure 3.4

Spectrogram of <stseelhtun>

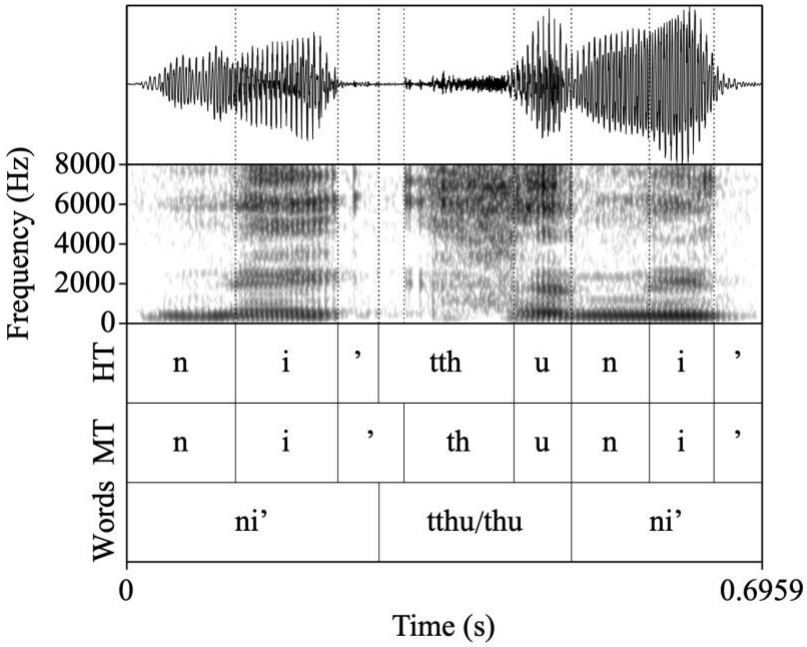


The error origin categorization was sometimes difficult due to the nature of the recording being in a narrative style, making the speech connected and varying in speed. One example of an error that was difficult to categorize was the transcription of <tthu> “the, a” as [thu], as shown in

Figure 3.5. The waveform and spectrogram of the word <tthu> are shown and annotated with the preceding and following words, extracted from a recording of the story “Qiseq-Stoneheads.”

Figure 3.5

Spectrogram of <tthu>



In this example, the glottal stop preceding the mismatched consonant makes determining whether the consonant is an affricate or a fricative difficult. For such cases, I asked Dr. Sonya Bird and Dr. Suzanne Urbanczyk for their opinions on the categorization based on their extensive experience researching Salish languages. The decision was made to categorize this error as OH based on the audio sounding more like a fricative, despite the ambiguous spectrogram.

Chapter 4

Results & Analysis

This chapter covers the results of fine-tuning XLS-R. I begin by reporting the error rates calculated in the testing phase of fine-tuning. The following sections contain analyses of the errors made by the language model. Each of these sections focuses on one of four error types in the following order: mismatch, deletion, insertion, and other. At the end of this chapter, I discuss the resources used for this project. Although XLS-R-ELAN was used in this project, there will be no analysis of it in this chapter because it was used exclusively for demonstration purposes.

4.1 Error rates

The median CER reached 11.1% and the median WER was 50%. CER is calculated by finding the total number of mismatches, insertions, and deletions then dividing by the total number of characters.[17] WER is calculated in almost the same manner: mismatches, insertions, and deletions are added together then divided by the total number of words.[18] Consider the following transcriptions in example (1). (1a) shows the HT with special characters removed and the English translation beneath. (1b) shows the MT.

(1) Little Wren Goes Hunting, line 56

a. *'ii ch 'uw' hiil'e'th 'ul'* (HT)

You are just fooling me.

b. 'i' ch'uw' hiyul'ethul' (MT)

[17] As this is CER as opposed to PER, errors involving one phoneme like <p'> being transcribed as [b] would count as two errors.

[18] For cases in which more than one error has occurred and the errors could be categorized in different ways, the categorizations that produce the smallest number of errors is favoured.

There are two character mismatches: <i> to [‘] in <’ii> and <i> to [y] in <hiil’e’th>. There is also one character insertion of <u>, followed by two character deletions of <’>. Thus, the CER of this sentence would be 5/22, or 22.7%. In comparison, the WER of this sentence would be 100%. Each word containing any character mismatch errors is counted as a mismatch error at the word level. Additionally, segmentation errors—like <hiil’e’th ’ul’> being transcribed as one word—are treated as a combination of a mismatch error plus an insertion or deletion error. While WER is a useful metric for comparing the accuracies of different large language models with each other, CER is a better reflection of how much effort would be needed to correct the MT.

A table of the number of errors (the sum of the times that each phoneme was the HT in mismatch errors, the HT in deletion errors, and the MT in insertion errors), occurrences in the portions of the HT allocated for testing, and occurrences across all of the HT fine-tuning data for each Hul’q’umi’num’ phoneme is shown in Tables 4.1 and 4.2.

Table 4.1

Errors and Frequencies of Phonemes - Consonants

	Phoneme	# of Errors	# in Testing Data	# in all Data
Stops	p	1	6	43
	p’	0	1	14
	t	10	93	910
	t’	3	9	141
	k	0	1	1
	kw	3	32	255
	kw’	2	14	115
	q	1	19	149
	q’	1	7	130
	qw	0	2	49
	qw’	0	5	36
	’	47	159	1153
Affricates	tth	4	17	258
	tth’	3	3	21
	ts	4	25	203
	ts’	2	1	67
	tl’	2	13	116
	ch	1	6	19
	ch’	0	0	0
Fricatives	th	8	27	251
	s	18	111	1066
	lh	2	41	469
	sh	0	14	173
	h	6	21	109
	hw	2	21	238
	x	0	14	121
	xw	0	3	44
Resonants	m	3	33	324
	m’	3	26	301
	n	9	105	813
	n’	1	10	130
	l	2	30	220
	l’	4	17	182
	y	3	20	226
	y’	1	10	119
	w	1	20	114
	w’	4	41	163

Table 4.2

Errors and Frequencies of Phonemes - Vowels

Phoneme	# of Errors	# in Testing Data	# in all Data
i	11	91	873
ii	7	7	34
e	6	77	560
ee	3	3	52
u	20	278	2921
a	4	56	379
aa	4	6	40
oo	0	0	4
ou	0	0	0

The phonemes with the most errors, relative to their frequency in the testing data were /ts'/, /tth'/, and all three long vowels. These phonemes had relatively few occurrences in the fine-tuning data. /t'/, /th/, /'/, and /h/ had the next most errors, proportional to their frequency in the testing data. These phonemes had relatively more occurrences in the fine-tuning data, so their errors may require more detailed analysis.

4.2 Mismatch errors

77 errors (28% of all errors) were mismatch errors. Tables 4.3 and 4.4 display every mismatch error, sorted by the HTs, which are the targets that the language model aims to transcribe, in the leftmost column. The central columns show what the machine had transcribed differently, with the number of occurrences of each mismatch in brackets if there is more than one. The rightmost column shows the total number of mismatch errors for each HT phoneme. As an example, <t> has 93 occurrences in the HT. In four instances where the HT has <t>, the language model transcribed [tth], [ts], [n], and [u]. When viewing the spectrograms of these mismatches, the first looked more like <t>, the second and third appeared closer to [ts] and [n],

and the fourth neither looked like <t> nor [u]. Thus, the first is placed in the OM Errors column, the second and third in the OH Errors column, and the fourth in the OU Errors column.

Table 4.3

Mismatch Errors - Consonants

	HT	OM Errors	OH Errors	OU Errors	Total Errors
Stops	p				0
	p’				0
	t	tth	ts, n	u	4
	t’	tl’	t, ts’		3
	k				0
	kw		kw’		1
	kw’	kw	k		2
	q		’		1
	q’		’		1
	qw				0
	qw’				0
	’	l’			1
Affricates	tth	th (2)	th (2)		4
	tth’	th			1
	ts	ts’, s			2
	ts’	ts			1
	tl’	t’			1
	ch		sh		1
	ch’				0
Fricatives	th	tth (2), s	tth, s		5
	s	sh (3), x			4
	lh	th			1
	sh				0
	h	t’, ’, th			3
	hw	qw, h			2
	x				0
	xw				0
Resonants	m	m’ (2)	m’		3
	m’	m	m (2)		3
	n	s			1
	n’				0
	l		l’		1
	l’	’	’, l		3
	y				0
	y’				0
	w				0
	w’	’	w		2

Table 4.4

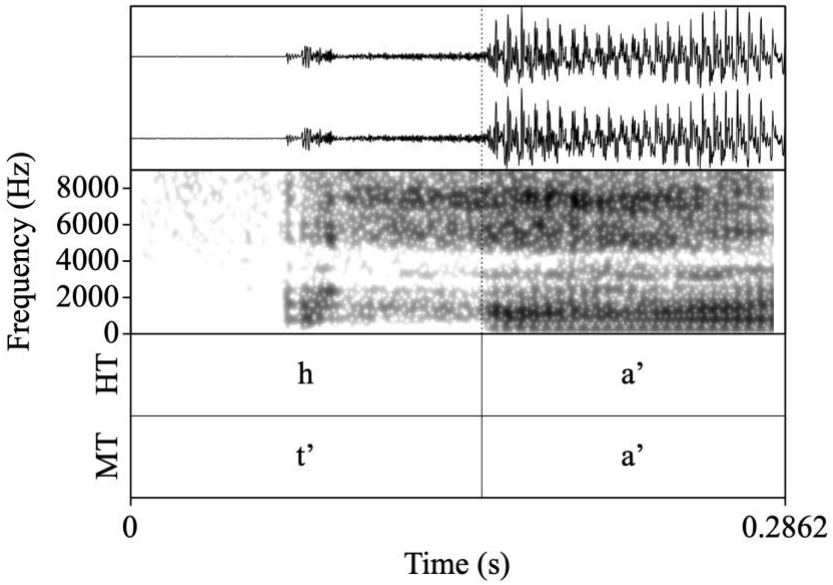
Mismatch Errors - Vowels

HT	OM Errors	OH Errors	OU Errors	Total Errors
i		u (2)	y	3
ii	i (4)	i (3)		7
e	i, u	u		3
ee	e (3)			3
u	i, e (2)		i	4
a	u	aa		2
aa	a	a (3)		4
oo				0
ou				0

The consonant mismatches often involved one phonetic feature at a time. There were several instances of plain consonants being transcribed as their glottalized counterparts, dental and alveolar sounds being transcribed as each other, and fricatives being transcribed as affricates and vice versa. There were three errors of <s> being transcribed as [sh]. Two of these mismatches occurred word-initially and one occurred word-finally. Two unexpected mismatches—the transcriptions of <t> as [n] and [u]—were not OM errors, meaning that the language model transcribed accurately to the acoustic signal in these. An unexpected OM mismatch also occurred when <h> was transcribed as [t’] in the story “Thunderbird and Orca.” The spectrogram for this mismatch in the word <ha’> “if, when” is shown in Figure 4.1.

Figure 4.1

Spectrogram of <ha'>



Although the spectrogram of the consonant has the light frication that is typical of /h/, the burst-like activity does make the consonant look like a word-final ejective if the frication was interpreted as the glottal release. While not fully supported by the waveform and spectrogram, especially because ejectives typically have silence between the release and vowel, there is some justification for the transcription of [t'].

The vowel mismatch errors in Table 4.4 show that long vowels were often transcribed as their short counterparts, but the inverse only happened once (<q'**a**ythamu> "kill you" as ['**aa**ythamu]) and was categorized as an OH error due to the vowel being almost 200ms in duration. 73% of long-short vowels mismatches had durations within 20ms of either 100ms or 200ms, which made the error origin categorization relatively simple. For the remaining long-short vowel mismatches, 150ms was chosen as the cutoff. Of the 16 occurrences of long vowels in the HT, only 2 were transcribed in the MT. However, long vowels had far fewer occurrences in

comparison to short vowels. The short vowel errors lacked much of a pattern, besides being transcribed as other short vowels. Every other short vowel has been transcribed as <u> at least once, which could be reflective of the large vowel space that <u> occupies.

Many phonemes with low occurrences in the fine-tuning data—such as /p/ and /l/—were still transcribed with high accuracy by the language model. This would be highly unexpected for many language models because sounds that are infrequent in training data are likely to be transcribed with lower accuracy (Kleynhans & Barnard, 2015) but since XLS-R is pretrained, sounds that are infrequent in the fine-tuning data yet transcribed quite accurately are likely frequent in the pretraining corpora.

To better examine the consonant errors in Table 4.3, they are grouped based on how the HT and MT differ in Figure 4.2. Table 4.5 provides examples for the consonant error differences used to make Figure 4.2. If the HT and MT differ in more than one way, the difference is counted for each category. Debuccalization refers to the transcription of a glottalized consonant in the HT as a glottal stop in the MT. Buccalization is the inverse and is grouped with debuccalization. The debuccalization category was made to separate these errors from the place errors, in which supralaryngeal place is swapped. The “class” label refers to when a vowel is transcribed as a consonant and vice versa.

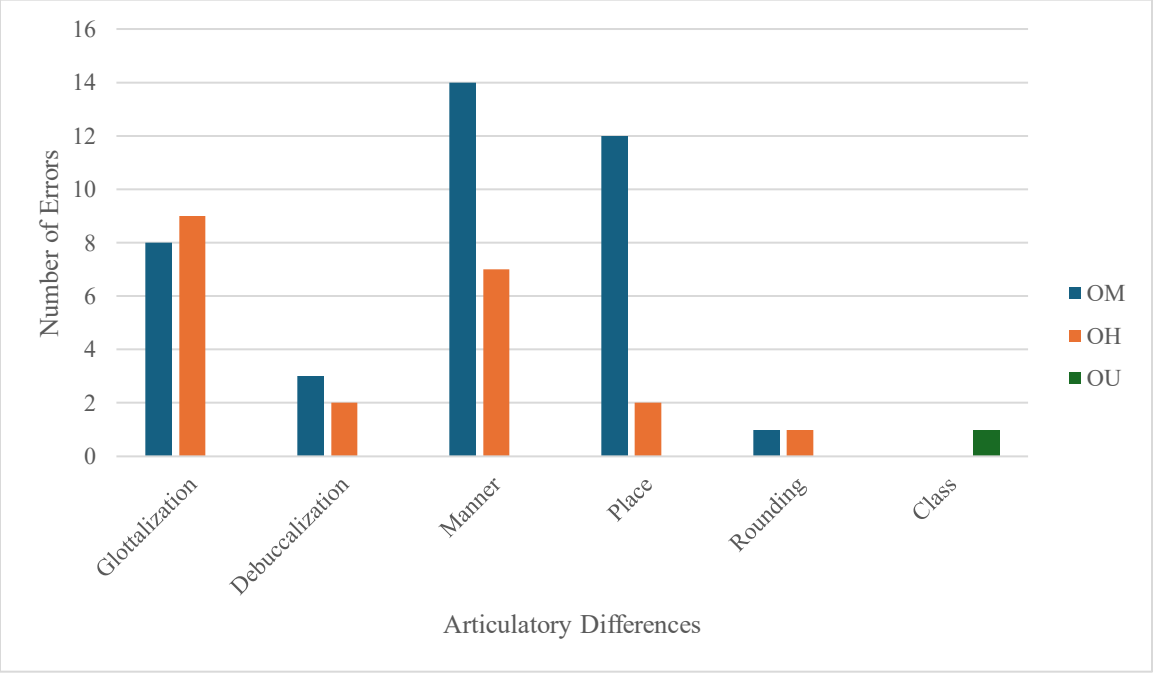
Table 4.5

Examples of Consonant Mismatch Error Difference Groupings

	HT	MT
Glottalization	sqw'u**l**qw'ulesh "birds"	sqw'u**l'**qw'ulesh
Debuccalization (and buccalization)	si**l'**anum "year"	xi**'**anum
Manner	**th**u "the, a"	**tth**u
Place	**s**xuxits "obvious, visible"	**sh**xuxitsul'
Rounding	**hw**kw'atus[19]	**h**kw'atus
Class	sew'q'**t** "looking for"	sew'q'**u**

Figure 4.2

Mismatch Differences - Consonants



[19] Likely <hwkw'at> "pull it, pull the slack up" with third person -us suffix.

The OM and OH errors were very similar for each category, except for manner and place, which were much higher for OM errors. Four debuccalization errors occurred, while buccalization only occurred once (<**ʼ**a'luxutus>[20] as [**l'**a'luxutus]).[21] The mismatches of glottalized consonants for glottal stops and vice versa mostly occurred with glottalized resonants with only one error involving an ejective (<**q'**aythamu> "kill you" as [**ʼ**aaythamu]). 48% of the manner errors involved the sounds /th/ and /tth/, mostly transcribing one as the other. This could be due to the speaker, Dr. Peter, having quite lenited /tth/ when speaking, "meaning that either the stop or fricative portion is very short and subtle" (S. Bird, personal communication, March 15, 2026). 86% of the errors swapping /th/ and /tth/ occurred word-initially. Three of four instances of <tth> being transcribed as [th] and one instance of <tth'> being transcribed as [th'][22] occurred following a glottal stop or glottalized resonant. The high number of errors and specific environments of /th/ and /tth/ mismatch errors suggests that there might be allophonic variation occurring. In 82% of glottalization errors, the transcriptions differed only in glottalization, making this the most likely error to make alone. 35% of the glottalization errors were transcribing <m> as [m'] and vice versa. 83% of /m/ and /m'/ mismatches happened word-finally. All three instances of <m'> being transcribed as [m], as well as one case of the inverse, preceded a word-initial glottal stop in the following word (e.g. <hunu**m'** 'utl'> as [uhwunu**m** 'utl']). Mismatches in rounding were low. It was extremely rare for consonants to be transcribed as vowels and vice versa. There was only one instance of a vowel being transcribed as an approximant (<m'**i**> "come" as [m '**y**u]), despite the similarity of the sounds.

[20] Likely <'aluxut> (English: collect, gather, select) (Gerdts et al., 1997) with third person -us suffix.
[21] There was no preceding sound causing this error.
[22] This was categorized as two errors (a substitution and insertion) because [th'] is not a phoneme in Hul'q'umi'num'. The formatting used here is for illustrating the similarity to the HT.

4.3 Deletion Errors

Of the 70 deletion errors—which comprise 25% of all errors—only 27% were OM errors compared to 64% OH errors. A full breakdown of deletion errors, sorted by how frequently each phoneme was deleted, is available in Table 4.6. For this table, deletions of multiple phonemes in a row were treated as separate. As mentioned in Section 4.2, character deletions that could be interpreted as phoneme substitutions were categorized as substitutions. All errors categorized as glottal stop deletions occurred in positions where the apostrophe could not indicate a glottalized consonant (e.g. <‘ul’> “just” to [ul’]).

Table 4.6

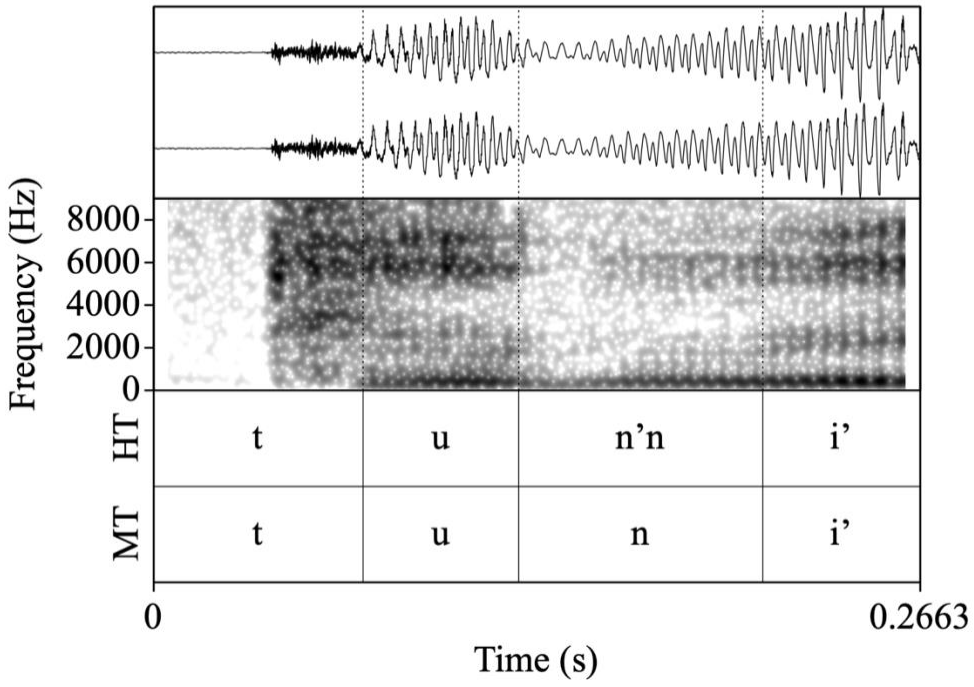
Deletion Errors

HT	OM Errors	OH Errors	OU Errors	Total Errors
’	5	20	2	27
s	1	5	1	7
u	3	3	1	7
i	1	2	1	4
t	1	2		3
n	1	2		3
kw	1	1		2
tth’	2			2
ts		1	1	2
h		2		2
p		1		1
tl’	1			1
th		1		1
e		1		1
a	1			1
n’		1		1
l’		1		1
y	1			1
y’	1			1
w		1		1
w’		1		1

The frequency of OH deletion errors suggests that the human(s) tended to transcribe more than what was present in the audio. This is standard for many forms of transcription (e.g. Clean Read and Clean Verbatim of the Amberscript transcription types (Amberscript, 2025)) because sounds are often deleted in connected speech—particularly during fast speech, which is utilized in narratives. In general, human transcribers may add sounds that are not present in the audio when they know the standard forms of the words they hear. An example of this type of error is shown in Figure 4.3, which is taken from the recording of "Thunderbird and Orca." The HT is <tun'ni'> "from" and the MT is <tuni'>. The MT is a closer match because the glottalization is not visible in the resonant and seems to have moved to the following vowel.

Figure 4.3

Spectrogram of <tun'ni'>



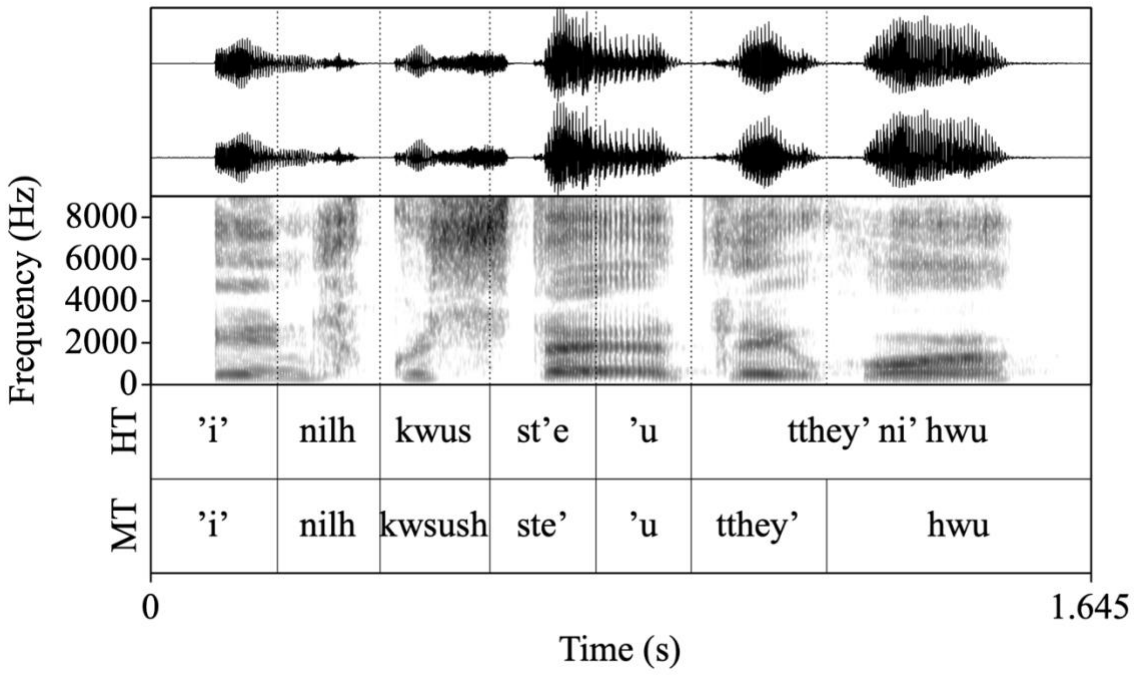
OH deletions of glottal stops were the most common. The majority of these glottal stop deletions occurred in short words known as linkers (Baetscher, 2014; Webb, 2025), such as <'uw'>. The initial glottal stop of <'uw'> as well as the initial word boundary were almost always deleted, merging it with the preceding word. For example, there were three cases of <'uw'> combining with <sus> to create [susuw']. Additionally, 59% of all glottal stop deletions occurred word-initially before a vowel, and 75% of all glottal stop deletions in this environment were OH errors. <s> was the second most frequently deleted. All but one <s> deletion occurred word-initially and all but one word-initial <s> deletion was an OH error. The word-initial <s> deletions occurred in the words <sutst>,[23] <si i is> "and", <st'i'am'> "stick to: on, stuck on, fastened", <stl'is>,[24] and <suw'>,[25] with the same error happening in two occurrences of <sutst>. The word-initial deletion of <st'i'am'> was categorized as OU because the preceding word ended in <s>. Word-initial <s> deletion may also occur due to it being a common prefix. It appears to be a prefix in <sutst> and may function like a prefix as well in <sis> (Gerdts et al., 1997).

There are three factors that may have influenced <u> deletion. Firstly, because it is a schwa—which is typically shorter than other vowels—it was sometimes deleted when a neighbouring consonant was deleted (e.g. <**su**tst> as [ts't]. <i> experienced the same manner of deletion as well, in the word <ni'> in the sentence shown in Figure 4.4 from "Thunderbird and Orca."

[23] Perhaps a prefix with <tst> "we"
[24] Seems to be <stl'i'> "want, desire, like" (Gerdts et al., 1997) with a suffix..
[25] Uncertain definition.

Figure 4.4

Spectrogram of <’i’nilh kwus st’e ’u tthey’ni’hwu>

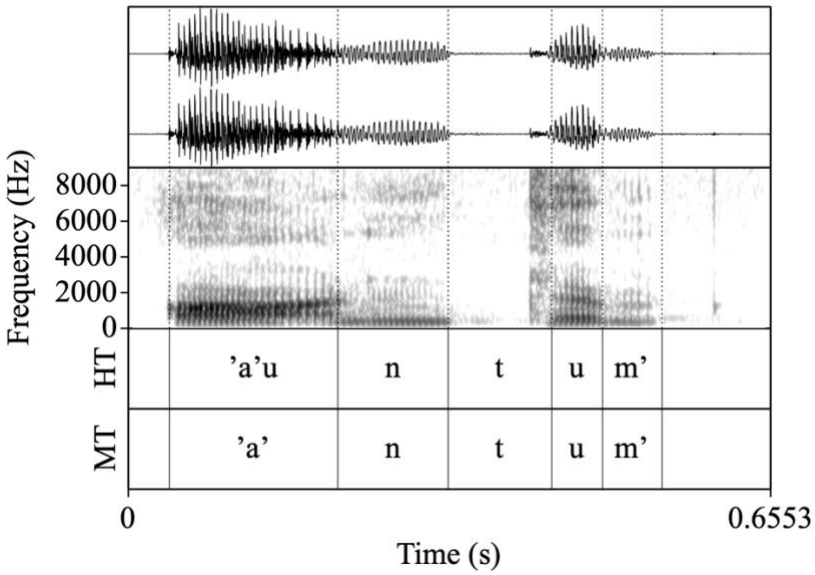


Note. “So it was like that—“

Secondly, the language model may have difficulty detecting <u> when it is next to a rounded consonant since it was frequently deleted in this circumstance (e.g. <shqwal**u**wun> “feelings, thoughts” as [shqwalwuns]). The frequency of deletion errors in this environment could also be due to the high likelihood of <u> occurring next to a rounded consonant (31%). Lastly, three of the four <u> deletions that happened in isolation occurred when the word contained a V’V structure with one of the vowels being <u>: <’**a’u**ntum’>, <kw**u’e**lh>, and <’**e’u**hwiin’>. Their spectrograms are shown in Figures 4.5 to 4.7. Figure 4.6 compares two spectrograms of <kwu’elh> with the left showing <u> deletion and the right showing <e> deletion. There were no other deletions of vowels in V’V structures.

Figure 4.5

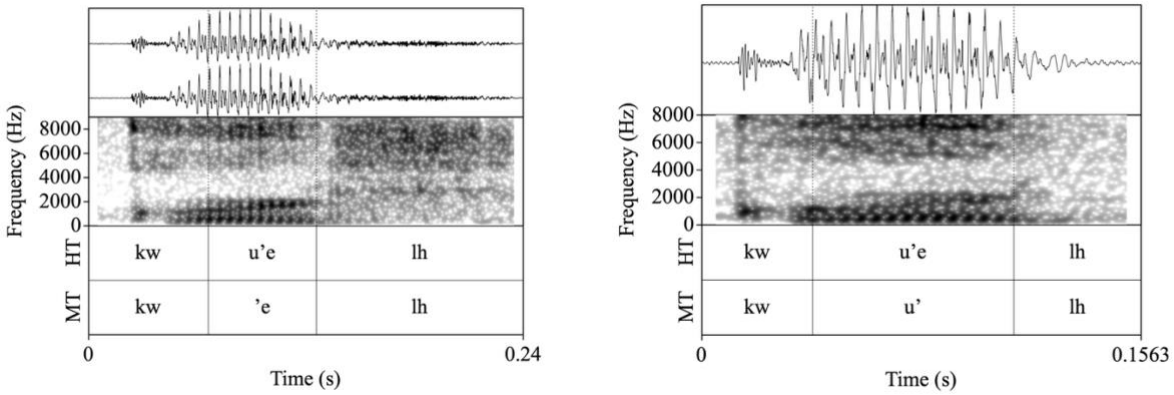
Spectrogram of <’a’untum’>



Note. Perhaps <’a’un’t> “letting him/her do it; get his own way” with a suffix. From “Thunderbird and Orca.”

Figure 4.6

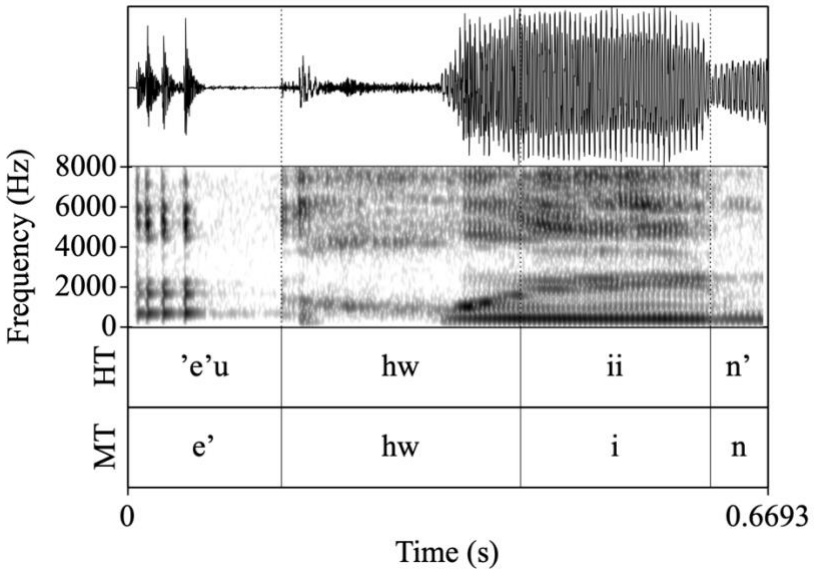
Spectrograms of <kwu’elh’>



Note. “Indeed.” From “Thunderbird and Orca.”

Figure 4.7

Spectrogram of <’e’uhwiin’>



Note. “Little, small, tiny.” From “Q’ise’q and the Stoneheads.”

In the spectrograms of <’a’untum’> and <kwu’elh>, the V’V segments look like diphthongs with a slight rise in the F2 and little else differentiating the vowels. There is no visible glottal stop. The two spectrograms of <kwu’elh> are nearly identical, despite being transcribed differently by the model. In <’e’uhwiin’>, there does not appear to be any change in the creaky <e> that would indicate the presence of <u>. This could indicate that <u> tends to assimilate to the other vowel in V’V segments in Hul’q’umi’num’.

4.4 Insertion errors

The insertion errors—shown in Table 4.7—totaled to 62, taking up 22% of the errors. Insertions of multiple phonemes in a row were counted separately, as with Table 4.5.

Table 4.7

Insertion Errors

MT	OM Errors	OH Errors	OU Errors	Total Errors
’	13	5	1	19
u	5	2	2	9
s	5	1	1	7
n	3	1	1	5
i	4			4
t	3			3
a		3		3
th	1	1		2
y	1	1		2
e	1	1		2
ts’	1			1
h		1		1
lh	1			1
l		1		1
w’	1			1

Similarly to the deletion errors, the most insertion errors were made with glottal stops. However, most of the glottal stop insertion errors were OM errors, meaning that the language model was over-transcribing glottal stops. Glottal stop insertion generally happened word-medially, with the exception of two cases in which the insertion occurred word-finally before a word with an initial glottal stop (e.g. <slheni ’i’> “woman and” as [slheni’ ’i’])[26] and two instances of the insertion reflecting glottalization throughout the vowel (e.g. <’ii ch ’uw’> “AUX.Q you LNK” as [’i̱ ch’uw’]).

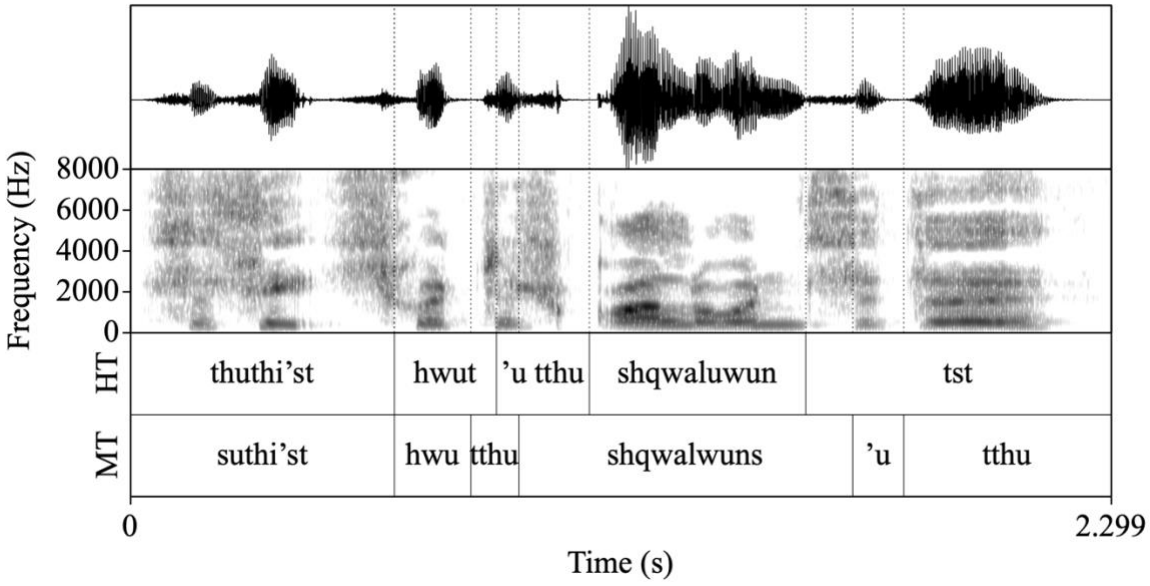
[s] appears to be inserted more frequently than other consonants, but half of these insertions were transcriptions of <kwus> “he, she, it, that he, when he, as he, etc.” as [kwsus]. The remaining [s] insertions were OH errors and one instance where the model could have either detected the frication in the following word or made a prediction based on previous occurrences

[26] <slheni’> is an acceptable spelling (Gerdts et al., 1997), which could have been recognized by the model.

of the word with <s> (<shtun'naalhtun xinupsum>[27] as [shtun'nalhtuns xinupsum]). A large portion of insertion errors appear to result from the language model recognizing which words are used frequently and over-transcribing them. In Figure 4.8, this over-transcribing can be seen in how <tst> "we"—which occurs 2 times in the testing data—is transcribed as ['u tthu]—two words that occur 8 times each in the testing data—with mismatches and insertions.

Figure 4.8

Spectrogram of <thuthi'st hwut 'u tthu shqwaluwun tst>



Note. "And this is how we were taught how to be as we walked this earth to like these stories."[28] From "Little Wren Goes Hunting," line 105.

This is especially noticeable when examining the insertion errors and segmentation errors together.

[27] Looks like a prefix plus <tun'naalhtun> "these people" and <xinupsum> "Green Point, Kinupsem."

[28] The HT formed a sentence with several other breath groups. The translation for the entire sentence was provided because Webb (2022) was uncertain about how to divide the English translation when adding the time-alignments.

4.5 Segmentation and other errors

There were 62 segmentation errors, which accounted for 22% of the total errors. 76% of segmentation errors were combining errors, in which separate words are transcribed as one word. Splitting errors were much less common, likely due to how connected speech often lacks pauses between words, which makes combining errors expected for ASR systems. The combining errors had very little in common outside of occurring in the absence of a pause. On the other hand, splitting errors almost always occurred with insertion errors and generated high-frequency words. The list of words used to compare with the insertion and segmentation errors was the "Hul'q'umi'num' 175 Little Words" document (HLCS, 2021). One example of a splitting error is the transcription of <'uwu> "no, not" as ['u wulh]. This split takes <'uwu>, separates one half into the oblique marker or question particle <'u>, and inserts a final consonant onto the other half to form the temporal marker <wulh>.

The 9 remaining errors, about 3% of all errors, were placed into subcategories: rhetorical lengthening, codeswitching, untranscribed speech. Rhetorical lengthening is a type of lengthening used in Hul'q'umi'num' storytelling that is typically transcribed by repeating the lengthened vowel with periods between them. For example, the word <thi> "big" when used with rhetorical lengthening would be transcribed as <thi.i.i>. When processing the fine-tuning data, periods were removed along with most other punctuation marks, as they would introduce unnecessary elements in the training. Thus, transcriptions with rhetorical lengthening look like <thi i i> in the fine-tuning data. Since there were very few cases of rhetorical lengthening in the data, there was no expectation that they would appear in the MT. Despite this, in the testing data, three of the six transcriptions of rhetorical lengthening in the MT contained repeated vowels separated by a space. The example <thi.i.i> was transcribed as [thi i]. There were two cases of

codeswitching errors, in which the speaker used English, causing the MT to create an approximation of the word using the Hul'q'umi'num' orthography. These were the transcription of <satellite island> and the following word <ni'> as [setley't'ayluni'] and <lagoon> as [lukwon]. Untranscribed speech was not necessarily an error, but a difference caused by the HT not transcribing a portion of the audio because the speaker corrected herself afterwards. There was only one case of this occurring.

4.6 Resource requirements

As one of the goals of this project is to aid the HLCS in determining whether training a language model would be a good use of resources, the resource requirements must be discussed. The automated portions of the fine-tuning process, which was running Dr. Coto-Solano and von Platten's code, took 48 minutes, making the entire process, including changing portions of the code to use the correct folders and files, approximately one hour long. The process used 5.2 GB of System RAM and 8.9 GB of GPU RAM. The Colab notebooks used the default Intel Xeon CPU with one CPU core. Using the Green Algorithms Calculator by Lannelongue et al. (2021), the computational resources used for this project can be contextualized as using 453.26 mg of carbon dioxide, which is equivalent to 0.00259 km in a passenger car or 0.0004% of a flight from Paris to Dublin.

Chapter 5

Discussion

This chapter begins by comparing the results of the fine-tuning with the accuracy of other XLS-R models. I then examine the error patterns listed in Chapter 4 against research on Hul'q'umi'num' phonetics and phonology to investigate the potential causes of these patterns as well as what the errors may suggest about patterns in Hul'q'umi'num' speech and orthography. I continue with discussing how useful XLS-R may be at transcribing other Indigenous languages when considering the resources used for this project and the skills required. Lastly, I summarize factors that were examined in this project and how if may affect the ability of XLS-R to transcribe other languages.

5.1 The Accuracy of XLS-R for Hul'q'umi'num'

My first research question asked how well XLS-R performs on Hul'q'umi'num' and therefore how useful it might be in the transcription process. When considering the amount of fine-tuning data that was used, the XLS-R language model that I fine-tuned on Hul'q'umi'num' transcribed speech achieved a reasonable CER of 11.1% and WER of 50%. This CER was better than the CERs found by Conneau et al. (2020) on out-of-pretraining languages, which averaged 18.7%. Conneau et al.'s WERs were similar but slightly better with a range of 31.1% to 44.1%. Babu et al. (2021), when evaluating XLS-R by fine-tuning several models on one hour of read speech in different languages, found that the PER ranged from 2.2% to 14.8% with a mean of 4.9%. Although the CER in the Hul'q'umi'num' model is higher than the average PER found by Babu et al., their models used more fine-tuning and pretraining data. The same can be said about Coto-Solano et al. (2022), who achieved around 22.2% WER and 6.1% CER with almost four hours of fine-tuning on Cook Islands Māori data. When compared with Chen et al. (2023), who

found an average CER of 43.4% with ten minutes of fine-tuning and 36.8% with one hour of fine-tuning on Quechua, Bribri, Guarani, Kotiria, Wa'ikhana, and Totonac, the CER of the Hul'q'umi'num' model was much lower. Overall, the fine-tuned Hul'q'umi'num' model achieved better accuracy than previously evaluated models but further improvement is likely possible with more pretraining and fine-tuning data, which indicates that it will likely be able to support transcription efforts for the HLCS.

5.2 Error Analysis Implications for Hul'q'umi'num'

My second research question asked if the the errors made by XLS-R when transcribing Hul'q'umi'num' can teach us anything about Hul'q'umi'num' phonology and the discrepancy between phonemic (and orthography) forms and phonetic realization. Many of the patterns found in the errors made by XLS-R can be attributed to features of Hul'q'umi'num' that have been previously documented, but some could point to where further research can be done. One example of an error pattern that could point to an area for future research was the model's tendency to transcribe long vowels as short vowels. This error happened far more than short vowels being transcribed as long. There were very few occurrences of long vowels in the fine-tuning data, which could mean that there simply were not enough samples for the model to learn the vowel length distinction. However, the model was able to transcribe half of six occurrences with rhetorical lengthening, showing that the model was able to detect a difference in transcription caused by duration using very few samples. Furthermore, many of the vowel length errors were OH which could indicate that the duration of long vowels is more variable or has a lot of overlap with the duration of short vowels. More detailed analysis of vowel duration could lead to a more comprehensive description for when and why this variation occurs.

There were several mismatch, deletion, and insertion errors involving <u>,[29] which is the orthographic form of /ə/ in the International Phonetic Alphabet (IPA) and American Phonetic Alphabet (APA). Every short vowel that is not <u> had at least one instance of being transcribed as <u>, which is likely in part due to schwa occupying a particularly large vowel space and partly due to co-articulatory effects from adjacent consonants. Cross-linguistically, vowels labelled as schwa take up a large portion of the vowel space, gravitating to the centre (J. Leonard, 2019; Nolan, 2017). In Hul'q'umi'num', /ə/ occurs in free variation with /o/ before /w/, with /ɔ/ before /x^w q^w q^w'/, with /ɨ/ after /y/, and as /ʌ/ in unstressed positions (Kava, 1969). Vowels in unstressed syllables are also reduced to schwa (Gerdts, 2014). A more standardized and detailed examination of the vowels involved in the mismatch errors is necessary to draw further conclusions about the cause.

There appeared to be three common environments for schwa deletions: when a neighbouring consonant was deleted, when schwa occurred next to rounded consonants, and when schwa and another vowel were in a V'V structure. According to Gerdts & Werle (2014), Hul'q'umi'num' lacks vowel-vowel sequences and such sequences are resolved by inserting a glottal stop between the vowels, deleting a vowel, or contracting (or coalescing) the two vowels. The spectrogram of <'e'uhwiin> in Figure 4.7, showing a glottal stop and a vowel deletion, demonstrated that multiple strategies for resolving vowel-vowel sequences may be used by a speaker at the same time. The language model's consistency with transcribing the glottal stop despite it not being present in most of the V'V spectrograms suggests that the glottal stop insertion is consistent enough in the HT that the model was able to recognize a strong pattern.

[29] In this chapter, angle brackets (<>) are used for the orthographic form and slashes (//) are used for IPA notation unless specified otherwise.

The model's inconsistency with transcribing <u> in this structure when it seems to be present in the spectrogram could mean that the model has learned from instances of <u> being present in the audio but not in the HT while near another vowel and a glottal stop. Determining whether this explanation is correct would require more detailed examinations of <u> in more samples of V'V, u'V, and V'u structures.

Glottal stops and glottalized consonants were frequently involved in every type of error. The model transcribing glottalized consonants—mostly glottalized resonants—as glottal stops happened much more than the inverse. Glottal stops were also the phonemes that were most likely to be inserted or deleted in an error. Humans tend to transcribe glottal stops more than they occurred in the audio, leading to the relatively high number of OH errors. This was expected because although there is an abundance of glottal stop deletion in Hul'q'umi'num' connected speech (Marshall, 2018), human transcribers have knowledge of the underlying forms of morphemes and transcription standards. When the speaker is not producing a glottal stop, transcribers may still transcribe a glottal stop to maintain the readability of the transcription. Glottal stop deletion is particularly noticeable in short words like <'uw'> and <'i'>, known as *linkers*. Linkers join elements in multiple syntactic constructions: conjunction, subordination, and modifier phrases (Webb, 2025). Linkers can exhibit reduced glottalization as a result of integration with words around them (Gerdts & Werle, 2014) and this can be affected by the location of prosodic boundaries, which could be affected by syntactic and discourse structure. Webb's (2025) work on Hul'q'umi'num' linkers examined pitch resets and pausing surrounding <'uw'> and <'i'> to determine their relation to prosodic boundaries and found that the linkers exhibit pitch reset differences that could be explained by glottalization, and that pitch reset and pausing can provide seemingly contradictory information about the linkers' relation to prosodic

boundaries. These linkers are also prominent in segmentation errors made by XLS-R, which will be discussed more in the following paragraph. The present work and Webb's research both point to a need for further investigation into the interaction of factors affecting the production of linkers in Hul'q'umi'num'. Finally, errors in which glottalized consonants were transcribed as plain consonants and vice versa, particularly <m> and <m'>, were common. This is also a frequent error for L2 Hul'q'umi'num' learners (Bird et al., 2021). Errors of transcribing <m>[30] as [m'] mostly occurred word-finally, before a word-initial glottal stop in the following word. This particular error shows an even split of OH and OM errors, suggesting that the speaker may not be glottalizing the resonant in this environment and that the model may have difficulty recognizing cases of double-glottalization.

A number of errors involving supra-glottal articulations were also present in the transcriptions. In particular, <th> and <tth> were frequently transcribed as each other word-initially and following glottal stops or glottalized resonants. This is likely due to the closure period of affricates being very short in Dr. Peter's speech, making it hard to distinguish <tth> from <th> (S. Bird, personal communication, March 25, 2026). The frequency of <s> deletion was also common and tended to be word-initial and match the spectrogram, suggesting that either word-initial <s> deletion is frequent in Hul'q'umi'num' or that <s> is shortened or lenited enough word-initially that it is sometimes not picked up in recordings. <s> insertion consistently happening to form the word <kwsus> shows XLS-R's bias for transcribing high-frequency words. XLS-R's insertion patterns and over-transcription of high-frequency words happen alongside its numerous segmentation errors, often forming linkers. The segmentation errors could be related to Hul'q'umi'num' prosody and, more generally, how defining what a word is in

[30] HT.

Salish languages can be complicated. For example, Beck (1999) argued using data from Lushootseed that "the phonological word differs markedly from what can reasonably be called a word in the morphosyntax even in languages that are only mildly polysynthetic, and that what is called a word in the syntax may not be a word in the phonology." Future research analyzing how XLS-R is segmenting Hul'q'umi'num' words could lead to new insight on these topics.

5.3 The Usefulness of XLS-R for Other Languages

To answer my third research question, which was regarding how useful XLS-R would be for people working to document other languages, the requirements for fine-tuning have to be considered. The required knowledge of programming, time required for training, and computational resources for training XLS-R were all very low. Most language communities would be able to fine-tune their own XLS-R model easily, using Dr. Coto-Solano and von Platen's code. Additionally, only 26 minutes of transcribed speech was necessary to achieve relatively good accuracy for Hul'q'umi'num'. The amount of transcribed speech necessary to achieve the same accuracy may be different for other languages because of the factors described in Section 5.4. Considering that the cost of fine-tuning an XLS-R model is so low, doing so is likely to be beneficial for most communities that are interested in having their recordings transcribed. When the results were shown to Dr. Gerdts, she expressed interest in doing further fine-tuning with more selective datasets, which has been shown to be effective for improving accuracy (Kleynhans & Barnard, 2015). More research is needed to assess how well the fine-tuned model can transcribe recordings that are not in its fine-tuning dataset, including recordings of different speakers. How much tools like XLS-R-ELAN (Cox, 2023) can increase the accessibility of language models warrants investigation as well. Further research into Hul'q'umi'num' speakers' perceptions of the generated transcriptions would also be necessary to

determine how useful the transcriptions are when considering the difficulty of correcting the transcriptions. For example, the inability of XLS-R to transcribe long vowels in Hul'q'umi'num' accurately could make correcting the generated transcriptions more difficult as determining vowel length may be harder for human listeners (S. Bird, personal communication, March 15, 2026). While a more experienced Hul'q'umi'num' speaker can lexically identify long vowels (S. Bird, personal communication, May 28, 2026), a learner relying on acoustic cues may struggle to correct this type of error.

5.4 Factors Affecting XLS-R Accuracy

The fourth research question asked what might cause XLS-R to transcribe languages that it has not encountered in pre-training with higher or lower accuracy. The errors support that languages with a more transparent orthography like Hul'q'umi'num' may be transcribed more accurately by a fine-tuned XLS-R model than languages with a less transparent orthography. Vowels like schwa overlapping with other vowels can create mismatch errors. Human transcriptions of sounds that are not present in the audio like glottal stops in the fine-tuning data can increase errors of all types, although the model is able to make predictions about where transcriptions of sounds that are not present in the audio may be desired. The segmentation errors and high WER support the idea that relatively more morphologically complex languages may be transcribed less accurately than less morphologically complex languages. The insertion of high-frequency words and tendency to combine multiple words are key aspects to pay attention to when assessing whether XLS-R will be useful for transcribing a particular language or correcting the errors made by a fine-tuned model. The number of individual sounds and segments in the fine-tuning data may have less of an effect on transcription accuracy than expected since rhetorical lengthening was extremely rare but transcribed in half of its occurrences, likely due to

the extreme difference in duration from other vowels. To conclude, XLS-R will produce transcriptions with differing accuracy depending on the target language's phonology, morphology, and other discrepancies between the written and spoken forms. Where another language is similar or different from Hul'q'umi'num' may provide a rough estimate as to how accurately XLS-R can transcribe the other language.

Chapter 6

Conclusion

For this project, I fine-tuned an XLS-R model to transcribe Hul'q'umi'num'. The CER was quite low for only 27 minutes of fine-tuning data and the WER was higher due to segmentation errors, which should be easier to fix manually if the phonemes are transcribed well enough that the words are recognizable. The errors remaining in the generated transcriptions can be used to create pedagogical tools by having learners practice identifying and fixing the errors. Many of the machine transcription errors can be explained by the features of Hul'q'umi'num', particularly errors involving glottalized resonants, ejectives, glottal stops, schwas, long vowels, and fricatives. This research shows that XLS-R can be used by language communities with low amounts of transcribed audio. The fine-tuning requires some familiarity with Google web applications and computer software such as ELAN but minimal experience with programming.

References

Adams, O., Cohn, T., Neubig, G., Cruz, H., Bird, S., & Michaud, A. (2018). *Evaluating phonemic transcription of low-resource tonal languages for language documentation*. 3356–3365. https://shs.hal.science/halshs-01709648

Alammar, J., & Grootendorst, M. (2024). *Hands-on large language models: Language understanding and generation* (1st ed.). O'Reilly Media.

Amberscript. (2025). *Transcription guidelines (2025): Taxonomy, language-specific features, and positioning in research & practice*. (v2025.11-1). https://doi.org/10.5281/zenodo.17700610

Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., Platen, P. von, Saraf, Y., Pino, J., Baevski, A., Conneau, A., & Auli, M. (2021). *XLS-R: Self-supervised cross-lingual speech representation learning at scale* (arXiv:2111.09296). arXiv. https://doi.org/10.48550/arXiv.2111.09296

Baetscher, K. M. (2014). *Interclausal and intraclausal linking elements in Hul'q'umi'num' Salish* [Master's Thesis]. Simon Fraser University.

Baevski, A., Conneau, A., & Auli, M. (2020). *Wav2vec 2.0: Learning the structure of speech from raw audio*. https://ai.meta.com/blog/wav2vec-20-learning-the-structure-of-speech-from-raw-audio/

Beck, D. (1999). Words and prosodic phrasing in Lushootseed narrative. In T. A. Hall & U. Kleinhenz (Eds.), *Studies on the phonological word* (1st ed., pp. 23–46). John Benjamins Publishing Company. https://doi.org/10.1075/cilt.174.03bec

Bernard Comrie, Martin Haspelmath, & Balthasar Bickel. (2015). *The Leipzig glossing rules: Conventions for interlinear morpheme-by-morpheme glosses*. The Department of

Linguistics of the Max Planck Institute for Evolutionary Anthropology & The Department of Linguistics of the University of Leipzig. https://www.eva.mpg.de/lingua/resources/glossing-rules.php

Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, *56*, 85–100. https://doi.org/10.1016/j.specom.2013.07.008

Bird, S. (2020). Sparse transcription. *Computational Linguistics*, *46*(4), 713–744. https://doi.org/10.1162/coli_a_00387

Bird, S., Claxton, R. A., & Percival, M. (2023). *Seeing speech: Using Praat to visualize Hul'q'umi'num' sounds*. *17*, 297–324.

Bird, S., Leonard, J., & Nolan, T. (2021). *Pronunciation patterns among L2 Hul'q'umi'num' learners*. https://www.semanticscholar.org/paper/Pronunciation-patterns-among-L2-Hul%E2%80%99q%E2%80%99umi%E2%80%99num%E2%80%99-Bird-Leonard/0f907c7ddd0b8a0ff6ade8498258ad46190cf626?p2df

Boersma, P., & Weenink, D. (2024). *Praat: Doing phonetics by computer* (Version 6.4.23) [Computer software]. https://www.praat.org

Chen, C.-C., Chen, W., Zevallos, R., & Ortega, J. E. (2023). *Evaluating self-supervised speech representations for Indigenous American languages* (arXiv:2310.03639). arXiv. https://doi.org/10.48550/arXiv.2310.03639

Chen, W., Zhang, W., Peng, Y., Li, X., Tian, J., Shi, J., Chang, X., Maiti, S., Livescu, K., & Watanabe, S. (2024). *Towards robust speech representation learning for thousands of languages* (arXiv:2407.00837). arXiv. https://doi.org/10.48550/arXiv.2407.00837

Conneau, A., Baevski, A., Collobert, R., Mohamed, A., & Auli, M. (2020). *Unsupervised cross-lingual representation learning for speech recognition* (arXiv:2006.13979). arXiv. https://doi.org/10.48550/arXiv.2006.13979

Coto-Solano, R. (2024). *Natural language processing for Indigenous languages* [Workshop]. CoLang 2024. https://www.colang2024.org/workshops

Coto-Solano, R., Nicholas, S. A., Datta, S., Quint, V., Wills, P., Powell, E. N., Koka'ua, L., Tanveer, S., & Feldman, I. (2022). Development of automatic speech recognition for the documentation of Cook Islands Māori. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (pp. 3872–3882). European Language Resources Association. https://aclanthology.org/2022.lrec-1.412/

Cox, C. (2023). *XLS-R-ELAN: An implementation of XLS-R automatic speech recognition as a recognizer for ELAN* (Version 0.2.0) [Python]. https://github.com/coxchristopher/xls-r-elan

D K, T., James, J., Gopinath, D. P., & Ashraf K, M. (2025). Advocating character error rate for multilingual ASR evaluation. In L. Chiruzzo, A. Ritter, & L. Wang (Eds.), *Findings of the Association for Computational Linguistics: NAACL 2025* (pp. 4926–4935). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.findings-naacl.277

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional Transformers for language understanding* (arXiv:1810.04805). arXiv. https://doi.org/10.48550/arXiv.1810.04805

FFmpeg Developers. (2025). *FFmpeg*. FFmpeg. https://ffmpeg.org/

Gale, R. C., Salem, A. C., Fergadiotis, G., & Bedrick, S. (2023). Mixed orthographic/phonemic language modeling: Beyond orthographically restricted Transformers (BORT). In B. Can, M. Mozes, S. Cahyawijaya, N. Saphra, N. Kassner, S. Ravfogel, A. Ravichander, C. Zhao, I. Augenstein, A. Rogers, K. Cho, E. Grefenstette, & L. Voita (Eds.), *Proceedings of the 8th Workshop on Representation Learning for NLP* (pp. 212–225). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.repl4nlp-1.18

Gerdts, D. B. (2014). *Object and absolutive in Halkomelem Salish*. Routledge. https://doi.org/10.4324/9781315852232

Gerdts, D. B., Edwards, L., Ulrich, C., & Compton, B. (1997). *Hul'q'umin'um' words: An English-to-Hul'q'umin'um' and Hul'q'umin'um'-to-English dictionary* (T. E. Hukari & R. Peter, Eds.). Cowichan Tribes. https://sqwal.hwulmuhwqun.ca/resources/

Gerdts, D. B., & Werle, A. (2014). Halkomelem clitic types. *Morphology (Dordrecht)*, *24*(3), 245–281.

Gessner, S., Herbert, T., & Parker, A. (2023). The report on the status of B.C. First Nations languages 2022. *First Peoples' Cultural Council*. https://fpcc.ca/resource/language-status-report-2022/

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527

HLCS. (n.d.). Welcome to the Hul'q'umi'num' Language & Culture Society. *HLCS*. Retrieved January 9, 2025, from https://hlcs.hwulmuhwqun.ca/

HLCS. (2021). *Hul'q'umi'num' 175 little words*. https://sqwal.hwulmuhwqun.ca/wp-content/uploads/2021/06/HLA_175LittleWords.pdf

Kava, T. (1969). *A phonology of Cowichan* [Master's Thesis]. University of Victoria.

Kleynhans, N. T., & Barnard, E. (2015). Efficient data selection for ASR. *Language Resources and Evaluation*, *49*(2), 327–353.

Lample, G., & Conneau, A. (2019). *Cross-lingual language model pretraining* (arXiv:1901.07291). arXiv. https://doi.org/10.48550/arXiv.1901.07291

Lane, W. (2023). *Sparse transcription service* (Version 1) [Python]. https://github.com/abbottLane/sparse-transcription-service (Original work published 2023)

Lane, W., Bettinson, M., & Bird, S. (2021). A computational model for interactive transcription. In E. Dragut, Y. Li, L. Popa, & S. Vucetic (Eds.), *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances* (pp. 105–111). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.dash-1.16

Lannelongue, L., Grealey, J., & Inouye, M. (2021). Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, *8*(12), 2100707. https://doi.org/10.1002/advs.202100707

LeCun, Y. (2019). *Yann LeCun*. Facebook. https://www.facebook.com/yann.lecun/posts/i-now-call-it-self-supervised-learning-because-unsupervised-is-both-a-loaded-and/10155934004262143/

Leonard, J. (2019). *The phonological representation and distribution of vowel in SENĆOŦEN (Saanich)* [Doctoral dissertation, University of Victoria]. http://hdl.handle.net/1828/10563

Leonard, W. Y. (2018). Reflections on (de)colonialism in language documentation. *Language Documentation & Conservation Special Publication*, *15*, 55–65.

Li, D., Zhao, H., Zeng, Q., & Du, M. (2025). Exploring multilingual probing in Large Language Models: A cross-language analysis. In H. Fei, K. Tu, Y. Zhang, X. Hu, W. Han, Z. Jia, Z. Zheng, Y. Cao, M. Zhang, W. Lu, N. Siddharth, L. Øvrelid, N. Xue, & Y. Zhang (Eds.), *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)* (pp. 61–70). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.xllm-1.7

Li, S. (2024). *Cross-lingual and cross-modal limitations of large language models* [Master's Thesis, University of Alberta]. https://ualberta.scholaris.ca/items/140fa033-d890-4f14-8bbf-c946a45c8d3c

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A robustly optimized BERT pretraining approach* (arXiv:1907.11692). arXiv. https://doi.org/10.48550/arXiv.1907.11692

Loweimi, E., Carmantini, A., Bell, P., Renals, S., & Cvetkovic, Z. (2023). Phonetic error analysis beyond phone error rate. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *31*, 3346–3361. https://doi.org/10.1109/TASLP.2023.3313417

Lowman, E. B., & Barker, A. J. (2015). *Settler: Identity and colonialism in 21st century Canada*. Fernwood Publishing.

Maiti, S., Peng, Y., Choi, S., Jung, J., Chang, X., & Watanabe, S. (2023, September 14). *Voxtlm: Unified decoder-only models for consolidating speech recognition/synthesis and speech/text continuation tasks*. arXiv.Org. https://arxiv.org/abs/2309.07937v3

Marshall, M. (2018). *The rhythm of Hul'q'umi'num': An exploration of Salish phonetics*. http://hdl.handle.net/1828/9243

Masuyama, Y., Miyazaki, K., & Murata, M. (2024, November 11). *Mamba-based decoder-only approach with bidirectional speech modeling for speech recognition*. arXiv.Org. https://arxiv.org/abs/2411.06968v1

Nijmegen: Max Planck Institute for Psycholinguistics. (2025). *ELAN* (Version 7.0) [Computer software]. The Language Archive. https://archive.mpi.nl/tla/elan

Noahedits. (2019). *Map of Coast Salish linguistic distribution in the early to mid 1800s* [Graphic]. https://en.wikipedia.org/wiki/Coast_Salish_languages#/media/File:Coast_Salish_language_map.svg

Nolan, T. (2017). *A phonetic Investigation of vowel variation in Lekwungen*. University of Victoria.

Otmakhova, Y., Verspoor, K., & Lau, J. H. (2022). Cross-linguistic comparison of linguistic feature encoding in BERT models for typologically different languages. In E. Vylomova, E. Ponti, & R. Cotterell (Eds.), *Proceedings of the 4th Workshop on Research in Computational Linguistic Typology and Multilingual NLP* (pp. 27–35). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.sigtyp-1.4

Percival, M., Yeung, H. H., Bird, S., & Jack, Q. (Randeana). (2025). Hul'q'umi'num' listening quizzes: Speech perception in a language revitalization classroom context. *Journal of Second Language Pronunciation*, *11*(3), 363–393. https://doi.org/10.1075/jslp.25001.per

PyTorch Foundation. (2025). *Torchaudio.load* (Version 2.8) [Computer software]. https://www.pinecone.io/learn/batch-layer-normalization/

Rodríguez, L. M., & Cox, C. (2023). Speech-to-text recognition for multilingual spoken data in language documentation. *Proceedings of the Sixth Workshop on the Use of Computational Methods in the Study of Endangered Languages*. ComputEL-6.

Sivakumar, M., Parthasarathy, S., & Padmapriya, T. (2024). Trade-off between training and testing ratio in machine learning for medical image processing. *PeerJ Computer Science*, *10*, e2245. https://doi.org/10.7717/peerj-cs.2245

Sun, M. (2020). *Trainer* (Version 2.0) [Computer software]. HuggingFace Inc. https://huggingface.co/docs/transformers/main/main_classes/trainer

The Hul'q'umi'num' Language & Culture Society. (2022). *Sounds of Hul'q'umi'num': The Hul'q'umi'num' phonetic inventory*. https://sqwal.hwulmuhwqun.ca/wp-content/uploads/2022/05/Hulquminum-phonetic-inventory.pdf

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention is all you need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

von Platen, P. (2021). *Fine-tune XLSR-Wav2Vec2 for low-resource ASR with 🤗 Transformers* [Blog]. Hugging Face. https://huggingface.co/blog/fine-tune-xlsr-wav2vec2

Webb, R. (2022). *Hul'q'umi'num' storytellers' use of gestures to express space and viewpoint* [M. A., Simon Fraser University]. https://summit.sfu.ca/item/35315

Webb, R. (2025). *Prosody of "i" and "uw" in Hul'q'umi'num'* [Unpublished]. University of Victoria.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., … Rush, A. (2020). Transformers: State-of-the-art natural

language processing. In Q. Liu & D. Schlangen (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 38–45). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Wong, R. (2008). Decolonizasian: Reading Asian and First Nations relations in literature. *Canadian Literature*, (199), 158–181.

Appendix A

How to use the trained XLS-R model

Use this link to view a video demonstrating how to use the XLS-R model that was fine-tuned for this project to transcribe you own audio files: https://youtu.be/_xsKXT0T9Uo

or scan this QR code:

Appendix B

List of ordered transcriptions

All of the transcriptions used in the testing phase of fine-tuning—and thus, in the error analysis—are shown in this appendix in the order that they appear in each story. The transcriptions are provided in the following format:

(Line number)
Human transcription (HT)
Machine transcription (MT)
English translation

There are many lines and portions of sentences missing because the audio was segmented based on breath groups and the audio segments used for testing were randomly selected. The HTs are shown with special characters removed—with the exception of apostrophes and double quotation marks—to show how they were edited for fine-tuning. Asterisks are used to indicate that the English translation for multiple breath groups is given, because the translations were not always divided by breath group in the EAF files. For these cases, the missing portions of the HT are provided in parentheses. Multiple transcribed breath groups from the same line number are shown separated by a vertical line.

yu ‘um’mush tthu t’ut’um’ (Little Wren Goes Hunting)

(12)
wulh lumnuhwus tthu ni’
wulh lumnuhwus tthu ni’
when he saw something

(19)
“q’aythamu tsun p’e’ ” thut t’ut’um’
’aaythamu tsun n p’e’ sutts’ut’um’
“I’m going to kill you,” said Wren.

(23)
“nem’ ’uhw wa’
nem’ uhwa’

"No way!

(34)
“hey’lh kwiya’
the’lhkwuiya’
“Okay, go ahead and try then!”

(35)
yu kwun’etus thu shuptun
yu kwun’etus tthu shuptun
carrying his knife.

(36)
ni i i’ huye’ lhakw’ nuw’ilum ’u tthu (q’uq’i’s tthu q’uyi’uts.)
ni i huye’ lhakw’ nuw’ilum ’u tthu
He flew into the moose’s innards.*

(39)
tl’e’ wulh hwu’alum’
ni’ tl’e wulh hwu’alum’
And then he went back in again.

(40)
m’i ’utl’qul ’i’ ni’ hwi’
m’i ’utl’quli’nuhw ’i
He came out and this time he came out of the moose’s bum.

(48)
ni’ (lhiluts’utum tthu q’uq’i’s tthu q’uyi’uts.)
ni’
He killed the moose by slicing its guts.*

(56)
’ii ch ’uw’ hiil’e’th ’ul’ ”
’i’ ch’uw’ hiyul’ethul’
You are just fooling me.”

(58)
tl’e’ wulh qul’et t’ilum:
tl’e’ wulh qul’et t’ilum
And he sang again:

(67)
nem’ tsun lumstamu ”
nem’ tsun lumstamu
Come and I will show you.”

(70)
nilh ni' hwu s'ulhtuns
nilh ni' hwu s'ulhtuns
That would be their food,

(72)
hay 'ul 'uw' ('uy' shqwaluwuns kwsus)
hay 'ul 'uw'
She was so happy*

(75)
ni' (hwu s'ulhtuns.)
hni'
would get some food.*

(84)
sutst 'uw' huye' 'i' kwthunu shuyulh ne e m' (t'ahw)
su tst 'uw' huye' tthunu shuyulhne e em'
My brother and I would go down*

(86)
sutst 'uw' tus 'u kwthu
ni' ts't uw' tus 'u kwthu
And we would get to the

(89)
hay 'ul' 'uw' thi i i
hay 'ul' 'uw' thi i
They would be so big.

(96)
sutst nem 'uw' sew'q't tthu s'ulhtun
utnem' 'uw' sew'q'u tthu s'ulhtun
We would go and look for food.

(97)
kw'a'luhw
kw'a'luhw
dog salmon.

(101)
('i' ni' hiqushus hiqushum, t'uyum'tum 'unyuns,) tl'elhum | pupu susuw' hiqutum
tl'elhlum | pupu susuw' hiqutum
And she would bake it with onions, salt, and pepper.*

(105)
(nilh kwu'elh st'e'ukw' 'uw' shhw'iiw'tsusta'ult kws) thuthi'st hwut 'u tthu shqwaluwun tst (kws 'i'mush tst 'i' u tun'a tumuhw 'uy'st-hwut lhey' sxwi'em'.)
suthi'st hwu tthu shqwalwuns 'u tthu
And this is how we were taught how to be as we walked this earth to like these stories.*

(106)
ni' hay hay ch q'u
ni' hay hay ch q'u
The end. Thank you.

s-hwuhwa'us 'i' lhu q'ullhanumutsun (Thunderbird and Orca)

(1)
(tl'uw' qux mustimuhw tthu ni' 'utl' qw'umi'iqun' shni's lhu) smeent (ti'wi'ulh'ew't-hw.)
sment
There were many many people over at Comiaken where the Stone Church is.*

(5)
tth'a'kwus sil'anum te'tsus sil'anum
thw'a'kwus xi'anum te'tsus sil'anum
they were about 7, 8 years old.

(17)
ni' xunuq't tthuw'nilh stl'itl'qulh 'i' ni' wulh (wil' tthu huy'qw.)
ni' xunuq't tthuw'nilh st'i'tl'qulh 'i' ni' wulh
He opened his eyes and fire would come out.*

(25)
'uwu te' ni' skw'ey kws tl'e's lemutus kw' stem (skw'ey kws lemut-s kw' lhwet.)
'uwu teun ni' shkw'ey kws tl'e lemutus ks nem
He couldn't ever look at anything or anybody.*

(27)
'i' nilh kwus st'e 'u tthey' ni' hwu (ni' hwu) | huy'qw tthuw' mukw' stem 'u kwsus lemutus
'i' nilh kwsush ste' 'u tthey' hwu | huy'qw tthuw' mukw'stemu kwsus lemutus
So it was like that—whatever he looked at would burn.*

(31)
ha' ni' xunuq't 'u kwsus sent ('i' nilh sus 'uw')
t'a ' ni' xunuq't 'u kwsus sent
When he opened his eyes at night,

(35)
sus 'uw' thuyuw't hwtum ni' 'utl' satellite island—ni' tsun mel'qt kwthu s hwulmuhwa'lh snes (thuytum tthu) | shni's thuyuw't hwtum

susuw’ thuyuw’t hwtum ni’ ’u tl’ setley’t’ayluni’ tsun me’qthu sqwulmuhwa’lhsnis | shni’s thuyuw’t hwtum
So they built him a home over at Satellite Island, fixed him a place, made him a home.*

(35)
(thuytum tthu) shni’s thuyuw’t hwtum
shni’s thuyuw’t hwtum
fixed him a place, made him a home.*

(38)
ha’ kwu’elh ni’ net ’u kwus snet
’a’ kw’elh ni’ net ’u kwsusnet
And when it was nighttime,

(49)
sht’es kwus (kwen’nuhwus tthu hwulmuhw tthu s’ulhtuns.)
sht’es kwus
and that’s how the First Nations people got their food.*

(52)
nuw’ sxuxits kwthu ni' (kwe’tum, nem’ ’aantum kws nem’s yul’ew’ ’u thu shxetl’.)
nuw’ sx’uxits kwthu ni’
They figured out what they which salmon they would let pass through the weir.*

(53)
nuw’ sxuxits ’ul’ kwthu m’i kwunutum ni’ tse’ (sq’i’lu.)
nuw’ shxuxitsul’ kwthu m ’yu kwunutum ni tse’
They knew what to take for their food for the winter.*

(55)
sht’es kwus ’a’untum’ tthu stseelhtun
sht’es kwus ’a’nutum’ tthu stselhtun
when the salmon came up the river.

(61)
“tstamut tst tse’ kwu’elh ’uwu te’ s’ulhtun tst
tstamut tse’ kwu’lh ’uwu te’ s’ulhtun tst
What are we going to do? We have no food!

(75)
sus ’uw’ ’uya’qthut (sqw’ulesh, ni’ hwu s-hwuhwa’us.)
susuw’ ’uiya’qthut
He changed himself into Thunderbird.*

(76)

si i is m'uw' lhakw' m'i (ewu 'utl' tl'ulpalus.)
is m'uw' lhakw' m'i
And he flew to Cowichan Bay.*

(80)
(nilh kwu'elh ni' hulinhw tthu mustimuhw) kwthey' swiw'lus | 'i' ni' hwu lhalhukw'
kwthey'swiw'lus | 'i' ni' hwu lhalhukw'
So that young man saved all the people and he became a flying creature.*

(84)
'i' wulh nele' tthey' swaaw'lus suw' (kwulushtum 'u tthu)
'i' wulh nele' tthey' stwaw'lus susuw'
And the young men started shooting at him with the*

(95)
ni' tun'ni' 'u thu lagoon sus 'uw' hwu quliima'
ni' tuni' 'u tthu lukwon susuw' hwuqulimu'
It's all dirty from the lagoon.

(98)
ni' hay nu sxwi'em' ni' 'uw'kw'
yue' haynusxwi'em' n' 'uw'kw
That's the end of the story. There's no more.

q'ise'q 'i' tthu munmaanta'qw (Q'ise'q and the Stoneheads)

(5)
(tus tthu) sqe'uqs
sqe'uqs
His younger brother got there,*

(8)
('i' nilh tse') nus 'uw' tstamut 'uw' niin' tse' q'ay 'i' nuwu tse' hwu
susuw' tstamut 'u' ni' n' tse' q'ay'i'nauwu tse' hwu
When the time comes for me to die,*

(9)
'i' nilh thulh suw' kwans kw' (swuy'qe'allh)
'i' nilh thuth suw' kwans kw
But when a boy child is born*

(10)
kwun'et ch kwu'elh tthey' ('i' nilhs m'is 'uw' tetsul kwthu kw' nu 'imuth swuy'qe' nu 'imuth 'i' nilh tse' suw'nilhs hwu kwun'et, nilh 'uye'qth.")
kwun'ets chkwe'ulh tthey'
So you will rule until a grandson is born and can take over."*

(23)
sus muw' hwkw'atus nem'
sis nem' 'uw'hkw'atus nem'
pulled it down

(25)
"ha' (tuw' swuy'qe' wa'!)
ha a'
"Hey, that's maybe a boy!*

(30)
"a a a shme'tth'un'qun ch
'a shmeth'unqunch
"You are lying,

(35)
stl'is kwsuw' tul'nuhws 'uw' thu'itus 'uw' slhelhni' tthu (ni' kwan qeq, qeqs thu swunumelhs.)
tl'i's kwsuw' tul'nuhwsuw' thu'itus 'uw' slhelhni' thu
He wanted to confirm that his niece's newborn baby was a girl.*

(38)
"slhelhni' thu qeq "
slhelhni' thu qeq
"The baby is a girl."

(47)
(nilh kwsusulh) ni' lhey' slheni 'i' xeem' tthu qeqs
nilh kwsus 'ulh ni' lhey' slheni' 'i' xem' thu qeqs
This is because that young woman was there and her baby cried.*

(51)
tl'e' wulh tl'qw'uthut (ni' wulh) | nem' 'uw' tuw' wulh hith kwus ('u tthey' ni' shni's.)
tl'e' wulh qw'uthut | nem' 'uw' tuw'ulh hith kwus
She gathered up her things again and went to another place for a while.*

(52)
"'uy' kwunus nem' tha'ithut
'uy' kwunus nem' tha'ithut
"I'd better keep moving.

(55)
tetsul 'utl' kwa'mutsun susuw'
tetsul 'utl' kwa'mutsun susuw'
When she reached Quamichan,

(56)

suw' pte'mut s tthu 'imuths
te'mut s tthu 'imus
There she made herself a place to live,

(60)
nuw' ni' 'ul' kwsus ('u kwthu nu shni'.)
nuw' ni' 'ul' kwsus
That rock is still there at my place.*

(67)
tens thuw'nilh q'e'mi'
thu tens thuw'nilh q'e'mi'
the mother of this girl,

(75)
ni' wulh hwu 'i'mush xwi'xwan'chunum'
ni' wulh hwu 'i'mushxwi'xwan'chunem'
He was already walking, running around,

(87)
(mukw' shakw'ut-s tthu mun'us 'i' 'uw' nilh nuw' sht'es kwsus wi'wul' tthu kw'suts,) qux kw'suts
qux kw'suts
Each time she bathed her son, a lot of trout would appear.*

(88)
('i' kwsus wulh yu ts'its'usum' tthuw'nilh q'ise'q, kwsus wulh nem' 'imush nem'sew'q' 'u tthu) sqw'ulesh
sqw'ulesh
And when Q'ise'q was growing up, he used to go out hunting birds.*

(89)
mu u ukw' stem sqw'ulesh 'i' ni' kwulushtus susuw' hwtth'a'tus (hwtth'a'tus susuw' thuytus tthu ni' hakwushus ni' tse' shlhalhukw's.)
mu kw' stemsqw'ulesh 'i' ni' kwulushtus susuw' hwtts'a'tus
He shot all kinds of birds, and he would skin them, he used the skins to make himself some wings to fly with.*

(96 & 97)
nutsim' 'a'lu ni' 'a'lu 'untsu kw' (qul'et mustimuhw?")
nutsim 'a'lu nu lu'unts 'ukw'
Why? Where are the other relatives?"*

(104)
('uwu te' kwu'elh shtatul'stuhws kwthunu shmuthi'elh kwun's 'uw') 'uw' huli "
uw'huli

My uncle doesn't know that you are alive."*

(106)
('i' ni' ts'u 'uw' 'ulh sa-ay' tthey' ni') 'a'luxutus sqw'ulqw'ulesh
l'a'luxutus sqw'ul'qw'ulesh
And he had already gathered many birds.*

(115)
'iilh huy'aatum' 'u thu tens "'uwu ch nem'uhw hunum' 'utl' xinupsum
'ilh hui'aatum 'u thu tens 'u wulh shnem' uhwunum 'utl' xinupsum'
His mother had warned him, "Don't ever go to Xinupsum.

(121)
(ni' ts'u tuw') st'e 'u tun'a haki | ni' ni' tthu ni' tuw' 'e'uhwiin' 'i' nilh kwus st'i'am' tthu (stl'q'een' susuw' qw'aqw'uqwutum'.)
st'e 'u tun'a haks 'i | ni' ni' thu nit 'uw' e'hwi ni' nilh kwsus tl'i'am' tthu
It was like a hockey puck, a small knot with a feather on it, and they would hit it.*

(125)
qux swaaw'lus hay 'ul' qux
qux swawlu say 'ul' 'ux
There were lots and lots of young men.

(128)
(xe'xtsitus, xe'xtsitus tthu ni') niilh shtun'naalhtuns
nilh shttun'naalhtuns
So he was watching them, studying the people at the place where he came from.*

(129)
sus nem' 'uw' t'akw' ni' kweyul 'i' (tl'e' wulh m'i qul'et lemutus.)
sis nem' 'uw' t'akw'ni' kwuyul 'i'
He'd go home and then the next day he'd go watch them again.*

(130)
(sht'es tthu ni' sul'ul'uthut-s tthuw'ne'ullh,) tthey' shtun'naalhtun xinupsum
they' shtun'nalhtuns xinupsum
He went to see what the people he was from were like, the ones from Xinupsum.*

(140)
hay 'ul' qux kwus naalts'tul tthu ni' yu (yu they'tus huy'tuns.)
hay 'ul' qux kwus snaltstul tthu ni' yu
He tried many different kinds of trees to make his weapon.*

(143)
ni' tsun mel'qt kwu (snes, sht'eewun' tsun xpiinlhp.)
ni' tsu mea'qt kwu

I forgot its name, maybe xpiinlhp.*