# Automatic Speech Recognition for Documenting Endangered First Nations Languages

A Thesis Submitted to the

College of Graduate and Postdoctoral Studies

In Partial Fulfillment of the Requirements

for the Degree of Master of Science in the

Department of Electrical and Computer Engineering

University of Saskatchewan

Saskatoon, Saskatchewan, Canada

By

**Zarif Al Sadeque**

# Permission To Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan, S7N 5C9
Canada

Or

Head of the Department of Electrical and Computer Engineering
University of Saskatchewan
57 Campus Drive
Saskatoon, Saskatchewan
Canada, S7N 5A9

# Abstract

Automatic speech recognition (ASR) for low-resource languages is an active field of research. Over the past years with the advent of deep learning, impressive achievements have been reported using minimal resources. As many of the world's languages are getting extinct every year, with every dying language we lose intellect, culture, values, and tradition which generally pass down for long generations. Linguists throughout the world have already initiated many projects on language documentation to preserve such endangered languages. Automatic speech recognition is a solution to accelerate the documentation process reducing the annotation time for field linguists as well as the overall cost of the project. A traditional speech recognizer is trained on thousands of hours of acoustic data and a phonetic dictionary that includes all words from the language. End-to-End ASR systems have shown dramatic improvement for major languages. Especially, recent advancement in self-supervised representation learning which takes advantage of large corpora of untranscribed speech data has become the state-of-the-art for speech recognition technology. However, for resource-constrained languages, the technology is not tested in depth. In this thesis, we explore both traditional methods of ASR and state-of-the-art end-to-end systems for modeling a critically endangered Athabascan language known as Upper Tanana. In our first approach, we investigate traditional models with a comparative study on feature selection and a performance comparison with deep hybrid models. With limited resources at our disposal, we build a working ASR system based on a grapheme-to-phoneme (G2P) phonetic dictionary. The acoustic model can also be used as a separate forced alignment tool for the automatic alignment of training data. The results show that the GMM-HMM methods outperform deep hybrid models in low-resource acoustic modeling. In our second approach, we propose using Domain-adapted Cross-lingual Speech Recognition (DA-XLSR) for an ASR system, developed over the wav2vec 2.0 framework that utilizes pretrained transformer models leveraging cross lingual data for building an acoustic representation. The proposed system uses a multistage transfer learning process in order to fine tune the final model. To supplement the limited data, we compile a data augmentation strategy combining six augmentation techniques. The speech model uses Connectionist Temporal Classification (CTC) for an alignment free training and does not require any pronunciation dictionary or language model. Experiments from the second approach demonstrate that it can outperform the best traditional or end-to-end models in terms of word error rate (WER) and

produce a powerful utterance level transcription. On top of that, the augmentation strategy is tested on several end-to-end models, and it provides a consistent improvement in performance. While the best proposed model can currently reduce the WER significantly, it may still require further research to completely replace the need for human transcribers.

# Acknowledgments

All praises to almighty Allah who created us, and I am thankful to Allah for granting me the knowledge, patience, and strength for pursuing my M.Sc. degree successfully.

I am very grateful to my supervisor, Dr. Francis Bui who constantly supported and guided me throughout my whole M.Sc. program at the University of Saskatchewan. I also thank my advisory committee members, Dr. Ebrahim Bedeer Mohamed and Dr. Chris W. Zhang for their insightful comments, reviews, and suggestions for improving this thesis.

Thanks to Dr. Olga Lovick for providing me with the dataset of this research as well as her valuable suggestion. I would like to thank my lab colleagues for their ideas and the excellent work environment in the lab.

Special thanks to my family members, my elder brother, my sisters, and my twin brother for providing me with continuous mental support, love, and inspiration from a distance.

# Dedication

This thesis work is dedicated to the memory of my father Mohammad Helal Uddin, who passed away from COVID-19 during my second year of the M.Sc. program, who inspired me a lot to pursue this degree, and nothing will be enough to show him my gratitude.

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| AM | Acoustic Model |
| ANN | Artificial Neural Network |
| BERT | Bidirectional Encoder Representations from Transformers |
| BLSTM | Bidirectional Long Short Term Memory |
| CER | Character Error Rate |
| CMVN | Cepstral Mean and Variance Normalization. |
| CNN | Convolutional Neural Network |
| CTC | Connectionist Temporal Classification |
| CV | Computer Vision |
| DBN | Deep Belief Network |
| DCT | Discrete Cosine Transform. |
| DFT | Discrete Fourier Transform. |
| DL | Deep Learning |
| DNN | Deep Neural Network |
| DPT | Discriminative Pretraining |
| E2E | End to End |
| ED | Encoder Decoder |
| ELPIS | Endangered Language Pipeline and Inference |
| EM | Expectation Maximization |
| FMLLR | Feature Space Maximum Likelihood Linear Regression |
| G2P | Grapheme-To-Phoneme |
| GMM | Gaussian Mixture Model |
| GPU | Graphics Processing Unit |
| GELU | Gaussian Error Linear Units |
| GRU | Gated Recurrent Unit |
| HMM | Hidden Markov Model |
| IU | Intonation Units |
| LDA | Linear Discriminant Analysis |
| LM | Language Model |

| | |
|---|---|
| LPC | Linear Predictive Coding |
| MAP | Maximum A Posteriori |
| MFCC | Mel-Frequency Cepstral Coefficient |
| ML | Machine Learning |
| MLE | Maximum Likelihood Estimation |
| MLLR | Maximum Likelihood Linear Regression |
| MLLT | Maximum Likelihood Linear Transform. |
| MLP | Multilayer Perceptron |
| MT | Machine Translation |
| NCCF | Normalized Cross Correlation Function |
| NLP | Natural Language Processing |
| NSERC | Natural Sciences and Engineering Research Council of Canada |
| NSF | National Science Foundation |
| NMT | Neural Machine Translation |
| NN | Neural Network |
| OOV | Out of Vocabulary |
| PCA | Principal Component Analysis |
| PDF | Probability Density Function |
| PoV | Probability of Voicing |
| PLDA | Probabilistic Linear Discriminant Analysis |
| PLP | Perceptual Linear Prediction |
| RELU | Rectified Linear Unit |
| RNN | Recurrent Neural Network |
| RBM | Restricted Boltzmann Machine |
| SAT | Speaker-Adaptive Training |
| Seq2Seq | Sequence to Sequence |
| SNR | Signal to Noise Ratio |
| SSL | Self Supervised Learning |
| SSRL | Self Supervised Representation Learning |
| STFT | Short Time Fourier Transform |
| TDNN | Time Delay Neural Network |

| | |
|---|---|
| TL | Transfer Learning |
| TVS | Total Variability Space |
| UBM | Universal Background Model |
| VQ-VAE | Vector Quantized Variational Autoencoder |
| WER | Word Error Rate |

# 1 Introduction

Speech and text are the most common modes of communication. Automatic speech recognition (ASR) is the technology that automatically converts speech to its text form. Nowadays, there are many speech recognition tools integrated into our everyday life such as Google assistant, Siri, Google Translate etc. With the increasing use of smart devices and huge data resources, many companies and government agencies are interested in the development of speech technology. Another important application of ASR is in language documentation. More than half of the world's languages are currently at different levels of endangered state, which may not persist another century [1]. First nations or tribal languages are amongst the highest severity of the endangered languages as most of them have few speakers and the new generations are diverting to major spoken languages due to economic, social or political reasons. Languages are carriers of cultural heritage, memory of important individuals or events, and they hold diverse information on linguistic evolution. Therefore, in order to preserve these languages, linguists have already started many documentation projects.

## 1.1 Automatic Speech Recognition for Language Documentation

ASR can play a key role in the language documentation process. Firstly, it can reduce the workload of field linguists. Manual annotation of recordings can take thousands of hours depending on the size of the corpus [2]. A survey from 2017 shows that, on average, each minute of audio data takes around 40 minutes depending on the difficulty of the associated file for a manual transcriber [3]. By contrast, a time-sensitive experiment reports that ASR assisted transcription takes 15% less time than an expert transcriber and up to 42% for a slow transcriber [4]. Most often, a fieldworker might have to work with any available transcriber rather than an expert transcriber. Secondly, ASR can provide better transcription in certain situations as well as assist in linguistic analysis for language documentation. Transcribers sometimes tend to overhear or ignore hesitations, repetitions or corrections by the speaker, which can be important for future analysis [5]. Humans also have physical limitations (dizziness, mental situation) that can influence the quality of the transcription. ASR assisted transcription provide better result in such situations.

Besides, language documentation often includes word-to-word alignment and phonetic analysis, which can be facilitated efficiently using an ASR system.

## 1.2 Challenges

There are approximately 7000 languages in the world. However, only about 100 languages are currently suitable for being used with a speech recognition system [6]. This is due to the fact that a reliable ASR system requires a tremendous amount of annotated speech data and linguistic expertise. A standard ASR pipeline has three prerequisites: 1) thousands of hours of audio-transcription pair for training an acoustic model; 2) a phonetic or pronunciation dictionary that describes how each word of the language is structured with the acoustic units/phoneme; and 3) a large collection of text data for estimating how words of the language are structured, for building the language model. As discussed before, endangered language documentation projects generally take a long time and also are quite expensive. Even if the recordings are collected, it takes even longer time to annotate them due to the shortage of expert transcribers. Some endangered languages also lack standard orthography or a proper writing system, and as a result, there might be little or no text data available for that language. Thus, given the long pipeline of data collection and preparation till the point of system deployment, the whole process can take years, by which point it could be too late for the target application for many endangered languages. In short, developing an ASR system focusing on language documentation should have the following considerations,

1. The system should be able to train from very little or no data from the target language.
2. It should incorporate well with the standard workflow of a typical language documentation process.

Another challenge for endangered languages is the morphological complexity. It is a common practice for low resource ASR systems to use other languages to supplement the data scarcity. This type of training requires native speakers or expert linguists with technical knowledge while building the ASR system. Because the added language may consist of fuzzy grammatical structure or different phoneme compositions which are better understood by the native speaker of those languages or expert linguist. However, it’s very difficult to find a native speaker or linguist

with the required expertise at the time of system development specially for a heavily endangered target language.

This thesis investigates the ASR technology on a critically endangered Athabascan language known as Upper Tanana. At present, Upper Tanana has 42 known speakers in the whole world with five dialects [7]. But this thesis focuses only on the "Tetlin" dialect of the language which has only 20 known speakers. This language is mainly spoken in some communities of Eastern Alaska and some parts of Yukon territory of Canada. Currently, there are only a few literatures available about this language, but no prior research done on automatic speech recognition. This language is considered a morphologically complex language and it has no pronunciation dictionary readily available. It is also considered a tonal language, but the Tetlin dialect has mostly lost its tonal variation.

Therefore, we must come up with a system or strategy that can tackle the above challenges associated with languages like Upper Tanana and requires the least human intervention.

## 1.3 Literature Review of Low Resource ASR

Hidden Markov Model (HMM) is the most typical algorithm for acoustic modeling in speech recognition. The reason behind its success came from its strong immanent mathematical-statistical framework [8], [9] and convenient training & decoding algorithm with flexible structure [10]. HMMs are usually paired with GMMs where HMMs model the signal sequence and GMMs represent the local spectral variability [11]. Artificial Neural Network (ANN) which have shown to be competent in modeling highly non-linear patterns has brought new ideas to speech recognition. Hinton et al, in [12] used Restricted Boltzmann Machine for initializing the Deep Belief Networks (DBNs) that utilizes the greedy layer wise pretraining started to dominate the mainstream ASR system. DBNs shown to be effective for many low resource scenarios as well provided that strong context dependent trees are used to train the model.

Deep learning based techniques recently outperformed conventional ML approaches and yields to be very effective for ASR. One common way is to map the posterior probabilities of HMM states through the DNN output layer usually referred as DNN-HMM model. However, traditional DNNs can offer limited temporal modeling of acoustic frames. They are unable to

illustrate the long-term dependencies within the context [13]. RNNs specially the LSTMs have overcome such issues and are able to store particular historical information for a long time [14], [15]. Recently, Google made significant progress utilizing LSTMs for large vocabulary speech corpus [16]. In terms of context dependency, unidirectional LSTMs only depend on past information. Whereas, bidirectional LSTMs (BLSTMs) are able to take full advantage of past and future contexts by jointly modeling two unidirectional LSTMs superimposed on each other [17]. Many research show BLSTMS are also quite useful in low resource scenarios [18]–[20]. Nonetheless, BLSTMs are certainly powerful but the models have high latency therefore incompatible with real-time decoding. Studies suggest chunk based training and modifying the decoding algorithm seem to improve such setbacks [16], [21], [22]. Even though variants of LSTM perform really well in different speech recognition tasks, they need a longer training time compared to typical feed-forward networks.

Convolutional Neural networks (CNNs) are well known for their compelling feature extraction capability. CNNs can be applied for one-dimensional acoustic modeling without any pooling layers. These are known as TDNNs in the ASR domain. TDNNs use a feed-forward architecture that can be trained significantly faster than plain RNNs. In traditional TDNN architecture, the first layer of the network learns the context from a narrow slice of either the raw or processed speech signal and every subsequent layer learns by slicing the output from its previous layer. This method is very useful for understanding a wider temporal dependency. TDNNs in a likely manner have shown success for short term feature representations [23]. Combining the TDNN with LSTM as a hybrid model increases the capability of the architecture by capturing longer context information. Researchers evaluated various combinations of TDNN and LSTM layers and demonstrated a promising improvement in Word Error Rate (WER) [24], [25]. It is important to note that regardless of which Deep learning method is coupled with HMM, the performance always has some dependency on the initial alignment provided by the GMM-HMM models.

End-to-End (E2E) techniques have also shown remarkable achievement on WER in a large data training setup. The key advantage of an E2E model is, it can be trained without explicit knowledge of the language structure and the morphology; in other words, can be modeled without acquiring any phoneme dictionary. The integration of Connectionist Temporal Classification

(CTC) with ASR enabled the alignment free training of E2E models. Deep Speech developed by Baidu and its new modification Deep Speech 2.0 architecture are the first systems that demonstrated the effectiveness of CTC in speech recognition task [26]. The concept of Self Supervised Learning (SSL) or Self Supervised Representation Learning (SSRL) have been applied into different fields of computer vision (CV) and Natural Language Processing (NLP). There are multiple frameworks have been developed based on this technique such as BERT, Wav2Vec, GPT etc. In ASR it can build a generalized representation from unlabeled acoustic data and be able to learn the mapping from speech to text without any phoneme dictionary which caught the attention of many researchers for low resource model development [27], [28].

## 1.4 Research Questions

As of now, there is only a handful of research available for Upper Tanana. But there is no study so far regarding the ASR system in this language. Although there are lots of studies in acoustic modeling for low resource languages, the lack of both acoustic resources and orthography have been rarely studied in the past [29]. There are some studies that address a similar issue using domain adaptation from cross lingual SSRL [27], [30]. However, as we are dealing with a morphologically complex language with an extremely small dataset, it is important that not every recipe for implementation, pipeline, or weighted staging will suit the characteristics of a target language. This research also follows a non-linguistic perspective considering the developer has no prior linguistic expertise. The main research question of this thesis are as follows:

- What kind of traditional modeling is best fitted for low resource endangered languages?
- Can feature selection improve the output of a traditional ASR system?
- For E2E, a self supervised model pretrained in a cross lingual setting are supposed to learn a general representation of speech structure. Is this representation good enough to model a previously unseen language?
- Can an E2E model fine-tuned on an extremely small amount of target data outperform traditional ASR models?
- What is the effect of data size for fine tuning a self-supervised model?
- Does adding augmented acoustic data for fine tuning in addition to target data improve the result over transfer learning?

## 1.5 Thesis Organization

The rest of the thesis is organized as follows:

- Chapter 2 reviews concepts related to ASR including feature extraction, modeling techniques, Evaluation criteria and different frameworks used for this thesis.
- Chapter 3 is presented as a manuscript which describes our investigation on feature selection and different modeling techniques based on ASR for Upper Tanana
- Chapter 4 is presented as a manuscript which demonstrates our proposed DA-XLSR model, experimental results and discussion over other state-of-the-art models.
- Chapter 5 summarizes the key contributions of the thesis and provides directions for future work.

# 2 Background and System Modeling

This chapter describes the basics of automatic speech recognition, background concepts related to every step of building an ASR system including feature extraction, feature transformation, acoustic model, lexicon, language model, the notion of End-to-End ASR, different frameworks and evaluation matrices.

## 2.1 Automatic Speech Recognition

Automatic speech recognition is generally defined as a process of transcribing speech signals automatically into a sequence of linguistic units typically words, by means of machines or computer programs [31]. The process can be represented by a block diagram which consists of three main components a Lexicon, an acoustic model (AM) and a language model (LM) shown in Figure 2.1.

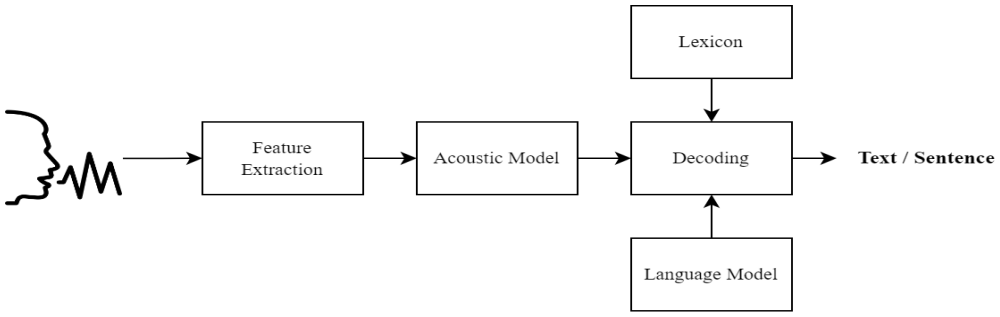


Figure 2.1: Generic Block Diagram of an ASR

The speech signal can be represented as a sequence of observation $X$ which captures the essential temporal information. In practice, the speech signal is processed to generate a feature vector using small windows of speech frames ranging from 20 to 30 ms shifting at a specific frame rate usually 10 ms over the whole signal. Given the speech sequence $X$, the problem can be defined as,

$$\widehat{W} = \arg\max_{w} P(W|X) \tag{2.1}$$

where the goal is to find the best word sequence $\widehat{W}$ that maximizes the posterior probability $P(W|X)$. The above equation can be rewritten applying Bayes theorem,

$$\begin{aligned} \widehat{W} &= \arg\max_{w} \frac{P(X|W)P(W)}{P(X)} \\ &= \arg\max_{w} P(X|W)P(W) \end{aligned} \tag{2.2}$$

Here, the conditional probability $P(X|W)$ is generally modelled using Hidden Markov Model (HMM) where the output probabilities are provided using pdfs from a Gaussian Mixture Model or Deep Neural Network. The hidden states $S$ of the HMM represents a sequence of subwords or phonemes which corresponds to the pronunciation of a given word. Therefore, the model can be represented under Markov's assumption as,

$$\widehat{W} = \arg\max_{w} P(X_n|S_n)P(S|W)P(W) \tag{2.3}$$

The likelihood term $P(X_n|S_n)$ is known as the acoustic model and the prior $P(W)$ is known as the language model. Since language models are usually built on word level, whereas speech signals are modelled at a phoneme level, a lexicon or commonly known as a pronunciation dictionary is required to map the phonemes to word.

## 2.2 Feature Extraction

As speech signal contains a lot of noise, extracted features provide much better results than the raw speech signal for speech recognition. This section presents some of the feature extraction methods used for this study.

### 2.2.1 Mel FilterBank

Mel FilterBank (Fbank) is a simple feature vector derived from Mel bands of a speech waveform. The distribution of the Mel bands is similar to the vocal system of a human; hence it is useful for both speech and speaker recognition. It is an earlier step of computing Mel Frequency Cepstral Coefficients (MFCC). Some of the literature suggests Fbank features tend to work better than MFCC for DNN based acoustic models [32]. The process of computing Fbank features is given in Figure 2.2.

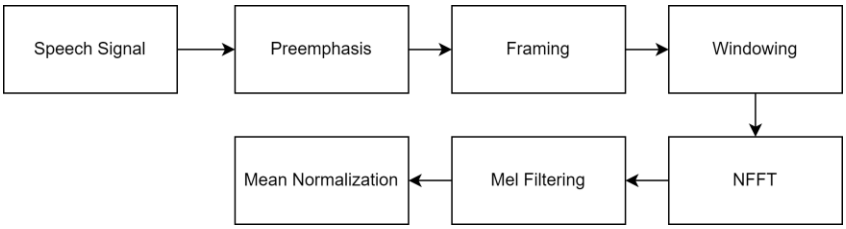


Figure 2.2: Computing Mel FilterBank features

The first step of computing Mel FilterBank is preemphasis. Due to the structure of the human vocal system higher frequencies produce less energy. Thus, this step balances the energy by boosting the higher frequencies through a high pass filter. If the speech waveform is $s(n)$ the output of the filter is given by,

$$\hat{s}(n) = s(n) - \alpha s(n-1) \tag{2.4}$$

where $\alpha$ is the preemphasis filter constant and usually set between 0.9 to 1. The resulting signal is then split into small frames of 20-30 ms as described earlier. As framing might cause discontinuities in the signal, thus a Hamming or Hanning window is applied to discard the discontinuities. A Hamming window can be formed using the following equation,

$$w(n) = 0.54 - 0.46\, cos\left(\frac{2\pi n}{N-1}\right) \tag{2.5}$$

where $N$ is the length of the window and $n = 0,..(N-1)$. After the window operation an N point FFT can be applied to individual frames to find the frequency spectrum. Followingly the power spectrum can be calculated using,

$$P = \frac{|FFT(s_n)|^2}{N} \tag{2.6}$$

where $s_n$ is an individual frame of signal $s$. Finally, the Mel spectrum can be computed by weighting the power spectrum with Mel filters. The weight is denoted by $H_m(k)$ and the Mel spectrum is given by $s(m)$.

$$s(m) = \sum_{k=0}^{N-1} PH_m(k) \tag{2.7}$$

$$H_m(k) = \begin{cases} 0, & f(m+1) < k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k \leq f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \leq k \leq f(m+1) \end{cases} \tag{2.8}$$

Once the Fbank features are computed mean normalization can be applied by simply subtracting the mean of all values to improve the signal to noise ratio (SNR).

### 2.2.2 Mel Frequency Cepstral Coefficients

Mel Frequency Cepstral Coefficients are the most widely used input features to an acoustic model. It is an extension of the Mel FilterBank features which is being used by speech researcher for long time. In general, the first 12 coefficient of the Mel Frequency Cepstrum along with the energy of each frame in total of 13 is considered as the MFCC features. It can be computed applying a Discrete Cosine Transformation (DCT) to the FilterBanks where the initial 12 are taken and the remaining are discarded. Before applying DCT, the Mel spectrum is converted to the natural log scale. The equation below can be used to convert any frequency to Mel scale.

$$mel(f) = 2595 \log_{10}(1 + f/700) \tag{2.9}$$

The co-efficient can be calculated using the following equation,

$$c(n) = \sum_{m=0}^{M-1} \log_{10}(s(m)) \cos\left(\frac{\pi n(m-0.5)}{M}\right) \tag{2.10}$$

The reason behind MFCC is so popular is because standard Machine Learning algorithms are susceptible to highly correlated features like FilterBank. Adding an extra step of DCT decorrelates the features while conserving the same information. MFCC features are not well suited in a noisy background and not considered good for generalization problems.

### 2.2.3 Perceptual Linear Prediction

Another important feature used in this study is the Perceptual Linear Prediction (PLP) coefficients. PLP was first proposed by Hermansky et al [33]. It is quite similar to the MFCC features such as it also implies an equal loudness on the preemphasis part, however, it uses a cube root compression in place of the log compression. A key difference with MFCC is that it uses Linear Predictive Coding to compute the final coefficients. The process of extracting PLP features is shown in Figure 2.3. At first, a Short-Time-Fourier Transform (STFT) is computed for every frame to transform the signal into frequency space. Then the power spectrum is estimated from the power of the complex valued output which gets through the Mel frequency filterbanks. Then, the loudness equalization is achieved by amplifying the power of high frequencies similar to MFCC. Then a cubic root is taken to transform the loudness to intensity and normalized using Linear Predictive Coding (LPC).

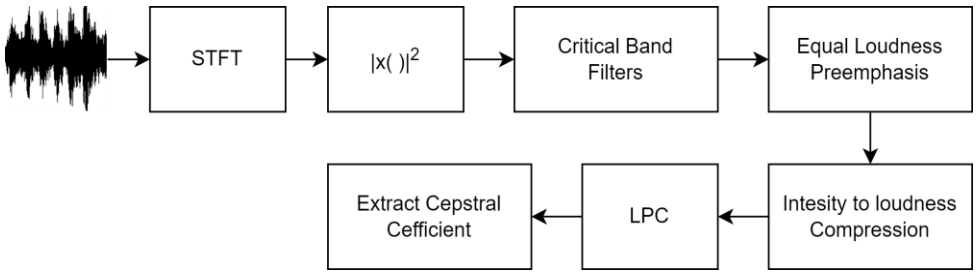


Figure 2.3: Computing Perceptual Linear Prediction (PLP) features

Finally, the PLP coefficients are computed by solving a set of recursive equations[34]. PLP is considered to be robust in a noisy environment and sometimes shows better results than other features.

### 2.2.4 Fundamental Frequency Feature/ Pitch Feature

The tones of a speech is related to the fundamental frequency or pitch of the sound which makes it highly effective for modeling tonal languages [35]. It can be extracted by computing the autocorrelation of a signal on a single frame or over a number of frames. There are different algorithms for pitch extraction. The method used for this research is associated with finding the lag values maximizing the Normalized Cross Correlation Function (NCCF) [36]. Instead of hard ruling over the voiced and unvoiced frames it uses the Viterbi algorithm to interpolate the unvoiced section. Therefore, a pitch is assigned even for unvoiced frames of the signal which helps it better fit with the standard GMM-HMM pipeline.

The pitch feature is sometimes processed further to use as a feature. For example, the Probability of Voicing (PoV) is often combined with the pitch which is used for anticipating a voice or unvoiced part. Additionally, the log of a normalized subtraction of the mean used for smoothing the output of the pitch. Apart from the PoV and pitch, a third component is also generated by computing the delta log pitch from ±2 frames of unnormalized log pitch [36].

In this study, the pitch feature is used coupling with standard features like MFCC, PLP or Fbank. More details are provided in Chapter 3. As per past studies, it was found to have better performance not only for the tonal languages but also a significant gain for non tonal languages [36].

### 2.2.5 I- Vector

I- Vectors are features extracted by adapting with an existing system generally based on a Universal Background Model (UBM) or Gaussian Mixture Model. I-vectors are well utilized for Hybrid DNN-HMM models in addition to standard MFCC features. It helps improving the speaker recognition or adaptation problem in speech recognition. In this research, the feature is also used for training DNN based models.

The standard procedure for extracting I-Vectors follows by forming a supervector stacking the means of standard feature vector taken from an audio sample adapting with a prior model [37]. Provided a supervector $S$ formed with respect to a UBM or GMM with a mean supervector $M$, the equation for I-Vectors is given by,

$$S = M + Tw \tag{2.11}$$

Here, T is a matrix that defines the low dimensional space in Total Variability Space (TVS) based modeling and $w$ is the extracted I-Vector. As it is a low dimensional representation of the speech signal, it still contains both speaker and channel information. Therefore, discriminative classifiers are used to discard unwanted information. Usually, a Linear Discriminative Analysis (LDA) followed by a Probabilistic LDA (PLDA) is applied on a length normalized I-Vector to reduce the dimension.

## 2.3 Acoustic Feature Transforms

This section will describe some of the feature transformation techniques used for improving the features.

### 2.3.1 Cepstral Mean and Variance Normalization (CMVN)

Due to different recording conditions and recording materials such as microphone, DVR or background chattering recordings often get noisy. CMVN is used to normalize such noise. These noises regarded as convolutive noise in the time domain which convert to additive noise in the cepstral domain [35]. CMVN normalizes the noise by removing the mean and variance from each coefficient. This technique is computationally inexpensive and widely used for robust speech recognition. Although in general it improves the overall speech recognition however, it found to be degrading the performance for shorter utterances. It happens as all the utterances are transformed to have a zero mean and variance, therefore due to inadequate data for the parameter estimation in shorter utterances useful information might get lost[38].

### 2.3.2 Delta and Delta-Delta Features

In HMM based modeling it is assumed that each of the observations are independent from each other. But, when the adjacent frames of the audio signal are highly correlated it might not hold. Besides, phonemes are often dependent on the dynamics of the features over time. The idea of Delta features is to capture the dynamics of the original feature by taking the differential from the previous and the next frame. It can be computed using the following formula,

$$d_t = \frac{\sum_{n-1}^{N}( C_{t+n} - C_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{2.12}$$

Here, $d_t$is the delta coefficient for frame t and $C_{t\pm n}$ are the original features.

To capture the dynamics even better a second derivative is also taken from the delta features is known as delta-delta. It can be computed using the same equation where the original features are replaced by the delta features. Both delta and delta-delta are usually coupled with the original feature to create an extended feature vector.

### 2.3.3 Dimensionality Reduction and Likelihood Maximization

Features in traditional ASR are often correlated to each other. Besides as the number of dimensions increases it also requires more data point to build an efficient model as we know from the curse of dimensionality in machine learning. Thus, its common to use algorithms such as Principal Component Analysis (PCA) or LDA for dimensionality reduction. PCA transforms the data to fewer dimensions with maximum variation in an unsupervised manner. In contrast, LDA finds a linear combination of features that describes the best class separation. In traditional ASR, the hidden states of the HMM used as class labels. Another linear transformation usually coupled with LDA is Maximum Likelihood Linear Transform (MLLT). It transforms the features globally to maximize the likelihood at a frame level for the model. Usually, LDA and MLLT are estimated together in a GMM-HMM system where a model is trained for a number of iterations and certain iterations include the estimations of LDA-MLLT.

### 2.3.4 Speaker Adaptive Training

In order to improve the performance of a speech recognition system it needs to adapt to the individual user. It is possible to train the system for that particular speaker, however in that case it will require large amount of data from that specific user to train the system. To avoid this issue HMM based systems use Feature Space Maximum Likelihood Linear Regression (fMLLR). It is an affine transform which works similar to the MLLT but in a speaker adaptive way. First, a speaker independent model is trained and the best path is computed using Viterbi decoder. Then, using the Viterbi path the parameters for fMLLR is estimated for each speaker. Finally, the features are transformed according to the adaptation of fMLLR and the model is retrained on the transformed features. The recognition procedure also follows first pass of feature transformation by the fMLLR and second pass of recognition from the transformed features of the speech signal.

## 2.4 Lexicon

In traditional ASR system each word is modeled in terms of smaller segments to avoid data sparsity as well as decoding unseen words. It also acts a bridge between words and pronunciation model. The smaller segments are usually represented by phonemes or graphemes. Phonemes are simply the smallest contrastive unit of spoken system. On the other hand, graphemes are the smallest contrastive unit for written system. The list of words with their pronunciation represented with phonemes is known as the pronunciation dictionary or a lexicon. An example of a word "Green" represented with phonemes would be,

green G R IY N

The lexicon is generally prepared manually by expert linguist. However, due to different accent there can be alternative pronunciations for the same word or same pronunciation for different words and its always a challenging task for diverse languages. Therefore, sometimes if a pronunciation dictionary is not available researcher choose to supplement it with graphemes, characters or rule based sub-word units. The relation between graphemes and phonemes depends on the language. For example, Finnish and Spanish has a regular relationship, on the contrary English or French has an irregular relationship [29]. The relationship might deviate as well based

on dialect, second language learners etc. More details of how this research uses lexicon is provided in Chapter 3.

## 2.5 Acoustic Models

The acoustic model represents the relationship of an audio signal with the corresponding phoneme or subword unit [39]. It is generally built specific to a certain language with one or different dialects, however multilingual acoustic models are also quite common in recent times [40]. The states of HMM are to represent the acoustic units in an acoustic model. The detail of HMM modeling structure is described below.

### 2.5.1 HMM Based Acoustic Models

An HMM based acoustic model $\lambda$ can be defined with five following elements.

- A set of states $S: S_1, S_2, \dots S_N$ where at any discrete moment the system will be in any of those states. In an HMM the states are hidden, and the status of a particular state depends on the observables.
- A discrete set of phonemes $V: V_1, V_2, \dots V_M$ for possible emission
- A state transition probability distribution matrix $A$, here the probability of moving from state $S_i$ to state $S_j$ is given by $a_{ij}$
- An emission probability distribution matrix B where the probability of emitting any symbol $V_k$ in any state $S_j$ is given by $b_j(k)$
- An initial probability matrix $\pi$ that assigns the probability of each state $S_i$ at their initial state.

There are two assumptions to be considered in a first order HMM. The first one is the Markov property which can be represented as,

$$P(S_t|S_1^{t-1}) = P(S_t|S_{t-1}) \tag{2.13}$$

Here, $S_1^{t-1}$ corresponds to the state sequence $S_1, S_2, \dots S_{t-1}$. Thus, the assumption is given as the probability of any given state only depends on its previous states and not any other states before that.

The second assumption is the output-independence characteristics,

$$P(X_t|X_1^{t-1}, S_1^t) = P(X_t|S_t) \tag{2.14}$$

where $X_1^{t-1}$ corresponds to the output sequence $X_1, X_2, \dots X_{t-1}$. This assumption states that, at any given time $t$ the probability of an emitted symbol only depends on state $S_t$ and independent of any past observation. Below is a representation of an HMM of three states with its transition and emission probability (Figure 2.4).

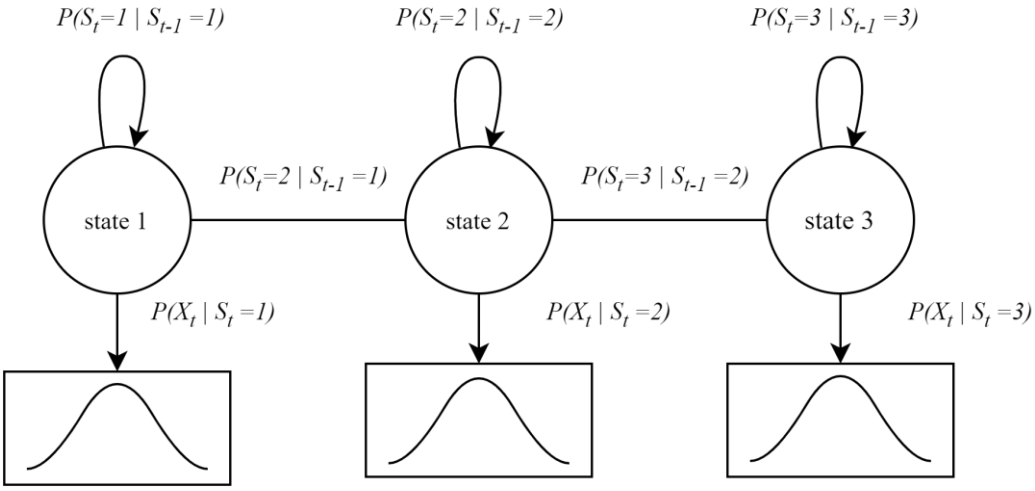


Figure 2.4 : Example of three states with left-to-right HMM and the emission probability

The HMM based acoustic modeling can be broken into three following problems in an ASR.

1. The Evaluation Problem: For a given HMM, the task of determining the probability of a particular sequence of visible states was generated by the model. This can be solved using either the Forward or Backward algorithms [41].

2. The Decoding problem: If a model and set of observation is given the task of determining the most likely sequence of hidden states which resulted into the given observations. It can be efficiently solved using Viterbi algorithm [42].
3. The Learning problem: For a given HMM model $\lambda = (A; B; \pi)$ with a set of observations $O$ for training, the task is to find the best parameters that maximize the probability of observing $O: \lambda* = \underset{\lambda}{\operatorname{argmax}} P(O|\lambda)$ . The Baum-Welch algorithm which is a special case of the expectation-maximization technique is usually used for tuning the model parameters.

The emission probability distribution matrix $B$ can be estimated using Gaussian Mixture Model or DNN. Based on the emission matrix estimating method the ASR systems are commonly names as GMM-HMM or DNN-HMM. Detail of the GMM and DNN modeling is provided in the next section.

#### 2.5.1.1 Gaussian Mixture Model

GMM is the most popular way of modeling the emission probability distribution for HMM based ASR. To determine the distribution only two parameters are considered which are the mean $\mu$ and the covariance $\sum$ . The components of the GMM are given by Gaussian pdfs. The likelihood for a given state $S_j$ can be calculated using the following equation,

$$b_j(x) = \sum_{m=1}^{M} c_{jm}(X|\mu^{(jm)}, \Sigma^{(jm)}) \tag{2.15}$$

Here, the $c_{jm}$ represents the mixture weights for a Gaussian $m$ of state $S_j$ and the priors need to follow the constraints of a valid probability mass function given below,

$$\sum_{m=1}^{M} c_{jm} = 1, c_{jm} \geq 0 \tag{2.16}$$

#### 2.5.1.2 Deep Neural Network

In earlier stages of ASR, Artificial Neural Networks (ANN) were used in combination with HMM to build speech recognition systems instead of GMMs. The first ANN-GMM model was proposed in 1990. The ANN is trained for predicting the posterior probabilities to generate the pseudo likelihood for each state of HMM. Due to single layer composition, this kind of models were not good enough and GMMs were leading the ASR research. DNN is a basically feed forward ANN with more than one hidden layer. Each hidden layer uses a non-linear activation function generally a Rectified Linear Unit (ReLU) for mapping the weights to a standardized state [43]. For multiclass problems such as ASR, a SoftMax nonlinearity is used on the output layer to normalize the probability distribution over the classes.

DNNs are trained using forward and backward propagation of the derivatives between the training output and expected output. The difference calculated via a cost function usually cross entropy loss given below where $p$ is the target probability of each symbol,

$$C = -\sum_{j} d_j log p_j \tag{2.17}$$

Due to the amount of training data updating weights after a whole walkthrough of the data is not encouraged, instead dividing the data into batches results in better training. Different optimizers such as Adam, SGD etc are also used to smoothen the gradient. As a deeper network takes more and more time to converge, unsupervised pretraining like Restricted Boltzmann Machine (RBM) or Discriminative Pretraining (DPT) studied by the researcher which allows greedy layer-by-layer training to initialize DNNs. Perhaps many hidden layers were able to be trained to build more sophisticated acoustic models which led to large improvements on the performance. RBMs are undirected graphical models with a number of nodes constructed from layers of observed stochastic units also known as visible units v and a layer of latent random units or hidden units h [15]. The joint probability of v and h is given by,

$$P\ (v, h) = \frac{1}{Z_{h,v}} e^{E(v,h)} \tag{2.18}$$

Where $Z_{h,v}$ is a normalizing partition function. The visible units are real-valued Gaussian distributed whereas the hidden units are binary valued Bernoulli distributed.

DPT works such, the first layer of DNN is trained with full convergence using the frame labels for every state of the HMM, then the outer layer which is usually a Softmax is replaced by another hidden layer with random initialization. However, only the second layer is updated during training this time. The training continues to full convergence and again a new layer is added. Instead of single layer pretraining multilayer training is also used in some cases.

Since accurately modeling the temporal dynamics of speech requires capturing long term dependencies in the acoustic signal, therefore typical DNN in other words Multi Layer Perceptron (MLP) has been replaced by more effective architectures like Time Delayed Neural Network (TDNN) or Recurrent Neural Network (RNN). TDNNs use a modular and incremental design for creating larger networks from sub components [23]. DNNs use fixed dimensionality vector for modeling for mapping sequential data which might not be ideal for speech. Therefore, a different type of architecture such as RNN or Long Short Time Memory (LSTM) has been utilized. RNNs use dynamically changing contextual windows over all the sequence history. LSTMs are a special type of RNN which avoid the vanishing/exploding gradient problems of RNN and produce even better results. Although RNN and LSTM has a better modeling structure, it always requires more data than standard DNNs, hence DNNs are more effective in low resource scenario[44]. This thesis uses both TDNN and a combination of TDNN with LSTM for some of the acoustic models. The details of the models are provided in Chapter 3 and Chapter 4.

## 2.6 Language Model

An acoustic model generally outputs a probability over phoneme or word sequence that best fits with the audio data. The decoder of an ASR system takes the output of the acoustic model and corrects it using the prior statistics of the word or phoneme based on large textual data. A language model is a statistical tool which is used for analyzing patterns in a human language [44]. LMs are trained using large scale plain text for estimating the probability distribution of word sequences. There are two main methods for building Language Models currently in practice. One is the N-Gram language model and the other is the Deep Learning Based language model. More detail about the two are provided in the next section.

### 2.6.1 N-Gram Language Model

N-Gram is the most common method for language modeling. If the word sequence is $W = w_1, w_2, \ldots\ldots w_N$ the probability of a word can be computed using the chain rule below,

$$\begin{aligned} P(W) &= P(w_1, w_2, \ldots\ldots w_N) \\ &= P(w_N | w_1, w_2, \ldots w_{N-1}) \\ &= \prod_{i=1}^{N} P(w_i | w_1, w_2, \ldots w_{i-1}) \end{aligned} \tag{2.19}$$

The probability of a word can be derived using a relative frequency count from the word's history. However, the method is not practical due to the high variability of possible context. The N-gram language model instead limits the preceding word to $N-1$ while estimating the probability from large text data. The typical N-gram model uses an input of $N = 3$, in other words a trigram model. But Bigrams and Unigrams are also useful for isolated word recognition and other speech analysis.

During the testing phase, it is possible that some of the words are not enlisted within the vocabulary of the language model which might result in a zero probability of the whole sequence. To avoid such instances smoothing techniques are used. This technique reallocated some probability mass to previously unseen word sequences. Various smoothing techniques are available such as Kneser-Ney, Witten-Bell, Katz smoothing etc.

## 2.7 Forced Alignment

In a forced alignment process the speech is automatically aligned with its corresponding orthographic transcription generally at a word or phoneme level. Given a mapping of the graphemes to the phonemes (a pronunciation dictionary) and an acoustic model, the forced aligner takes the speech and identifies a sequence of phones which best fits the actual pronunciation [45]. Forced alignment is very similar to an ASR. But in an ASR, the model is given a list of words to search for, in contrast for forced alignment the model is given an exact set of transcription. This is the reason it's called a forced alignment.

In traditional ASR system, regardless the system is built on DNN-HMM or GMM-HMM, training the acoustic model requires the features to match with the corresponding label. For a typical model if the features are MFCC and labels are phonemes we need to know which feature vector corresponds to which phoneme for estimating the parameters of the model. However, our training data generally consists of a lot of noise and silent parts without actual speech. Therefore, the audio signals are broken into shorter segments and aligned beforehand with the transcription before training. In earlier times, the alignments were produced by the linguists manually which takes a lot of time as well as expertise over the language. Nowadays a lot of tools are available to automatically align speech to text, however, depending on the target language additional training or development of toolsets are necessary to get more accurate output.

During the training of ASR, a forced alignment is also applied in some of the iterations to realign the data with its transcription. The process follows as an initial model is built using the available noisy data. Although the labels are not very clean at this stage, yet the model learns something for generating a likelihood score for each frame. The scores might not be very accurate however as the model already knows the labels it can be used to realign the data. This is an iterative process where, as the training progresses the model produces better alignment and with better aligned training data the model learns better parameters.

Apart from speech recognition, the forced aligners are also used for isolating speech sounds, keyword search, language documentation, phonetic analysis etc. In this thesis, we developed a forced aligner as part of our baseline ASR system. The modeling detail and the evaluation is provided in Chapter 3.

## 2.8 End to End Speech Recognition

Despite the success of conventional ASR systems, it relies on very sophisticated pipeline of complex multi-module structures and hand engineered processes. There are several weaknesses in the conventional ASR system,

- The architecture is module based and each module needs to be optimized individually
- As the modules are trained separately it may contain incoherence

- Due to separate training of each module it's also difficult to get high performance as well as making the development process complex
- The decoding goes through each individual module making it slow and complicated
- It involves several hand engineered components like lexicon design, label alignment
- The assumptions made for HMM are also not valid for speech signal

Therefore, to overcome such weaknesses substantial amount of research have been done with End to End (E2E) ASR systems over the years [46]–[50]. E2E systems integrate the different components of a traditional ASR system i.e. the acoustic, pronunciation and language model into a single network of E2E model which allows the joint optimization of all the components at the same time. It also reduces the requirement of any special type of feature engineering and also able to train on raw audio signal. E2E system works as a sequence to sequence (seq2seq) model which directly converts the acoustic sequences to word sequences. However, to build an effective E2E system there are some major challenges,

First, there has to be a way for building large, labeled training datasets as most E2E systems are mainly different combinations of deep learning networks. This challenge has been addressed by sharing knowledge among different corpora of languages (transfer learning) and methods of producing synthetic data (data augmentation). However, HMM based architectures are still outperforming in most low resource cases.

Second, the networks must be large enough to properly accommodate the knowledge from all of this data. Recent development of many architectures address these issues, for example Self Supervised Representation Learning (SSRL) models are able to capture the knowledge of huge corpora of over a hundred languages in an unsupervised fashion [51].

Another challenge is handling the alignment of the text labels with the input speech signal. This problem is also addressed by a group of researchers who developed Connectionist Temporal Classification (CTC) method [52]. CTC enables easily training large datasets skipping this alignment part.

E2E architectures can be divided into three major groups for ASR systems including Attention based Encoder-Decoder systems, CTC based approaches and RNN-Transducers [53].

### 2.8.1 Attention based Encoder-Decoder systems

Attention based Encoder-Decoder (ED) networks were first introduced in the field of neural machine translation (NMT) [54]. It has three main components, the encoder, attention mechanism and the decoder [55]. A schematic representation of the architecture is given in Figure 2.5 [53].

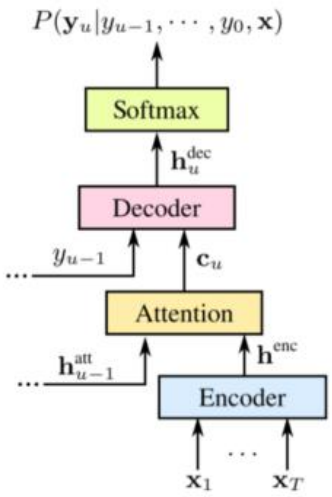


Figure 2.5: A schematic representation of Attention based Encoder-Decoder architecture

For speech recognition, the encoder works similar to an acoustic model. It takes the features $x = x_1, x_2, \ldots . x_T$ and transforms them into a hidden representation $h^{enc} = h_1^{enc}, h_2^{enc} \ldots . h_T^{enc}$. At a given time step $u$ the decoder network uses the label $y_{u-1}$ predicted at the previous time step with the context vector $c_u$ generated from the attention mechanism to output the logit $h_u^{dec}$ for the current step. Then, the SoftMax layer converts the logit to a probability distribution $P(y_u, y_{u-1}, \ldots . y_0, x)$ conditioned to the previous predicted labels and the input sequence. As the output label is conditioned to the previous prediction it continues to the output as long as the sentence end token is not predicted. As the attention depends on all the encoded input, the system needs to wait for the full sentence to be processed before it can start decoding. This limitation

makes the system unusable for online decoding or live decoding. Besides this kind of system often degrade for longer texts used in the training sequence.

#### 2.8.1.1 Transformer

Transformer is a DNN architecture introduced very recently in 2017 built for sequence transduction [56]. The task is similar to seq2seq models however this architecture entirely depends on attention mechanism without applying any recurrent or convolution operation. Transformer consists of an encoder and a decoder where the encoder takes the input sequence, converts it to a vector and then transforms it using attention mechanism. On the other hand, the decoder takes the transformed vector and decodes into the output sequence. Both the encoder and decoder has an almost identical internal structure but different weights and biases. Figure 2.6 shows a high-level illustration of the Transformer architecture.

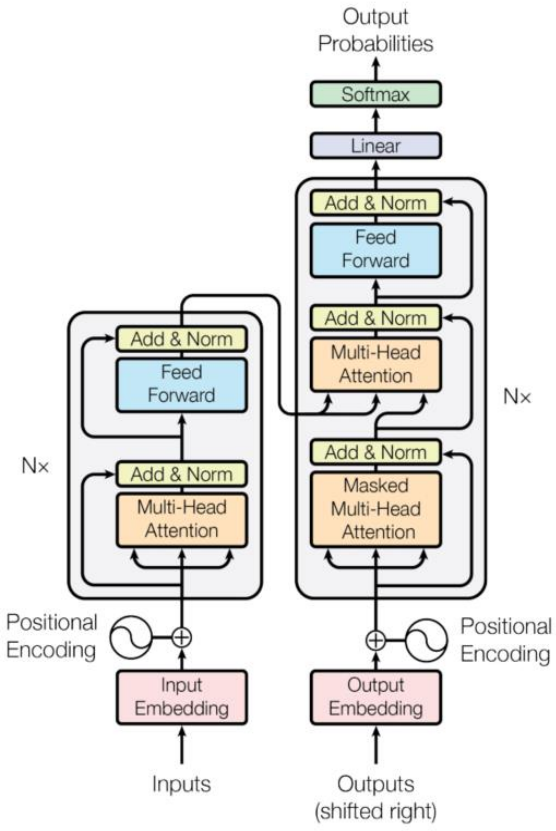


Figure 2.6 High-level Illustration of transformer architecture

The encoder has two main blocks, a self-attention sublayer consisting of multihead attention mechanism and another sublayer of a feed forward network. Both sublayer contains a residual connection and a layer normalization [56]. Self-attention is a process where the transformer attends to the other labels of the input sequence during encoding of a specific label. An attention head takes three matrices including the query $Q$, Key $K$ and the value $V$ computed from the input matrix. The output of a single attention layer can be calculated using,

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.20}$$

In practice the transformer utilizes multiple self attention head parallelly whereas each contains their own query, key and value. The output matrix of self attention sublayer is obtained by concatenating the output of each self attention head which then fed into the feed forward network.

The decoder block has similar sublayers like the encoder with one additional self-attention sublayer between the first self-attention layer and the feed forward network. The middle layer computes its query matrix $Q$ from the first self-attention layer but takes the key $K$ and value $V$ from the output of the encoder. The first sub layer has a masking process before the SoftMax layer for restricting it attend into any future position [53].The linear layer on top of the decoder converts output of the decoder to logits which then transformed to probabilities by the SoftMax layer like any other ED architecture. Finally, the label of highest probability is selected and fed back into the decoder to decode the next label. The process continues until the end token is not generated.

### 2.8.2 CTC-based Architecture

Connectionist Temporal Classification was first proposed in 2006 allowing alignment free training of E2E models [52]. It is much simpler and faster than any other E2E techniques as it requires only the encoder to be trained. It is also faster in decoding due to its token-level iterative decoding system.

The encoder part of any CTC architecture works similar to Encoder-Decoder systems where the model takes an input sequence and transforms it to encoded logits. The logits are then

passed through a SoftMax layer to generate probability distribution conditioned to the input sequence. The labels of the CTC architectures are encoded with a special blank token. This gives the network the flexibility to output label for any part of the input sequence. It also allows the network to generate higher probability to the correct labels at a specific time and just generate higher probability to the blank token the rest of the time. For an input sequence $X$ and output sequence $Y$ the objective of CTC can be written as,

$$p(Y|X) = \sum_{A \epsilon A_{X,Y}} \prod_{t=1}^{T} p_t(a_t|X) \tag{2.21}$$

The loss for CTC is calculated over the entire unsegmented input sequence to the target sequence. Figure 2.7 shows how CTC finds the true labels aligning the input to output [57]. During the alignment process it is assumed that the output labels are independent of each other although it's not true for speech recognition. Thus, the outputs can be improved by incorporating a language

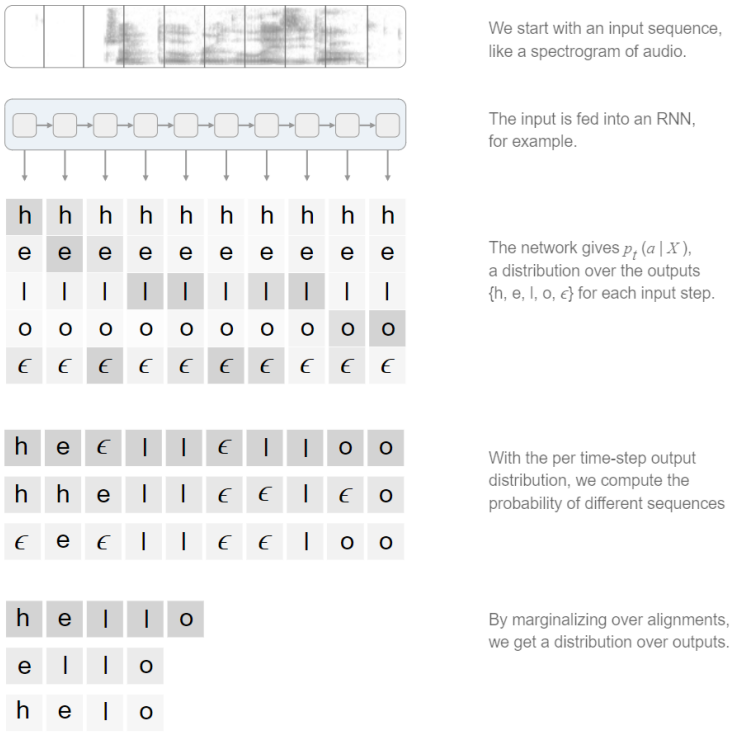


Figure 2.7: CTC alignment process for ASR

model built at a character level. However, the language model needs to be trained separately which therefore may not be an ideal E2E system.

### 2.8.3 RNN Transducer

The assumption for conditional independence between each label while modeling with CTC architectures results lower performance. This problem was first addressed in RNN-Transducer without any external language model. The RNN-Transducer has an encoder network similar to other E2E architecture with an additional prediction network. The prediction network can be treated as a language model which can be optimized jointly with the whole ASR system. The prediction network produces a vector $p_u$ based on all previous predicted labels. The vector is then fed into a network jointly with the output $h_T^{enc}$ from the encoder to compute the logits. The rest of the process follow similar to other E2E networks. The computation of the logits $Z$ can be given by,

$$z_{t,u} = w_3 h_{t,u}^{joint} + b_2 \tag{2.22}$$

Here, $w$ and $b$ are just the weights and biases of the network. Due to larger size and more parameters RNN-Transducers takes higher memory while training thus computationally more expensive then other E2E architectures.

### 2.8.4 Self-Supervised Representation Learning (SSRL)

Self-Supervised Representation Learning is a method of training DNN without any annotated labels. It is a special form of unsupervised learning where instead of clustering or grouping it learns a representation of the data using psudeo labels generated from the data itself. It is known as the 'pretext' task or pretraining which enables using part of the data to solve an unsupervised problem through supervised techniques. In ASR, the labels are created by masking some part of the input and the model focuses on recovering the missing part of the input. This way the model learns a very powerful representation of the input data which can be used later for many downstream task such as speech recognition, translation, speaker recognition etc.

SSRL has become a very active research topic in the ASR system. In recent year, there have been a lot of models introduced based on SSRL such vector quantized variational autoencoder (VQ-VAE) [58], Wav2vec [59], Wav2vec 2.0 [60], Mockingjay [61], Audio ALBERT [62] etc.

## 2.9 E2E Frameworks

This section will describe about the End to End frameworks used in this thesis. There are mainly two frameworks including Deep Speech 2.0 and Wav2vec 2.0 used for this research. Some other variation of the Wav2vec 2 also have been used which we describe more in Chapter 4.

### 2.9.1 Deep Speech 2.0

Deep Speech is an RNN based E2E ASR system developed by Baidu [63], [64]. It is a complete E2E model which does not model noise, speaker variation etc. using separate transformation technique or hand crafted methods. Deep Speech uses spectrogram as input feature. The original deep speech architecture has five layers where first 3 are non-recurrent and the fourth one is a bidirectional recurrent layer. The fifth layer is a non-recurrent layer, and the output is a SoftMax layer which generates the probability for each characters in the language. Figure 2.8 shows the structure of RNN model in Deep Speech [63]. As a single recurrent layer is probably not enough to capture the diverse representation of the data in their second version Deep Speech 2.0, they increased the model capacity using up to 11 layers of bidirectional recurrent layer and convolutional layer. Instead of simple RNN they utilized the GRU and optimized with SortaGrad. Both of the models exploit CTC for an alignment free training. They mainly experimented the model with large data from English and Mandarin. However, their use of data augmentation techniques such as inducing noise, inflecting pitch explains the possible use in low resource languages. In this thesis we only implemented the Deep Speech 2 and tuned the hyper parameters like number of layers and choice of hidden units tailored to our language. More details of the training configuration and experiments are provided in Chapter 4.

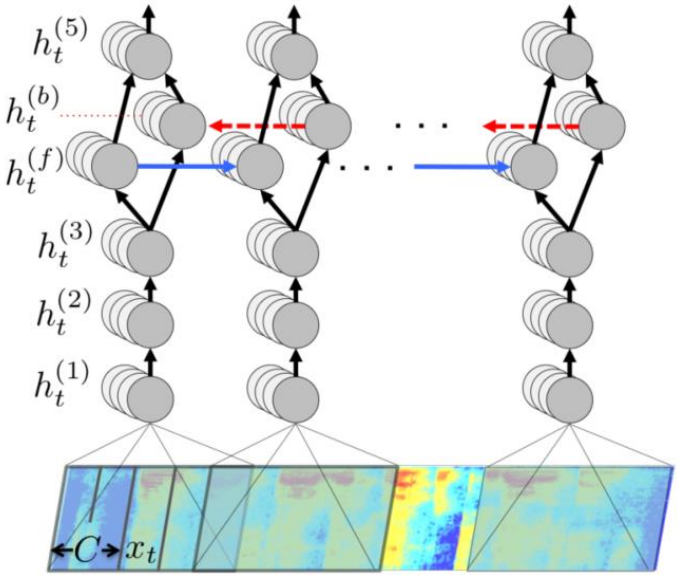


Figure 2.8 the structure of RNN model in Deep Speech

### 2.9.2 Wav2vec 2.0

Wav2Vec [59], [60] combines the power of SSRL with CTC to build a strong ASR framework. It has a multilayer feature encoder composed of CNN which takes raw audio $X$ and outputs latent representation $Z$. Figure 2.9 demonstrates a high level architecture of wav2vec 2.0. The latent variables go through a quantization module to get finite set of discretized speech representation $q$. At the same time $Z$ are partially masked and fed through a transformer network to build the contextualized representation $C$ over the raw input sequence $X$. Wav2Vec utilizes the concept of contrastive representation learning previously implemented in BERT [65] where the similar samples are near to each other and different ones are pushed further from unsupervised data. The masking is done using three different process,

- The time-step replaced with a mask-token.
- The time step replaced with another random time-step.
- Cutting off a time-step with no defined mask.

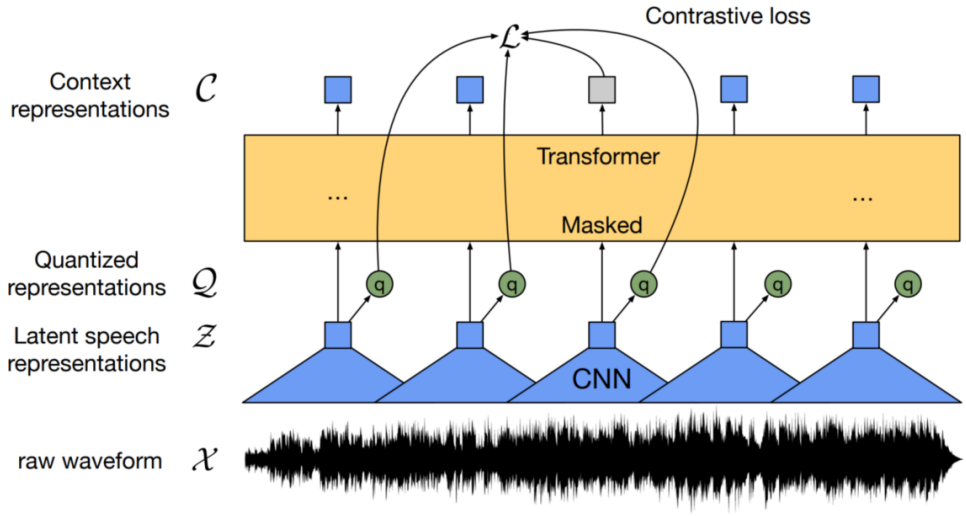


Figure 2.9 High level architecture of wav2vec 2.0

In this thesis, we used three different pretrained wav2vec 2.0 model. These models are cross lingually trained from 53 up to 128 languages. The idea is sharing these representations benefits the possibility to use features from a different language without pre-training the model again on a new language. More details of the training setup and the internal operation of the method is provided in chapter 4.

## 2.10 Evaluation Matrices

The performance of ASR is typically evaluated in terms of Character Error Rate (CER) or Word Error Rate (WER). There is also other metric such as Out of Vocabulary (OOV) count and Perplexity for measuring individual performance of the language model. In this thesis, the WER is adopted to track the performances of all the ASR models. The details of WER calculation is given below.

### 2.10.1 WER

The output of an ASR is a hypothesis of the speech signal. WER calculates the distance between the hypothesis and the reference or true transcription in a lowest number of modification required for correcting one into the other in a percentage measurement. The measurement is

calculated using the Levenstein distance [2]. First a dynamic string alignment is performed to align the output from ASR to the reference text. Given the optimal alignment, the following errors are counted:

- Substitution: If a word is misrecognized
- Deletion: If the word is not present in the hypothesis
- Insertion: If a word is inserted in the hypothesis which is not spoken

Finally, the WER is calculated from the counts of the errors using the following equation,

$$WER(\%) = \frac{\#substitutions + \#insertions + \#deletions}{\#words(reference)} * 100\ \% \tag{2.23}$$

### 2.10.2 CER

The CER is a similar measure as WER. Instead of counting the errors at a word level for CER the errors are counted at a character level. The process of alignment and error calculation follows the same procedure as WER. It is mostly used for the models where the character is used as a modeling unit, so it reflects a more in depth reflection of the performance for some cases. Besides it also provides some insights of the wrong words in terms of closeness to the original transcripts.

# 3 Automatic Speech Recognition for Documenting Critically Endangered Athabascan Language

This chapter includes a manuscript entitled "Automatic Speech Recognition for Documenting Critically Endangered Athabascan Language" by Zarif Al Sadeque and Francis M. Bui, which is under preparation for submission. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). The background concepts related to different components of the modeling technique, feature extraction, etc., are explained in Chapter 2.

## 3.1 Abstract

Endangered language documentation is a key process of preserving the language along with its history and cultural heritage. Automatic speech recognition (ASR) can be a very effective tool to expedite such endeavor. Although recent advancements in conventional and End to End (E2E) ASR systems have shown promising performance, it still remains a challenging task for endangered languages due to extremely limited resources. Languages with rich morphology and complex structure compose further obstacles. This study includes a critically endangered Northern Athabaskan Language 'Upper Tanana'. It is one of the highly spoken native languages in the Alaskan Athabascan region, now at a critically endangered stage. In this research, we focus on developing an ASR system from scratch including different modules like pronunciation lexicon, acoustic model and language model. Here, firstly we explore different feature sets and evaluate the alignment accuracy at the word level on selected features. Secondly, we investigate the applicability of some popular algorithms for acoustic modeling based on traditional HMM and Deep learning. One major challenge of this research involves an extremely small data size of about 1 hour and 9 minutes of curated speech with no pronunciation dictionary. The purpose of this study is to obtain a best suited combination of feature set and modeling techniques for Upper Tanana to develop an efficient ASR system in support of linguistic documentation. Experiments show that due to the limited data constraint traditional GMM-HMM methods perform better than deep hybrid methods. Besides, adding tonal features e.g. pitch along with standard MFCC can slightly better the Word Error Rate (WER). Different n-grams are also tested and reported in conjunction with

different feature sets. Although the approach is evaluated only for Upper Tanana, but it can be applied to any under resource language as it addresses most of the primary aspects of low resource speech recognition.

## 3.2 Introduction

Documentation of endangered languages has become a crucial need recently for both linguists and native speakers. It has been predicted that, almost half of the world's 7000 languages might not outlast till the next century [1]. Linguists have already started developing diverse language archives, corpora, lexicons etc for the world's major languages. However, for endangered languages, these archives are very limited. Traditional documentation procedures involve interlinear time-coded transcriptions including aligned parses & glosses and sometimes a translation as well [66]. However, endangered languages suffer from a shortage of standard orthography and linguistic expertise in most cases. Besides the fact, this process is also quite time-consuming; even for an expert linguist or a native speaker, just transcribing an hour of audio can take approximately 30 to 50 hours [67]. This challenge in transcription is known as the "Transcription bottleneck"[68]. It is also imperative that the number of endangered languages is growing and only a few linguists are available with the specialization needed for this transcription task. Hence, the amount of resulted transcripts are oftentimes much smaller than one fourth of the actual recorded audio data [67]. The state-of-the-art ASR technology has attained astounding performance in meeting the human level transcription [69]–[71] which can support overcoming the bottleneck. Especially this can assist as a tool to provide a draft transcription for the linguists to easily carry out the documentation projects.

In past years, ASR technology has succeeded remarkably for most of the highly spoken languages. The advancement of deep neural network (DNN) along with End to End (E2E) frameworks has significantly reduced the word error rate (WER) for languages like English or Mandarin. However, the limitation of available data makes it less compatible with endangered languages. In contrast, Standard Hidden Markov Model with Gaussian Mixture Model (HMM-GMM) frameworks have been more popular among researchers for Endangered language documentation [1], [72], [73]. It's also found to be more precise and robust in case of small data settings [74]. However, HMM based models require an extant language lexicon and a lot of

linguistic resources, to begin with. This work focuses on a critically endangered Athabascan language Upper Tanana. It is also considered a morphologically highly complex language [75]. Unlike common languages or some other under resourced languages where pronunciation dictionaries and standard coding scripts like Kaldi recipes are available, Upper Tanana has no freely accessible pronunciation lexicon or large annotated corpus. This is one of the main challenges for this research to build the ASR system from the scratch. Recently semi-supervised [76] and self supervised models [51] have also gained attention from the ASR research community. However, as this is the first study so far regarding ASR for Upper Tanana, such technologies are out of scope at this moment due to the dearth of data. Therefore, this research currently uses the traditional HMM based models including both GMM and Hybrid-DNN networks as a baseline model for any of the future studies related to this language. All the models developed for this research use KALDI RECOGNITION TOOLKIT [77]. Essentially, we develop an easily understandable Kaldi recipe for Upper Tanana that will create interest to more researchers for coming forward working with endangered languages. Besides, Kaldi comes with a lot of prebuilt recipes as well as online resources which we took advantage of to expedite our research.

## 3.3 Contribution

The aim of this work is to report the initial steps and findings towards developing the ASR system for Upper Tanana from a non-expert linguistic perspective. We build a Kaldi recipe demonstrating the complete procedure of Acoustic modeling, lexicon design and language modeling. Instead of manually building a pronunciation lexicon we utilized open accessed G2P libraries to automate the lexicon development. We investigate all the models from both quantitative and qualitative point of view including different feature sets. The contribution of this research is summarized as follows,

- Extending ASR methods to Upper Tanana, an endangered northern Athabascan language
- Examining the diverse ASR modeling technique for resource constrained language. Results show that state of art Hybrid DNN models are unable to meet the performance of typical GMM-HMM, even with a difference as small as 0.27% WER.

- Input feature comparison for Upper Tanana. We find that even though the dialect lacks tonal variation, a small improvement of 1% WER can be made by combining pitch with other features for GMM-HMM models.

The rest of the Chapter is structured as follows. We revisit some existing ASR studies for endangered language documentation in Section 3.4. Then we explain the background of the Upper Tanana corpus including data collection, recording materials and some linguistic details in Section 3.5. Followingly the Experimental Details and Results are reported in Sections 3.6 and 3.7 respectively. Finally, the conclusion is given in Section 3.8.

## 3.4 Existing ASR studies for Endangered Language Documentation

A number of efforts have been made for automating language documentation using ASR technology. These involve different endangered languages as well as different modeling techniques. Some of these studies are part of long-term ongoing projects. There were few attempts on Seneca over last three years utilizing both traditional HMM based models [1], [74] and Deep E2E models [78]. They made a detailed report regarding how it improves the overall process by assisting linguists using their ASR system [79]. Yongning Na, a SINO-Tibetan endangered language has been a part of similar research for over 12 years [73]. Earlier researchers mostly utilized CMU sphinx as the ASR tool and train it using monolingual or multilingual information based on similar languages [67]. In contrast, recent studies for Yongning Na exploited deep learning methods eg: LSTM along with connectionist temporal classification technique (CTC) to improve the result [5].

Several studies also published using state-of-the-art technologies based on transformers. Qin et al published a research on ASR for preserving the endangered Lhasa dialect of the Tibetan Language. They leveraged multilingual information from Bengali, Nepali and Sinhalese for pretraining a transformer and refining their model. Another study on Yoloxochitl Mixtec also used a similar technique using multiple transformers and an automatic transcription correction system [68]. Most of these studies have contributed to accelerating the documentation process over the years, however long-term research as such Yongning Na or technologies that require large amounts of transcribed data like Yoloxochitl Mixtec which used around 125 hours of annotated audio data

might not be feasible for many endangered languages. Depending on the severity of the extinction and the number of available speakers some languages are in need of urgent documentation. This work anchors on a similar case study of Upper Tanana where at most 1 hour and 9 minutes of transcribed data is available. It also has less than 50 speakers reported by *UNESCO Atlas of the World's Languages in Danger* [80].

To make it more practical and time convenient a recent project titled Elpis (Endangered Language Pipeline and Inference) has been conducted that also inspired this research. The project includes 16 endangered languages from Asia-Pacific region with data ranging from less than 1 hour up to 3 hours. The study also uses Kaldi to implement their pipeline and building the components. However, their pipeline requires the pronunciation lexicon developed beforehand or manually by the linguists. This might not be suitable for most endangered languages due to the of lack standard orthography or the availability of expert linguists. Apart from endangered languages, there are also numerous works on low resource or under resource languages [43], [81]–[85]. Although these languages might not be in danger for extinction, a lot of them share a similar kind of challenges e.g. scarce training data or complex morphology for the development of ASR systems.

## 3.5 Background of Upper Tanana Corpus

### 3.5.1 History of the Language

Upper Tanana is a Northern Athabascan Language which also used to be known by Nabesna earlier [86]. It is part of the Alaskan subgroup of the Northern Dene language family and closely related to Tanacross, Hän & Gwich'in languages [87]. But it has the most resemblance with the Tanacross language where the only difference is in tone marking. It is traditionally spoken in four communities of Eastern Alaska including Tetlin, Northway, Nabesna, Scottie Creek and also in the Beaver Creek of Yukon territory in Canada [88]. Each community has a slightly different dialect than others. Although it used to be one of the highly spoken languages in the region, however currently it only has less than 50 speakers. Most of the speakers are elderly, above their sixties. While some mid-age members of the community can partly converse in the language, the younger generation doesn't speak it anymore. Earlier the region had a fame for preserving its

culture and linguistic heritage which got changed in recent times. Therefore, it has been declared a critically endangered language by UNESCO.

### 3.5.2 Linguistic Background

Upper Tanana is considered a tonal language consisting of low tone and unmarked tone. Tonal contrast is the primary distinguishing factor between the five dialects of this language. While the other dialects have higher or lower tonal contrast, the Tetlin dialect has almost no contrast or hardly some vestigial tone [88]. The corpus of this study is based on only the Tetlin dialect. The Upper Tanana writing system was first developed during the 1960s by Paul Milanowski [86]. Originally the language has 13 vowels and 34 consonants in its writing system, however, the Tetlin system discerns only six vowels and a dipthong. Upper Tanana is distinguished from the other Alaskan Athabascan languages through the buildup of its stem vowels and the removal of stem-final coronal non-lateral consonants [89]. The linguistic documentation of Upper Tanana first started in 1929 comprising mostly animal names, body parts etc. However, an extensive research including the proposal of some literacy materials, an orthography and also a dictionary was conducted long after in 1961 by Milanowski[86]. Later different researchers studied the grammar, phonology and lexicon of this language and it's still ongoing.

### 3.5.3 Recording Settings and Materials

The corpus for this study is originally recorded as part of the expanded edition of the book "Teedląy t'iin_naholndak niign: Stories of the Tetlin people By Cora H. David" [90]. All the recordings used for this research were obtained by Dr. Olga Lovick, a prominent linguist and professor at the University of Saskatchewan. The recording took place with just one speaker Mrs. Cora David at several sessions for around five years starting from 2007 till 2012. Mrs. David was regarded as an expert in the Upper Tanana language and culture as well as an outstanding storyteller who died shortly after the last session in 2013. The corpus is openly accessible under the Alaska Native Language Archive comprising a total of 30 recordings [91]. Part of the recordings of this corpus were collected as audio-only with a professional recording device at a sampling rate of 44.1 KHz with 16-bit PCM wav format. The rest were taken as video, recorded with a mini-DVR camera at 240p AVC format. Most of the recordings are just monologues except

for some conversational instructions, questions or clarifications. The stories have different topics including some history or tales of the Tetlin people eg: Fishing, hunting, specific events etc. Some stories are also taken from her own life including her childhood or her mother's early life.

### 3.5.4 Dataset Preparation

The recordings are annotated using ELAN, a professional annotation tool [92]. The current state of the corpus has a single level of transcription based on just orthography. The transcription includes two levels of segmentation i.e. Utterance and Intonation Units (IU). The Intonation Units are just shorter segments of large or compound sentences. The corpus contains a total of 886 utterances and 1879 IUs. Both the segmentation levels are time aligned however, we believe the alignments are not of the gold standard as that includes a lot of silent time frames and noises. Some of the recordings also included an English translation of the utterances and some notes or comments regarding the utterance if applicable.

## 3.6 Experimental Detail

Traditional HMM based ASR systems are composed of three main components including the acoustic model, lexicon, and language model. Acoustic modeling enables representing the speech sequence distinguishing the classes of acoustic units such as phones or subwords considering the variability with respect to different speakers and environments [81]. The lexicon

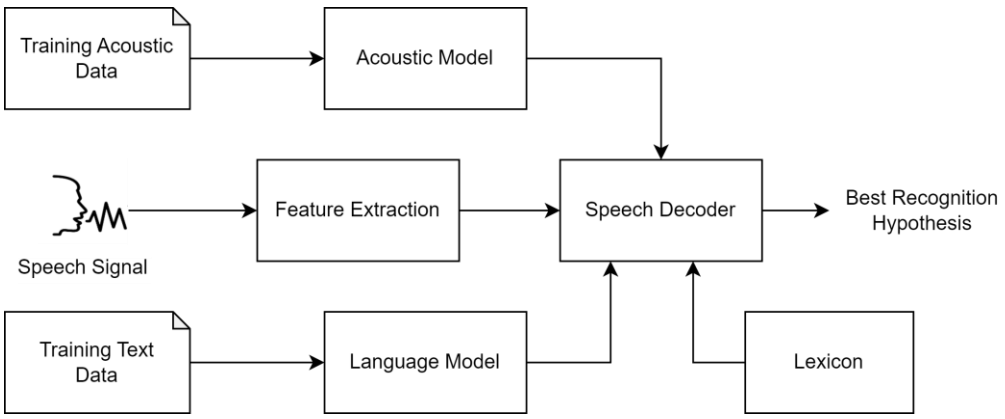


Figure 3.1 Architecture of an HMM based ASR system and its component

demonstrates the phoneme structure of each word corresponding to its spoken representation. The decoder of an ASR system builds an N-best list of hypotheses from the acoustic model in combination with the lexicon which is further refined using a statistical language model to find the best recognition hypothesis. A high level overview of the whole process is shown in Figure 3.1,

The rest of the section will describe our experimental detail including the data preprocessing, feature extraction and different modeling components stated above.

## 3.7 Data Preprocessing

As described earlier, although the dataset contains two levels of segmentation for the transcription, we only considered using the IUs for developing all our models. Generally shorter length audios are easier to train as well as promote better performance. The length of most IUs ranges from 2 to 4 seconds except for some longer IUs up to 15 seconds. The corpus includes some segments with missing transcription, silent audio and also some questions and clarifications in English. We removed all these portions resulting in a total of 1813 IUs before proceeding to the next step. It's important to note that some of the resulting IUs still include some loan words from English within the narratives which we kept as it is. Although we had only one speaker for all the recordings, we organized each recording session as a different speaker inside the Kaldi data folder. We resampled all the audio from 44 KHz to 16 KHz and converted the audio to the mono channel. We normalized the text removing some special symbols, tags, or unwanted elements. The data were split to training and testing set in a 90:10 ratio. However, for alignment evaluation, we just visually compared a small set of recording manually aligned by ourselves.

### 3.7.1 Lexicon Design

To the best of our knowledge, there is no pronunciation dictionary readily available for Upper Tanana. Developing a pronunciation lexicon manually is time consuming and requires in depth expertise of the language. Grapheme-to-phoneme (G2P) methods are quite common on the other hand and have shown satisfactory performance in past studies [93], [94]. There are two general approaches for G2P, 1) Using a rule-based conversion system which also requires some level of expertise of the language 2) Bootstrapping G2P using statistical machine translation (MT)

methods. Since for this research we focus on a non-expert modeling of ASR, we followed MT method utilizing an open sourced G2P conversion tool 'Sequitur G2P' based on joint multigram modeling [95]. To train our G2P model the initial lexicon was compiled from the study by Siri et al which includes 272 Upper Tanana words along with their IPA pronunciation [96]. However, for easier understanding, we first converted all the IPA symbols to their corresponding ARPABETs using the standard IPA to ARPA conversion chart. Although some of our recording files included a separate word list, we extracted all the words directly from the transcriptions of our corpus using a simple rule based split function. We trained the G2P model up to a $5^{th}$ order N-gram and cross validated it on a small test set from the initial lexicon. After successfully decoding all the words in the test set, we used the model to construct the final pronunciation lexicon for the whole corpus.

### 3.7.2 Feature Extraction

Traditional speech recognition systems usually rely on handcrafted features as input. Mel Frequency Cepstral Coefficient (MFCC) is the most widely used feature set by researchers for a long time. However, in a Kaldi based training environment it's possible to extract other features such as Perceptual Linear Features (PLP), Mel Filter-Banks (Fbank), Spectrogram Features and Pitch features. Besides some literature suggests that Fbank features provide better WER for DNN based acoustic models [32]. In this research, we used MFCC, PLP and Fbank feature separately as well as combining them with the pitch for training different models. For GMM based models the common practice is 13 dimensions for MFCC or 23 dimensions for PLP or Fbank features. The features are extracted for a frame size of 25 ms with a shift of 10 ms allowing overlapping. In contrast, the hybrid models such as TDNN use 100 dimensional i-vectors extracted over standard features from longer non-overlapping frame sizes usually 1500 ms [97]. The i-vectors encode speaker and channel information which is helpful for speaker adaption.

### 3.7.3 Acoustic Modeling

For acoustic modeling, the sequential property of the speech signal is modeled by HMMs where the states correspond to each phonetic unit. The output probability density of the HMMs (pdfs) can be modeled using either GMM or DNN. Here, we explored both GMM-HMM as well

as hybrid DNN-HMM models for acoustic modeling. Before feeding into the model the features are normalized using Cepstral Mean Variance Normalization (CMVN).

### 3.7.4 GMM Based Models

We followed the "Kaldi for Dummies" recipe for modeling GMM based models. First, a Mono-phone model is trained using one of the three features and their combination with the pitch described earlier to produce better forced alignment for training more complex models. For context dependent modeling triphone models are trained by grouping the left-right adjacent phonemes. These models use the first and second order derivatives (Delta and Delta-Delta) on top of their original features. In Kaldi, this model is labelled as *tri1*. For more complex models we applied Linear Discriminative Analysis (LDA) including ±4 neighboring frames and Maximum Likelihood Linear Transform (MLLT) in addition to the triphone model (*tri2*). We also applied a speaker adaptive training (SAT) using feature space Maximum Likelihood Regression (fMLLR) provided inside Kaldi (*tri3*). As models get better the training data also gets better aligned to the phonemes which facilitates further training process.

### 3.7.5 DNN Based Hybrid Models

Three Hybrid Models were implemented as part of this research. As the Hybrid Models can't be trained directly from raw speech, frame-level alignments are produced using the best GMM based model trained earlier. Our first Hybrid model is a simple DNN-HMM model based on top of the Kaldi "NNet2" recipe. The architecture uses 3 dense layers of 500 nodes with tanh nonlinearity. Essentially this is just a deep neural network with no special modeling unit. We wanted to assess how a basic DNN performs over GMM. The network is trained for 20 epochs with an incremental setup for adding hidden layers at every 3 epochs. We built our second model following the 'TDNN' recipe comes with mini librispeech example scripts in Kaldi. The architecture consists of 12 TDNNF (factored TDNN) layers with an initial batch normalization and a final linear layer for output probability. Each factored TDNN layer has a linear affine sequence of operation similar to a bottleneck transformation with an input dimension of 768 nodes and a bottleneck dimension of 96 nodes. The model is trained on high resolution 40 dimensional MFCC along with 100 dimensional i-vectors. It also utilizes speed and volume perturbation for

augmenting audio data. Our third model is based on the 'TDNN-LSTM' recipe from Kaldi LibriSpeech scripts. It uses both TDNN and LSTM to characterize long and short dependencies and filter the key information. It comprises 6 TDNN layers and 3 LSTM layers where each LSTM layer is placed after every 2 TDNN layers. Both types of layers has 512 nodes. Every TDNN layer accommodates an affine sublayer with Relu activation and a batch normalization. This model also uses a similar input structure of (40 MFCC+100 i-vectors) and the data is also volume and speed perturbed as the earlier model.

### 3.7.6 Language Modeling

Language models are used to model the inter-word relationship in a larger context of preceding or succeeding words[35]. Traditionally N-gram or RNN based language models are used to produce individual word probability. However, during the decoding process, Kaldi only supports N-gram language models. It is possible to use DNN or RNN based language models afterwards for rescoring and refining the transcriptions. Due to limited data, we only used N-gram language models in this study. SRILM toolkit has been used to produce the N-gram models included in the Kaldi script. During the training and evaluation, we only used the training transcriptions to build the language models. Although it is a common practice to utilize all available text data to build a language model. But due to the nature of the endangered language and specifically for Upper Tanana there is no additional text resources currently available. Therefore, we decided to only use the training data to get a more practical result of the ASR. We adopted a bigram and a trigram model for decoding all our models.

### 3.7.7 Results & Discussion

#### 3.7.7.1 Implementation Platform

All the experiments were performed on a platform with Intel® Core™ i7-6700 processor (3.4 GHz) and GeForce GTX 745.

#### 3.7.7.2 Feature Comparison

To compare the performance of ASR models we only used the Word Error Rate as our evaluation metric. As mentioned earlier the performance of the DNN based hybrid models depends on the alignment produced by the GMM based models. So first we evaluate the results from different features only on the GMM based models. Although the Speaker Adaptive Training generally leads to better performance. However, as our corpus includes only one speaker, depending on the type of feature sometimes speaker independent systems such as the vanilla triphone model (*tri1*) or triphone with LDA and MLLT (*tri2*) outperformed more complex models. Thus, we report only the best results for each feature here. From Table 3.1, it can be seen that the recognition capability of every feature varies depending on the n-gram. For example, PLP yields the best WER while using a 2-gram language model, in contrast, MFCC yields better WER with a 3-gram model. However, it is evident that adding the pitch consistently improves the WER for

Table 3.1: Comparison using different features with respect to different n-grams

| Model | WER (%) | |
|---|---|---|
| | 2-gram | 3-gram |
| MFCC | 52.92 | 51.97 |
| MFCC + Pitch | 52.49 | 51.44 |
| PLP | 51.95 | 52.76 |
| PLP + Pitch | 53.81 | 51.71 |
| FBANK | 57.48 | 60.89 |
| FBANK + Pitch | 55.91 | 56.43 |

every feature in GMM based models reducing an average of 1% WER. Hence, we selected the best feature (MFCC + Pitch) for producing forced alignment of the training data.

### 3.7.7.3 Forced Alignment

The acoustic model developed in this research can be used as a separate forced alignment tool for Upper Tanana. Although the current vocabulary is limited to the words from the training corpus which might not be ideal for a standard forced alignment tool. However, as this is the only dataset available at this point, it could be useful for linguistic analysis like isolating speech sounds, keyword search, phonetic analysis etc. for Upper Tanana. Figure 3.2 shows a sample textgrid file demonstrating the boundaries of the words inside an utterance along with its reference. The model output is denoted as tier "words" on the right side of the Figure and the handcrafted one is denoted

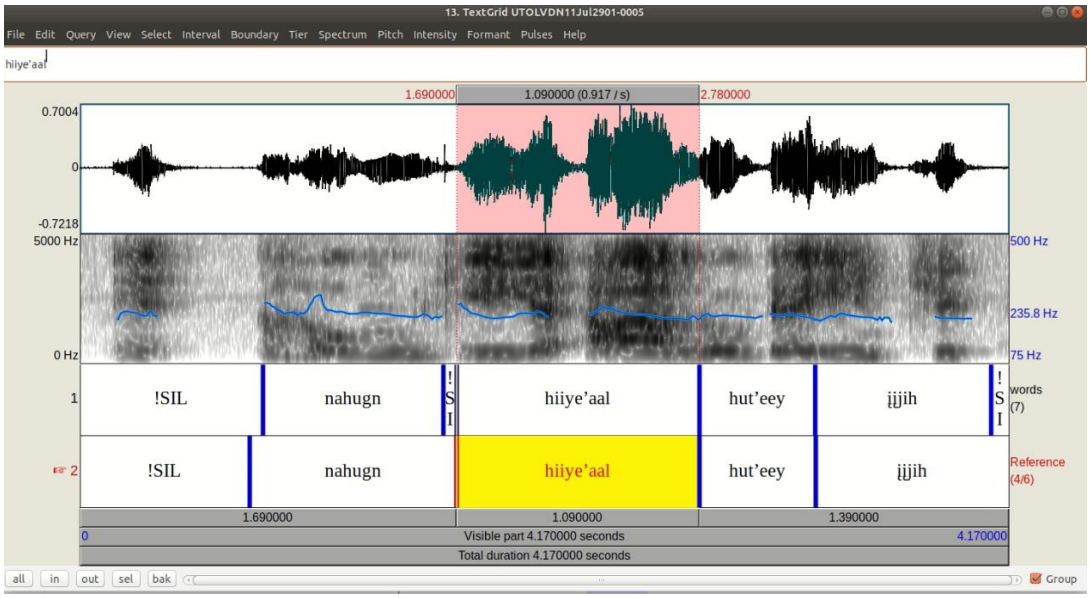


Figure 3.2 Sample word level alignment for Upper Tanana. The top tier("words") is the aligned output from our acoustic model whereas the bottom tier ("Refence") is a handcrafted alignment produced for reference

as "reference". If we see closely, the output from our acoustic model is rather somewhat better than the handcrafted alignment since the model can detect even the shorter segments of the silent part in the utterance. In this research, we are unable to quantify the results of the forced alignment due to the lack of a gold standard alignment file which is usually provided by an expert linguist.

#### 3.7.7.4 Comparison Between GMM and DNN Based Models

Although it is natural to use the same features for training the hybrid models, experiment shows that different features work better with different architectures. Table 3.2 enlists a comparison of the best WER found from different modeling techniques including both GMM and Hybrid models. We have found the best WER for basic DNN-HMM using Fbank features which

Table 3.2 : Comparison of DNN based models with best GMM-HMM model

| Model | LM | Feature | WER (%) | CER (%) |
|---|---|---|---|---|
| DNN-HMM | 3-gram | FBank | 59.06 | 34.28 |
| TDNNF | 2-gram | MFCC | 62.47 | 40.91 |
| TDNN-LSTM | 3-gram | MFCC | 51.71 | 25.25 |
| GMM-HMM | 3-gram | MFCC+Pitch | 51.44 | 37.77 |

confirms the statement of the literatures mentioned earlier. Followingly, TDNNF and TDNN-LSTM were shown to perform better with MFCC. From a general impression, it can be seen that none of the Hybrid models was able to outperform the best GMM model in terms of WER. However, in terms of the CER the performance is a bit different. From a very limited data, sequential training with TDNN-LSTM achieved a relative reduction of 33.14% in CER from the GMM-HMM model. The intuition of the two different matrices is that, while a low WER provides a higher degree of correctness for the exact reproduction of the words but the CER demonstrates the relative error within the incorrect words. As a result, the TDNN-LSTM model may recognize a slightly lower number of correct words, but the incorrect words are much closer to the original transcriptions.

Figure 3.3 shows a histogram of the WER as well as CER for the GMM-HMM and TDNN-LSTM models. Here we can see in terms of WER both have a similar distribution. But for CER, the TDNN-LSTM has a higher occurrence between 0 to 25% and is exponentially distributed over the rest of the error bins. In comparison, the distribution for GMM-HMM is more evenly spread

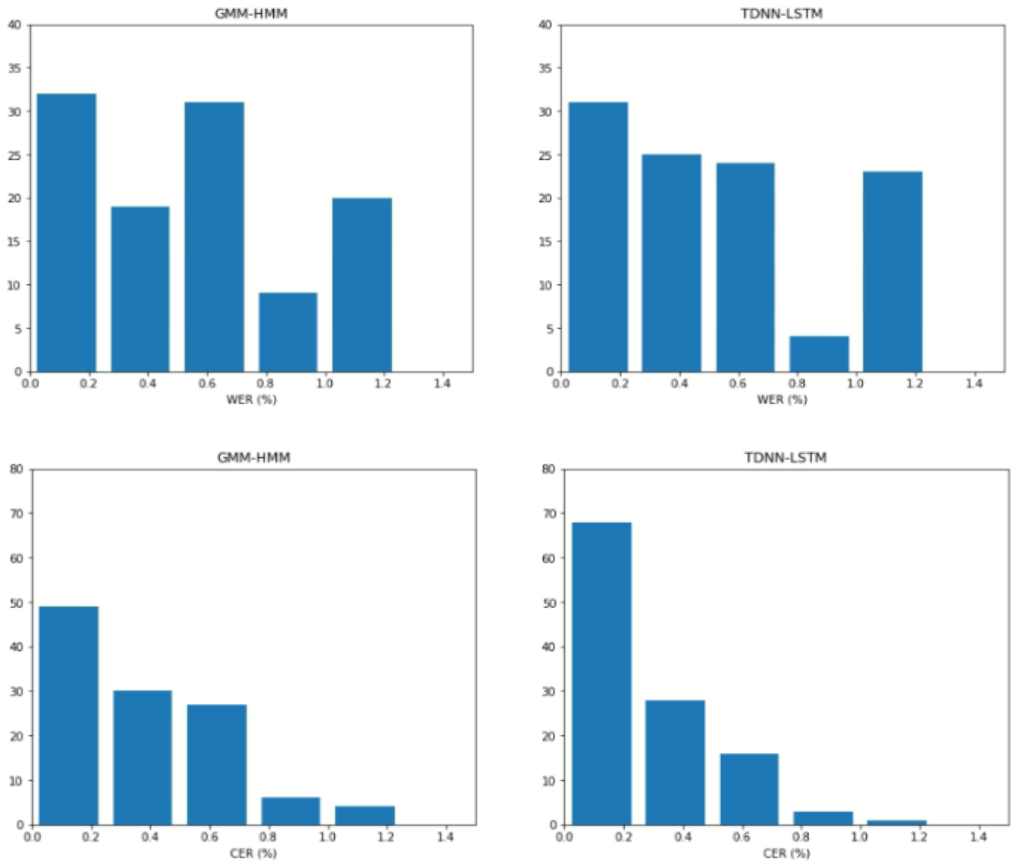


Figure 3.3 Distribution of WER and CER in GMM-HMM and TDNN-LSTM model

across the different ranges of CER. This signifies that TDNN-LSTM is more likely to have less CER for most of the utterances. Table 3.3 provides a randomly chosen set of utterances according to the different bins of WER from the GMM-HMM model. Similarly, Table 3.4 shows the WER

Table 3.3 Example of utterances according to different levels of WER and CER for GMM-HMM model

| Original | GMM-HMM | WER (%) | CER (%) |
|---|---|---|---|
| hugn t'eey łahtthagn nts'ą' įįjih | hugn t'eey nts'ą' įįjih | 20 | 30.30 |
| ay shyiit tah shyi' hiiye'įįł | ay shyiit tah shyįį | 40 | 34.48 |
| jign nįtsįį | jign dii | 50 | 58.33 |
| nǫǫgaay eł hugn niiduuy eł hugn | noogaay iin huugn niiduuy iin | 83.33 | 38.70 |
| nahnalxon ch'a | nan' naa nts'ą' | 150 | 78.57 |

Table 3.4 Example of utterances according to different levels of WER and WER for TDNN-LSTM model

| Original | TDNN-LSTM | WER (%) | CER (%) |
|---|---|---|---|
| hugn t'eey łahtthagn nts'ą' įįjih | hugn t'eey or nts'ą' įįjih | 20 | 27.27 |
| ay shyiit tah shyi' hiiye'įįł | ay shyiit tah jį' hiiyehnih | 40 | 27.58 |
| jign nįtsįį | chinh dii | 100 | 91.16 |
| nǫǫgaay eł hugn niiduuy eł hugn | noogaay iin hugn niiduuy iin | 66.66 | 35.48 |
| nahnalxon ch'a | nahatdal xol' ch'a | 100 | 42.857 |

and CER for the same utterances for the TDNN-LSTM. The first two samples confirm that for a same WER the hybrid model is closer to the true transcription. The 3rd and 4th examples indicate a hypothesis that the GMM based model is better suited for short utterances and the TDNN-LSTM is better for longer utterances. Although the WER and CER both reflect the performance of an ASR system, generally the aim of an ASR is to exactly reproduce the transcriptions. So, the WER is generally more preferred as an evaluation metric.

To better analyze the hypothesis, we compare different acoustic models for different length of utterances. Figure 3.4 demonstrates the result based on quartiles of character length of the

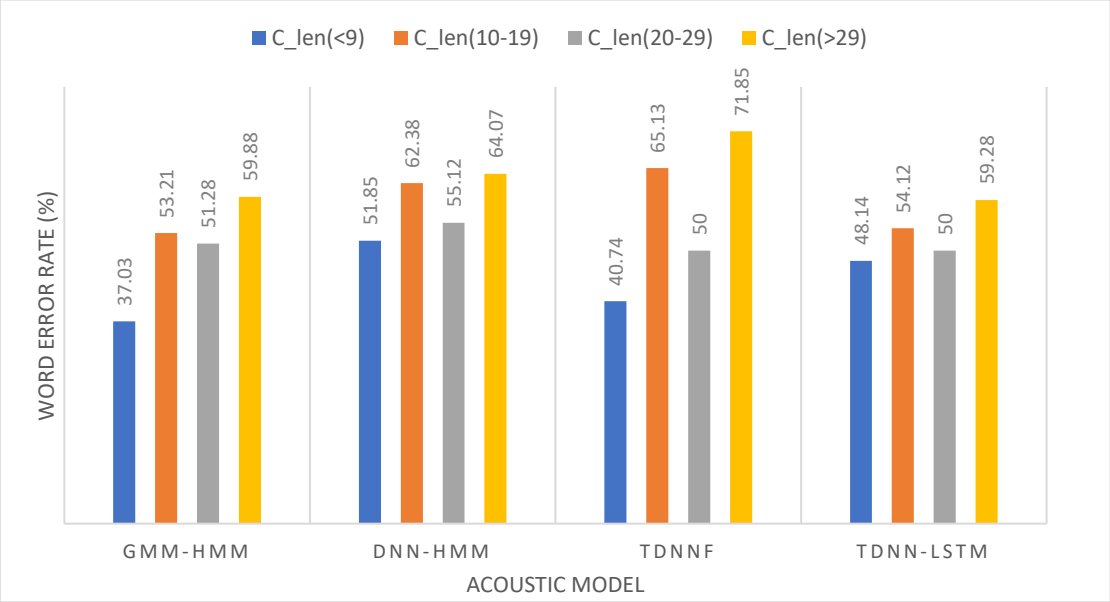


Figure 3.4 Comparison of different models based on character length of an utterance. C_len stands for character length

utterances. Because even a single word with a longer character length may pose complexity for speech recognition. As it can be seen for a shorter utterance length the standard GMM-HMM models perform much better than any of the hybrid models which include both isolated word recognition and a combination of multiple shorter length words. On the other hand, as we grow the length of the utterance TDNN-LSTM model tends to have less WER than the best GMM based model. The TDNNF however model has a very close WER to GMM-HMM for shorter length utterances. But for other quartiles, the TDNNF model shows a bit of uncertainty in its performance. To have a better understanding a random example from each quartile is shown in Table 3.5.

Table 3.5 Example of Transcriptions for each quartile of utterance length

| Transcription | C_len(<9) | C_len(10-19) | C_len(20-29) | C_len(>30) |
|---|---|---|---|---|
| Original | *neetsay* | *dihįįshyįį'* | *mbeł ch'ikee julkąą xa hii'eł* | *nts'aa' tsuudį' hiinįįthan niign t'eey* |
| GMM-HMM | *neetsay* | *dii iin shyįį'* | *meł gąą xa hiiyehnih* | *nts'aa' ts'uunih dii niitanh niign t'eey* |
| DNN-HMM | *neetsay yih* | *diign shyįį'* | *meł ts'iikeey ts'uutaanak hiiyehnih* | *nts'aa' tuu niihąą niign t'eey* |
| TDNNF | *neetsay* | *dishyįį'* | *eł ch'itay julkąą xa hii'eł* | *nts'aa' dinįį' haniig eł* |
| TDNN-LSTM | *neetsay* | *diign shyiit* | *meł ts'iikeey ch'il gąą xa hiiyehnih* | *meł ts'iikeey ch'il gąą xa hiiyehnih* |

To analyze even further we examine the WER for the number of words in utterances regardless of the character length. The results are shown in Table 3.6. Here an interesting observation is that TDNNF has a WER of 61.11% and strongly outperforms other models for single or isolated word recognition. Whereas the TDNN-LSTM shows 100% WER for the same. This also stands the same for longer words. An example of a long isolated word from the test set is given below,

Original ; *naach'ihnaak'ąą'*
GMM-HMM ; *naachihnaakąą*
DNN-HMM ; *naachihnaakąą*
TDNNF ; *naach'ihnaak'ąą'*
TDNN-LSTM ; *naachihnaakąą*

Table 3.6 Comparison of different models based on Number of Words in an utterance. N_Words stands for Number of Words

| Model | WER (%) | | | |
|---|---|---|---|---|
| | Single Word | N_Words(2-3) | N_Words(4-5) | N_Words(>5) |
| GMM-HMM | 72.22 | 51.36 | 48.24 | 63.10 |
| DNN-HMM | 88.88 | 58.22 | 58.77 | 62.13 |
| TDNNF | 61.11 | 56.16 | 65.78 | 70.87 |
| TDNN-LSTM | 100 | 49.31 | 56.14 | 54.34 |

Here although the outputs from different models are very close but only the TDNNF seems to accurately map the apostrophes (‘) within the word. As of our previous character length based evaluation, for a higher number of words TDNN-LSTM again has a better WER than others. Due to the limitation of data availability, our sample size for test data was also kept small. Therefore, from a non-expert linguistic perspective, it is difficult to determine a true qualitative comparison between the two modeling schemes. But analyzing the different results it can be intuitively said TDNN-LSTM can better capture the longer sequence of words and also the output is often closely produced to the original transcription. For isolated word recognition, TDNNF has a better modeling capability even for longer length words, but overall, the GMM-HMM has better word accuracy over any other models.

It is also important to note that, using a GPU machine the TDNN-LSTM takes around 52 min 19 sec for training while the GMM based model takes only 1 min 30 sec on average. Similarly, the average decoding time for a TDNN-LSTM model is 59.32 ms higher than a GMM based model. This summarizes that even though deep learning techniques have shown to be very effective for high resource speech recognition, the potential of such technology for endangered languages with limited data is still under discussion.

## 3.8 Conclusion

This work presents a comparative study on feature selection and modeling techniques towards building an efficient ASR pipeline for the endangered Upper Tanana language. We

demonstrate that MFCC and PLP both can be useful for modeling traditional GMM based acoustic models. However, additional pitch features can certainly boost up the performance of tonal languages. We analyze three DNN based Hybrid models with respect to selective features. Experiments show that TDNN-LSTM can potentially improve the transcription task in a certain aspect, but it still needs further examination by an expert linguist for a true evaluation. Although a closely correct transcription will provide a good baseline for linguists, sometimes it can also take more time to correct it if a lot of words are incorrect. Therefore, it is a small tradeoff between the GMM-HMM and Hybrid TDNN-LSTM, but in general, both models can be similarly useful for endangered speech recognition.

The bottleneck in language documentation specially for endangered languages is a big challenge that requires early attention. Although our best result with 51.44 % WER is definitely not comparable with a human-level transcription, but it is the first step towards building a more accurate ASR system and also a contribution to further research. For future work, our next step is to study multilingual acoustic models to build a more robust ASR system for the target language. Since there is no pre-existing lexicon for a lot of endangered languages, we also plan to explore End to End systems with state-of-the-art technologies that can skip the lexicon requirement and improve performance.

# 4 Leveraging Cross-Lingual Transfer Learning and Data Augmentation for Endangered Speech Recognition: A Study on Upper Tanana

This chapter includes a manuscript entitled “Leveraging Cross-Lingual Transfer Learning and Data Augmentation for Endangered Speech Recognition: A Study on Upper Tanana” by Zarif Al Sadeque and Francis M. Bui, which is under submission to the IEEE Access journal. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). The background concepts of this chapter including SSRL, transformers, CTC, etc., which are explained in Chapter 2.

## 4.1 Abstract

Unlike high-resource languages, endangered languages often lack any pronunciation or language model available for directly training Automatic Speech Recognition (ASR) systems. End-to-end (E2E) ASR techniques have been studied to overcome such obstacles, wherein acoustic features are mapped directly to the graphemes or characters. Although E2E models can achieve good results, these systems usually require larger amount of speech-text paired data than traditional ASR systems to provide similar performance. In recent years, self-supervised pre-training has achieved great success leveraging unlabeled speech data from multiple languages to build an acoustic representation that can be fine-tuned on a downstream task, e.g., ASR with limited language resources. This strategy has also found utility on applications with domain shift issues, particularly for major languages such as English or Spanish. However, for an extremely low-resource scenario, where the target language is completely unobserved in the pre-training stage (i.e., out of language), the performance leaves much to be desired. In this work, we investigate such a scenario, focusing on the critically endangered Athabascan language Upper Tanana, and propose an effective method based on cross-lingual transfer learning from pretrained transformer model and data augmentation. Using only 1 hour and 9 minutes of available speech data and no language model, the proposed model exhibits a relative reduction of 12.8% in Word Error Rate (WER) compared to the best known state-of-the-art conventional ASR system, and 7.3% relative

reduction over using only transfer learning. We also evaluate the impact of data size and complexity in contrast to previous state-of-the-art systems and validate the proposed augmentation strategy with several other E2E techniques. Results demonstrate that the proposed model is more resilient to the impact of data size, thus ensuring robustness, low computational complexity and better performance. Moreover, the augmentation strategy is consistent for different E2E methods.

## 4.2 Introduction

Speech is considered the most direct form of communication that carries accurate information and emotion between humans. Researchers have endeavored to build intelligent systems that can interact through speech for many years. In this context, Automatic Speech Recognition (ASR) is an important stepping-stone toward successful human-machine interaction. Accordingly, collaboration from multiple disciplines and advancements in artificial intelligence have resulted in enormous success for most major spoken languages. However, minority languages from the indigenous and tribal communities such as Upper Tanana still lack resources and attention from researchers. Many of these languages are on the verge of extinction and are commonly regarded as endangered languages. Approximately 50% of languages of about 7000 languages are now on this endangered list and may not persist for another century [1]. Consequently, without suitable intervention or remedy, this undesirable trajectory may induce a tremendous loss in cultural heritage and diversity. One of the major approaches to preserving these languages is by documenting audio and textual evidence, which typically requires significant financial resources, workforce, and subject matter expertise. Moreover, these projects are usually time-consuming and may not be feasible for nearly extinct languages. In this context, ASR technology can assist people with minimal linguistic knowledge and resources to support this process [74].

There are two general approaches for modeling ASR systems currently in practice: one is the traditional approach that uses HMM-based methods [77], and the other involves E2E systems based on deep learning [98]. The traditional ASR methods can be decomposed into acoustic, pronunciation and language models, which require a handcrafted lexicon designed by linguists, external language modeling and expert knowledge of the specific language. E2E techniques, on the other hand, can potentially avoid the complex modeling procedure and also require less human

intervention [63]. However, these models need to have significantly more labeled data to match the performance of the traditional models, which is a big challenge for endangered languages [28].

Recent evolution in partially supervised or self-supervised pretraining has established promising potential of E2E speech recognition systems with limited data [27], [99], [100]. Supervised pretraining is essentially another form of transfer learning that focuses on a single predefined task, but it potentially converges much faster than a standard E2E technique [27]. Self-supervised pretraining instead leverages large-scale unlabeled data to effectively train more generalized and robust representations of the data, which can be utilized to fit multiple downstream applications [60]. This type of learning framework based on unsupervised pretraining is often referred to as self-supervised learning (SSL) [101] or self-supervised representation learning (SSRL) [102].

While SSL requires more data to achieve a good representation, it can be fine-tuned later with much less labeled data to fit the targeted task. Moreover, it also improves the performance on cross-lingual speech recognition, where multiple languages are utilized for the pretraining session [103], [104]. By training in this manner, the model is pushed to learn a language-invariant robust speech representation that can be exploited to reduce the search parameters for low resource languages and give the training process a head start. It is important to know that the self-supervised pretraining approach notoriously demands a very high computational cost [51]. However, the idea is that, once the pretraining is completed, the model can be fine-tuned easily with lower data requirements as well as less computational resources.

Due to data shortage, endangered languages have been generally trained under traditional ASR systems. However, for some critically endangered languages like Upper Tanana, neither a large amount of data nor an already established pronunciation dictionary is available. This makes the speech recognition task extremely difficult. For traditional ASR, one way to solve this task by using closely related languages to build a pronunciation model from graphemes of the available data [29]. This still requires an extensive knowledge of the particular language family, and the availability of closely related languages.

Another way is to use a generalized representation from cross-lingual acoustic model and fine-tune it to improve the model. To this end, most of the existing research works using SSL include a small ratio of data from the low-resource languages in the pretraining session and analyze

the performance. Some works also evaluate the performance based on data from different domains of the same language for fine-tuning and examine adaptability [28]. This research work, by contrast, tackles the problem by applying no additional pretraining, but rather by fine-tuning using a completely unseen language from the pretrained session, and subsequently improving it even further using our strategically designed data augmentation method.

The rest of the Chapter is organized as follows. Section 4.3 clarifies the contributions of the Chapter. Then, related works are described in Section 4.4. This is followed by the proposed methodology in Section 4.5 and the experimental setup in Section 4.6. Next, experimental results and discussions are reported in Section 4.7. Lastly, the conclusion is presented in Section 4.8.

## 4.3 Contributions

In this study, we focus on developing an ASR system for Upper Tanana and investigating the performance of cross-lingual SSL compared with different existing acoustic modeling techniques. Upper Tanana is a critically endangered Athabascan language. It belongs to the Alaskan subgroup of the Northern Dene language family [105]. According to the UNESCO Atlas of the World's Languages in Danger [80], this language has fewer than 50 speakers and most speakers are now grandparents. As the number of speakers for this language is constantly decreasing, preservation has become an urgent necessity. To the best of our knowledge, there is no published research so far regarding ASR for this language. This language is designated as an extremely low-resource language, as the only dataset available to date has merely 1 hour and 9 minutes of transcribed speech data. There are four main contributions in this study, summarized as follows:

- We develop a complete E2E ASR modeling strategy for extremely low-resource endangered languages without any pronunciation model or external language model, which delivers better performance compared to other traditional and E2E models.
- We show that self-supervised learning can be fine-tuned effectively for out of language speech recognition even on a low scale dataset. It will be demonstrated that, even without applying any data augmentation, this approach can still outperform state-of-the-art traditional models.

- We provide an effective data augmentation technique that can be implemented with any E2E ASR model to potentially improve the performance. Our proposed system attains a relative reduction of 7.3%–21% WER over state-of-the-art E2E models like wav2vec2-voxpopuli, XLS-R and Deep Speech.
- We analyze the effect of data size and complexity compared with other traditional and E2E ASR models, confirming that the performance is consistent over data size with less complexity.

Furthermore, this work should serve as a reference for future research activities on ASR based on this endangered language, and more broadly for designing acoustic modeling schemes for other extremely low-resource scenarios.

## 4.4 Related Works

Various research investigations have been conducted to date in order to compensate for the limited amount of labeled data for the ASR development in endangered languages. There have been several studies on Seneca, an endangered Iroquoian language. Jimerson *et al.* [1], [74] proposed using additional data from online resources along with synthetic verb forms to improve traditional ASR modeling with limited resources. Thai *et al.* [78] further studied E2E modeling frameworks for the same language using Deep Speech and mini-GCNN model. They proposed a multistage data augmentation method to improve the WER up to 15% over using transfer learning.

Transformer based methods also garnered a lot of attention very recently for endangered speech recognition. Qin *et al.* [106] introduced a multilingual and multilevel unit modeling technique for the low-resource Lhasa dialect of Tibetan language. They pretrained a transformer-based model with Bengali, Nepali and Sinhalese along with Tibetan and fine-tuned the model on the Lhasa dialect of Tibetan language to improve the character error rate. Another study by Shi *et al*. [68] used transformer based models for Yoloxochitl Mixtec, a Mexican endangered language. The paper compared rule-based transcription correction and fusion of multiple transformer-based architectures for supplementing the data scarcity problem, showing both delivering competitively similar performances.

Self-supervised E2E models have also been well-utilized for endangered languages. Notably, Yi *et al*. [27] investigated the performance of pretrained wav2vec2 models on six low resource languages from the *Call home* dataset, and achieved an improvement around 20% over supervised pretraining.

There are also some papers that share similar objectives and constraints, thus providing inspiration for our work. Gomez *et al.* [30] studied self-supervised methods, including pretrained wav2vec2 models, to evaluate performance on domain shifted ASR. However, they did not apply the models on endangered speech recognition. Focusing on a significantly diverted and coded speech, they investigated the impact of fine-tuning the data size and robustness of the models. A similar study was conducted by Al-Ghezi *et al*. [28], where they applied different pretrained wav2vec2 models on L2 Swedish language (Swedish as second language learner). They showed that large multilingual pretrained models achieved better WER over monolingual Swedish models. Another study by Wang *et al.* [107] introduced the term "out of language", and evaluated the performance of 5 common voice languages without including them in the pretraining section of the SSL. Their results also advocated cross-lingual pretraining over monolingual baseline models for better stable performance. However, the paper did not investigate the effect of data size used for fine-tuning.

## 4.5 Methodology

This section describes our proposed model based on cross-lingual transfer learning from a pretrained transformer model and data augmentation (DA-XLSR). First, we briefly describe the problem formulation. Then, we provide a comprehensive overview of the overall structure of the proposed model, including a high-level summary of the wav2vec 2.0 framework, its pre-training procedure and adaptation to our endangered speech recognition system. Last, we explain the data augmentation process and further fine-tuning to refine the model.

### 4.5.1 Problem Formulation

Four major issues impel this research study. First, most of the previous research works on low-resource languages experimented with at least 10 hours of training data [68], [79], [81], [108].

Pushing the boundary on the data scarcity, this research endeavors to improve the ASR performance on an extremely low-resource scenario, where only 1 hour and 9 minutes of curated speech data are available.

Second, endangered languages often lack sufficient pronunciation and orthographic documentation, such as a phoneme dictionary, which is a primary requirement for most traditional ASR systems. Therefore, this study considers developing an ASR system without any pronunciation or language model. We have to make do with only the transcriptions available from the extremely limited training data.

Third, it is noted that the majority of the previous studies based on pretrained Wav2Vec2 included the corresponding language in the pretraining phase [27], [28], [60], [103], [104]. As a result, the existing evaluation criteria may no longer be appropriate for a scenario involving previously unseen language. Accordingly, this motivates us to also investigate the domain adaptation problem and improve the performance even further. Besides, as the language is not included in the pretraining, the added challenge is that the model often does not have any prior knowledge of certain phonemes or characters and their associated pronunciation.

Fourth, ASR can boost the preservation process of any endangered language. Therefore, it is reasonable to deploy such a technology in end-user applications, with field data collection by the linguist. Thus, the computational complexity and resource requirements need to be practical. To address these issues, we provide a thorough analysis on training and testing time on the proposed model.

### 4.5.2 Wav2Vec 2.0 framework

Our proposed model is built on the transformer model Wav2Vec 2.0 originally developed by Baeveski *et al*. [60]. The architecture of Wav2Vec 2.0 framework is comprised of three major components. As shown in Figure 4.1, the first one is the feature encoder $f : \mathcal{X} \rightarrow \mathcal{Z}$. It is based on a multilayer convolutional network that allows raw audio $\mathcal{X}$ as input and maps to the latent

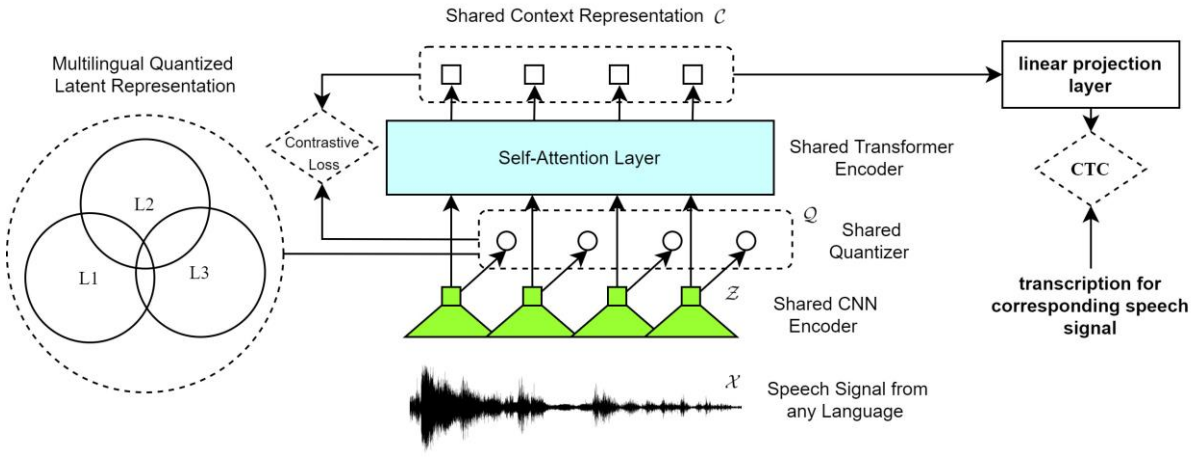


Figure 4.1 On the left side: the structure of cross-lingual wav2vec2.0 containing a shared quantizer on top of a shared CNN encoder, producing cross-lingual quantized speech embeddings for self-supervised pretraining through contrastive loss. On the right side: the decoding module consisting of an additional linear projection layer trained by CTC loss criterion.

output $z_1$ , $z_2$ , . . . $z_T$ for $T$ timesteps. The encoder uses temporal convolution and layer normalization with GELU activation.

The second component is the transformer network $g : \mathcal{Z} \rightarrow \mathcal{C}$ that follows a similar architecture like BERT [56], [65]. Despite using a fixed positional encoding, the network utilizes a convolutional layer in its beginning, which effectively works as a relative positional encoder. The self-attention layers of the transformer produce a contextualized representation $c_1$, $c_2$, ..., $c_T$ from the latent information.

The fundamental advantage of wav2vec 2.0 model comes from its self-supervised pretraining, where the pseudo labels are generated from its quantization module $\mathcal{Z} \rightarrow \mathcal{Q}$ which is the third component of the architecture. As the output from the feature encoder $z_1$, $z_2$, . . . $z_T$ are continuous, this module discretizes them to $q_1$, $q_2$, ..., $q_T$ for a finite set of labels, thus feeding through the transformer for self-supervised training.

#### 4.5.2.1 Pretraining:

Here, the idea of self-supervised pretraining is implemented by masking a proportion of the output timesteps from the feature encoder network and train it with the unmasked quantized

outputs as target. The network learns the latent representations from the speech audio through a contrastive learning process optimizing the loss function given by,

$$\mathcal{L} = \mathcal{L}_m + \alpha \mathcal{L}_d \tag{4.1}$$

where, $\mathcal{L}_m$ is the contrastive loss, $\mathcal{L}_d$ is the diversity loss and $\alpha$ is a tunable hyper-parameter [10]. Unlike autoregressive training, the objective of $\mathcal{L}_m$ is to characterize the true quantized latent representation $q_t$ within a set of $K + 1$ candidates $\tilde{q} \in Q_t$ including $K$ negative candidates, similar to a classification problem. The negative candidates are sampled uniformly from other timesteps of the same utterance in case of a monolingual training or utterance from a different language for multilingual. The contrastive loss is given as,

$$\mathcal{L}_m = -\log \frac{\exp(sim\,(c_t, q_t))}{\sum_{\tilde{q}-q_t} \exp(sim\,(c_t, \tilde{q}\,))} \tag{4.2}$$

where $sim$ defines the cosine similarity of context representation $c$ and quantized representation $q$.

The diversity loss motivates the model to utilize both $V$ positive and negative samples that are stored in a codebook. For all $G$ codebooks, the model maximizes the entropy $H$ of averaged softmax distribution for each codebook $\overline{\rho}_g$ over a batch of utterances. Given these parameters, the diversity loss can be formulated to minimize the negentropy as

$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} -H\left(\overline{\rho}_g\right). \tag{4.3}$$

During the multilingual pretraining phase, the batches are built by sampling from a multinomial distribution

$$(\rho_l)_{l=1,\ldots,L} \text{ where } \rho_l \sim \left(\frac{n_l}{N}\right)^{\beta} \tag{4.4}$$

where, $n_l$ is the pretraining data length for language $l$, $N$ is total data length, and $\beta$ is a parameter allowing for the control of the importance between high and low resource languages [103].

#### 4.5.2.2 Fine tuning:

After the pretraining phase, the wav2vec2 model is fine-tuned by appending a randomly initialized linear projection layer to the context network, as illustrated in Figure 4.1. The pretrained model is generally trained to optimize a Connectionist Temporal Classification (CTC) loss criterion, in order to directly predict the target tokens. In our case, we perform this fine tuning in a multistage transfer learning process.

### 4.5.3 Proposed ASR model

The high-level architecture of our proposed DA-XLSR model is depicted in Figure. 4.2. The proposed model is implemented in a two-stage transfer learning scheme along with data augmentation. In the first stage, we utilize a transformer model pretrained on a high resource

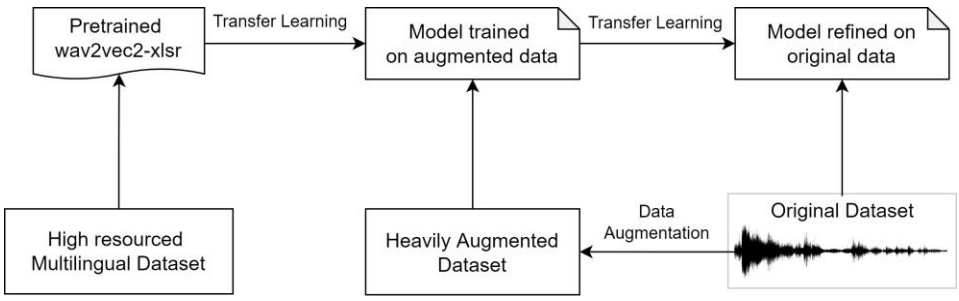


Figure 4.2 High level block diagram of the proposed DA-XLSR-53 model

dataset to initialize our model. As training a self-supervised model from scratch without sufficient data to obtain good performance is rarely plausible, most works in the existing literature for low-resource languages focused on fine-tuning already pretrained models to carry out certain downstream tasks [100]. In this research, we experiment with three pretrained models, trained in a cross-lingual setting, which are mainly extended from the wav2vec 2.0 framework. Ideally, the pretrained model can be trained on any monolingual or cross-lingual setting. However, as this

study needs to tackle a downstream domain shift problem, it is evident that the donor languages used for pretraining plays a vital role on the final performance. Depending on structural and pronunciation similarity between the donor and target languages, the performance can deteriorate even further. The main idea of cross-lingual pretraining is to learn the common knowledge among the shared data from different languages. Therefore, instead of using a monolingual pretrained model, we intentionally focus on transferring weights from a cross-lingual wav2vec 2.0 model. The dataset used for the pretraining depends on the pretrained model used. We fine-tune this model on the extended dataset of Upper Tanana generated from data augmentation. We provide a brief detail of the pretrained models later in this section.

In the second stage, we transfer the weights from the initial model, and refine on the original limited dataset of Upper Tanana. Augmented data often incorporate significant digital artifacts. Since the amount of augmented data is substantially larger than the original data in the extended dataset, the network could be skewed towards the representation of augmented data rather than original data [78]. This second stage helps suppress the skewness and improve the performance for the original dataset.

#### 4.5.3.1 Data Augmentation:

In solving the low-resource ASR problem, data augmentation approaches have been found to be quite effective in improving performance. Many augmentation methods have been proposed, including speed perturbation [109], pitch adjust [110], vocal tract length perturbation, and SpecAugment [111]. As our system is based on wav2vec 2.0 framework, which uses raw audio for input, we need to utilize augmentation techniques compatible with raw audio samples.

In this context, perturbation-based methods are mainly applied to generate new samples to extend the original dataset. Figure 4.3 shows an overview structure of the proposed augmentation process. Here, we combine six audio resampling techniques, including adding Gaussian noise (GN), frequency masking, time masking, pitch shift, clipping distortion and time stretching. Each technique is strategically selected, based on insights from the existing literature, as well as empirical performance results with our datasets. In the following, we provide a brief rationale for each technique.

Adding Gaussian Noise to audio samples smoothens the input space, and may potentially make it simpler to learn [112]. Given an input signal $x(t)$, with the Gaussian normally distributed noise $n(t)$ and a desired noise variance of $\sigma^2$, the output signal $y(t)$ can be generated as

$$y(t) = x(t) + \sigma \times n(t). \tag{4.5}$$

Frequency masking and time masking have also been shown to be effective techniques in improving ASR performance [111]. For the number of frequency channels $v$ in the audio signal, we mask $[f0, f0 + f]$ channels, where $f$ is selected from a uniform distribution and $f0$ from $(0, v - f)$. Time masking works similarly, with timesteps $[t0, t0 + t]$ masked, where $t$ is selected from a uniform distribution and $t0$ is taken from $[0, \tau - t]$. Here, $\tau$ is the number of total timesteps in the signal.

Time stretch and pitch shift work on opposing principles. Time stretch modifies the tempo without affecting the pitch, whereas pitch shift modifies the pitch of the signal without affecting the tempo. However, having both types of augmented samples may help generalize the representation during model training.

Clipping distortion is another popular augmentation technique for ASR, and it operates by clipping each audio signal in a random percentage of points. This percentage is chosen from a uniform distribution of two parameters, i.e., the min-percentile and the max-percentile.

Overall, Figure 4.3 shows visually how a sample signal may change in the time and frequency domains, after each augmentation method has been applied. We apply six methods separately and produce an extended dataset that is expanded effectively seven times, including the original audio samples.

### 4.5.4 Pretrained models

The following pretrained models used in this research are open source and publicly available in the Hugging Face online library.

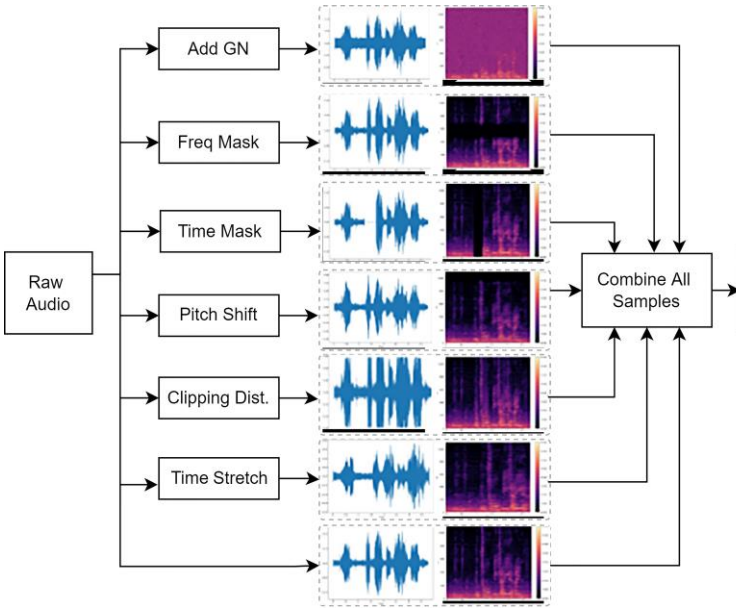


Figure 4.3 Overview of the data augmentation process

#### 4.5.4.1 Wav2vec2-xlsr-53:

XLSR-53 follows the same architecture as the original wav2vec 2.0 model. The quantization module is shared across languages which produces multilingual quantized unites for contrastive learning. The model assimilates shared discrete tokens from different languages and creates a bridge between them. It was pretrained using three large multilingual dataset including Common Voice, Babel and Multilingual LibriSpeech (MLS). The training corpus combines 53 languages with 56000 hours of speech data. This model outperformed previous monolingual models with 16% relative improvement in word error rate. It also shows its effectiveness for speech recognition on previously unseen languages, which makes it a good match for our research.

#### 4.5.4.2 Wav2vec2-large-100k-voxpopuli:

This model was trained on 100k hours of speech data on 23 European Languages. The dataset was collected from European parliamentary event recordings. Therefore, it contains European Union (EU) languages only [107]. The model uses the same hyperparameter settings as the original wav2vec2 model. It was evaluated for out of domain pre-training setup, where it uses

common voice ASR corpus in addition to the political domain EU parliament recordings. The model outperforms previous multilingual models like the XLSR-53 for most of the EU languages. It was also trained for out-of-language scenarios, such as our research problem, where the model is assessed for previously unseen languages, and the results are competitive to the XLSR-53 model. Although this model was mostly tested on EU based languages, it is nevertheless considered as one of the initial baselines for our study.

#### 4.5.4.3 Wav2vec2-XLS-R-300m:

XLS-R is a large-scale cross-lingual speech representation model, also based on the wav2vec 2.0. Originally the model was trained in three different parameter settings, including 300 million, 1 billion and 2 billon parameters. The parameter difference comes from its number of convolutional blocks for feature encoding. The model was pretrained on 436k hours of publicly available data from VoxPopuli (VP-400), MLS, Common Voice, VoxLingua107 and Babel. The whole dataset comprises 128 languages from all over the world and involves several domains. XLS-R is evaluated on a wide-range and diverse downstream tasks, i.e., Automatic Speech Translation (AST), ASR and Speech classification (Language Identification and Speaker Identification).

### 4.5.5 Baseline Mainstream Systems

We select two well established traditional ASR models including GMM-HMM and TDNN-LSTM as a baseline for this study. We also compare with some state-of-the-art end-to-end models, such as Deep Speech 2.0 and different pretrained multilingual wav2vec 2.0 models, for assessing the model performance. These models previously showed promising results on various low-resource ASR problems.

**Traditional ASR models:** We implement all the traditional ASR models using the Kaldi toolkit [77], with the SRILM language modeling toolkit for the language modeling [113]. Although our proposed DA-XLSR model itself does not require any pronunciation dictionary, for the implementation of traditional ASR models, we need to build a pronunciation dictionary as part of this research. The construction of this customized dictionary is briefly described next. It should be noted that due to low-resource nature of this language, this simplified construction should not be

considered a comprehensive or robust approach. It also shows the difficulty encountered by using traditional approaches, thus motivating the necessity for alternative E2E methods.

First, the initial dictionary is built from a list of 270 words, with their corresponding IPAs collected from the paper "Vowels of Upper Tanana Athabascan" [96]. The IPAs are then converted to ARPAbet using the standard IPA-ARPAbet conversion table [114] for easier understanding and training in the Kaldi platform. We develop the pronunciation dictionary utilizing the Sequitur G2P toolkit [95] that uses a joint sequence modeling technique to convert grapheme to phoneme. We train the Sequitur model by feeding in the initial phoneme dictionary and build a unigram model. The model is retrained up to 5-gram and the final model is used to convert all the unique words from our dataset to develop a pronunciation dictionary for Upper Tanana.

#### 4.5.5.1 GMM-HMM:

The baseline GMM-HMM model is adapted from the "Kaldi for Dummies" tutorial recipe. First, a context-independent mono-phone model is trained using 13-dimensional MFCCs normalized using Cepstral Mean and Variance Normalization (CMVN). The features are extracted over 25-ms frames shifted at every 10 ms. This produces a rough estimation for the force-alignment of the training dataset. We also experiment with using other features including Perceptual Linear Prediction (PLP) and Mel Filter-Bank (F-Bank). However, for this research, it is empirically determined that using conventional MFCC shows the best results.

The model is extended to a context-dependent tri-phone model, trained using 39-dimensional stacked MFCCs generated from the original MFCC features and their first and second order derivatives. The second model produces better alignments for the training data. Some more complex models are produced subsequently, by applying Linear Discriminant Analysis (LDA) with Maximum Likelihood Linear Transform (MLLT) and Speaker Adaptive Training (SAT) to the tri-phone model. We use a Language Model (LM) of 3-gram with Witten-Bell discounting. For all the experiments using GMM-HMM framework, the vanilla triphone model shows the best results, and is accordingly used for aligning the training dataset for DNN-based models.

#### 4.5.5.2 TDNN-LSTM:

We implement the hybrid chain Kaldi model following the “librispeech” recipe. The acoustic model is based on Time-Delayed Neural Network (TDNN), combined with Long Short-Term Memory Network (LSTM). It combines the attributes of TDNNS with LSTMs to model the long and short-term dependencies and obtain essential information. The architecture consists of 6 TDNN blocks of 512 nodes interleaved with 3 LSTM blocks of 512 nodes after every two TDNN blocks. Each TDNN block contains an affine layer accompanied by ReLU and a batch normalization. The network takes an input feature of five consecutive 40-dimensional MFCCs vectors concatenated with 100-dimensional i-vectors. We apply volume perturbation and 3-way speed perturbation to augment the data before training. For language modeling, we apply different n-grams. Again, it is also found empirically that the previous 3-gram language model, with Witten-Bell discounting, performs the best for this model as well.

#### 4.5.5.3 Deep Speech 2.0:

The Deep Speech 2.0 model was originally developed for English and Mandarin Languages [64]. However, it has been recently explored for many low-resource ASR tasks and found to be effective. In this research, we use an architecture that consists of two convolutional layers and 5 bi-directional Gated Recurrent Unit (GRU) layers. Each convolution layer is accompanied by a Batch Normalization layer and a ReLU activation. We utilize the spectrogram feature extraction for the input to the network. The power of the spectrogram is normalized over all the frames of each utterance. The architecture uses CTC loss criterion as the cost function with Adam optimizer. We use character-based tokenization and greedy search for decoding the output. We also experiment with applying the proposed data augmentation module with the Deep Speech 2 architecture.

## 4.6 Experimental Setup

This work implements both traditional ASR and E2E deep learning based ASR methods for analyses and comparisons. While the former methods do not need special requirements, the latter ones require a GPU for proper implementation and timely training. This is especially relevant

for the proposed model, which incorporates a transformer model. Accordingly, for demanding training involving a GPU, a Tesla P100 with 16 GB VRAM (as made available on Kaggle) is utilized for speed. However, when it comes to testing, to enable a fair and practical complexity comparison, we revert to a more consumer-grade mid-range Nvidia GeForce RTX 3060 GPU.

### 4.6.1 Dataset

The dataset used for this research was originally collected for an expanded edition of the book “Teedląy t’iin naholndak niign: Stories of the Tetlin people By Cora H. David” [90]. To support endangered language preservation for North American Native Languages, this dataset was further curated and annotated for research by Lovick et al [91]. The curated dataset consists of 886 utterances with native transcription and comments about the transcription in English. All the utterances are spoken by one female speaker, although the audio files were recorded in different times and different recording conditions. The dataset has a total of 1941 unique words and there is no pronunciation dictionary publicly available for this language yet.

The utterances are further segmented into 1879 shorter-length sentences denoted as Intonation Units (IU) provided along with the dataset. Since the segmentations are done manually by domain experts, and shorter segments are more likely to facilitate the training process, the IUs are utilized for training both the baseline systems and the proposed model. Most of the IUs are

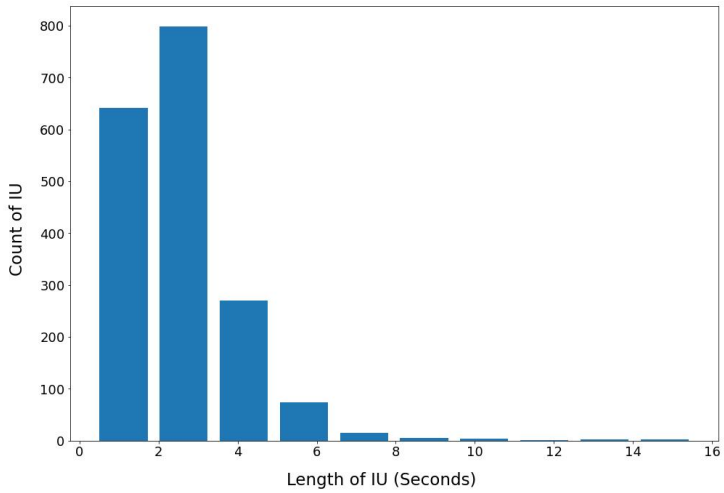


Figure 4.4 Distribution of utterance length in the dataset

within 2 to 4 seconds length, with a few long-length IUs up to 15 seconds. For easier understanding, we will refer to the IUs as utterances for rest of the manuscript. Figure 4.4 shows the corresponding distribution of utterance length in the dataset.

#### 4.6.1.1 Data Preprocessing:

In order to obtain good downstream performance results, the corpus is preprocessed to mitigate noise and redundancy. The speech data is collected in a form of interview, which includes some redundant conversational speech in English, some spoken noise, and long silent parts in the recording. The dataset also includes some utterances without corresponding transcription and vice versa. We have omitted these unpaired speech data and the redundant speech segments from the corpus before proceeding to the next step. We also normalize the text data, removing any special characters and symbols. However, the single quote mark (‘) is kept, as this character is used to notate some acoustic variations in words.

#### 4.6.1.2 Training:

Before training different models, we split the preprocessed dataset into a development set and Testing set using a 90:10 split ratio. The traditional models require the training audios to be aligned with the corresponding phonemes. Therefore, to verify alignment quality, we evaluate the forced alignment with a hand-aligned test set previously prepared by the experts before training the final model. As a result, the test sets in the traditional models and E2E models have different number of utterances. The development set is further split into a 90:10 training and validation set. The training configurations for the Deep Speech 2 and traditional models are already discussed in previous section.

For our proposed model, we use the same hyperparameter setting for all three pretrained models. We train in an incremental setting, starting with only 10 minutes of training data and increase it to 20 min, 40 min and full 1 hour 9 min of data. The only difference in hyperparameters for different data sizes is in the mask time probability, which is selected as 0.075 for 10 minutes and 20 minutes data sizes, and as 0.05 for 40 minutes and full data size. The other parameters are mostly adopted from the original wav2vec 2.0 model [60] and other variants [103], [104]. Essentially, we select the hyperparameters heuristically, and also based on the previously available

literature whenever appropriate. The common hyperparameters used for finetuning all the pretrained models are summarized in Table 4.1.

All the models are primarily trained for 70 epochs, with an initial learning rate of 3e-4. Moreover, we utilize early stopping based on the word error rate (WER) on the validation set. The models are optimized using Adam with a warm up for 500 steps, held constant for next 40% updates, and linearly decayed for the rest of the time steps as training progresses. We do not train the feature encoder layers, as those layers are supposed to be already trained sufficiently during the pretraining session.

Figure 4.5 shows a corresponding overview for the training and evaluation process. For all

Table 4.1 : Hyperparameters for the proposed model

| Hyperparameter | Value |
|---|---|
| Attention Dropout | 0.1 |
| Hidden Dropout | 0.1 |
| Feature Projection Dropout | 0.0 |
| Layer Dropout | 0.1 |
| Warm up steps | 500 |
| Learning rate | 3e-4 |

three pretrained models, we use the large sized wav2vec 2.0 setup, with approximately 300M parameters and primarily fine-tune only utilizing the cross-lingual transfer learning without any data augmentation. This initial fine tuning takes around 2 hours 10 mins for each model, which provides some observational data to find the best pretrained model for the Upper Tanana Language as well as to find the performance of just using transfer learning. Finally, all three models are retrained in a two-stage learning described earlier on the augmented dataset with a similar incremental setting corresponding to the augmented set prepared from 10 min, 20 min, 40 min and the complete record of the original data. The training on the augmented set of total data takes around 11 hours for each model.

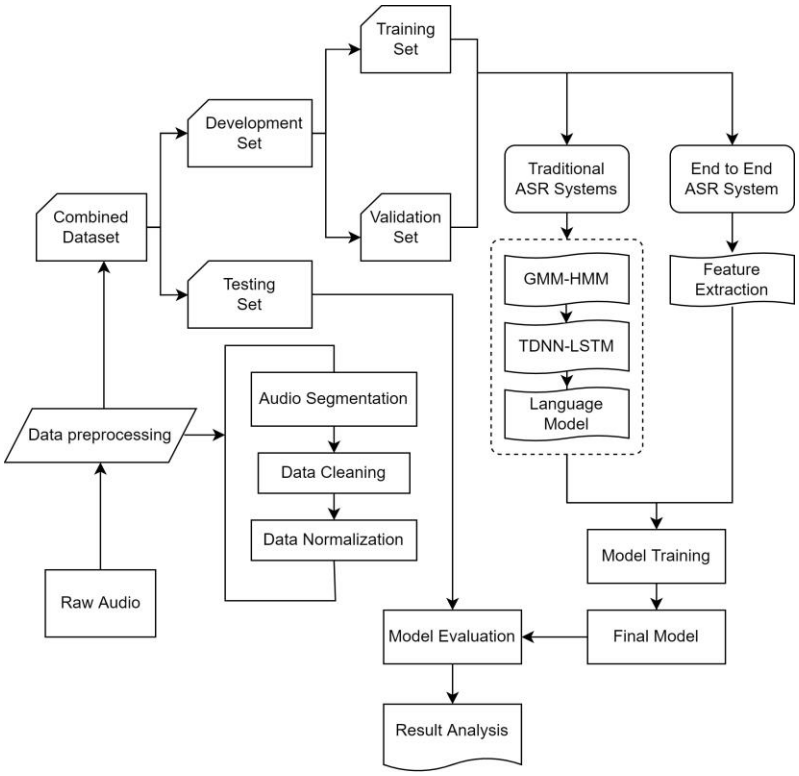


Figure 4.5 Overall training and evaluation pipeline for selected models.

## 4.7 Results And Discussions

This section presents the results and discusses various performance observations of the proposed model. We mainly use WER as a performance metric, which is the most commonly used metric in ASR model evaluations.

### 4.7.1 Performance Improvement:

First, we evaluate the model performance compared with the traditional HMM based models. From Table 4.2, it can be seen that, at 45.06%, the proposed DA-XLSR-53 model yields a WER that is 6.91% lower than the baseline GMM model, and 6.65% lower than the best hybrid model TDNN-LSTM.

Table 4.2 : Results for the proposed model compared to the traditional HMM based models

| Model | Feature | LM | WER% |
|---|---|---|---|
| GMM-HMM | MFCC | 3-gram | 51.97 |
| DNN-HMM | Fbank | 3-gram | 59.06 |
| TDNN-LSTM | MFCC | 3-gram | 51.71 |
| DA-XLSR- 53 | Raw audio | None | 45.06 |

Due to very low amount of training data, it is easier to train the GMM-HMM model to a convergent performance result, compared to the DNN-HMM. The TDNN-LSTM is a powerful model. However, even after using volume and speed perturbations, it is still unable to improve much from the GMM-HMM. The proposed model attains a 45.06% WER without using a language model, which is a 12.8 % relative reduction in WER from the best traditional model. Since pretrained models are leveraged, the proposed model has already learned a variety of speech information, as well as environmental noise and sounds from recording equipment, among other distortions. Moreover, multilingual information extends the model's generalization ability over out-of-language data. No language model is used, and greedy search is instead needed for decoding.

Next, we compare the proposed model to various transfer learning approaches, which involve directly fine-tuning the pretrained Wav2Vec2 models with no data augmentation. Also, we evaluate the results against DeepSpeech2, which is another state-of-the-art E2E model. From Table 4.3, it can be seen that the proposed method achieves a relative improvement of 7.39% over the best-performing transfer learning method Wav2Vec2-XLSR-53. While most previous studies suggest that the larger XLS-R-300m model should outperform XLSR-53 or wav2vec2-100k-Voxpopuli [28], [100], it is found that for our dataset and application scenario, the XLSR-53 variant actually outperforms the rest. This is because, although the pretrained model with more language and variation should generally lead to better performance, if the pretraining data is more dissimilar, it lacks the specificity for the target language. The proposed model utilizes more in-domain data, generated from the augmentation module on top of the pretraining information, thus leading to better performance. Since the DeepSpeech2 model is trained from scratch, and this type

Table 4.3 : Results for the proposed model compared to State-of-the-Art E2E models

| Model | Feature | LM | WER % |
|---|---|---|---|
| DeepSpeech2 | Spectrogram | None | 83.12 |
| Wav2vec2-xlsr-53 | Raw audio | None | 48.66 |
| Wav2vec2-xls-r-300m | Raw audio | None | 54.91 |
| Wav2vec2-100k-voxpopuli | Raw audio | None | 50.74 |
| DA-XLSR- 53 | Raw audio | None | 45.06 |

of model requires large amount of data [64], the resulting performance, at 83.12% WER, is worse than the Wav2Vec2 models for the application scenario under investigation.

Last but not least, for a more practical and relevant assessment of system performance, we also justify our results against other published works that also deal with low-resource endangered languages. It should be noted that this type of comparison is not as direct analytically, because there are differences in methodology as well as in datasets. Nevertheless, it should reveal how our proposed model is performing relative to other realistic application scenarios, with similar linguistic constraints and objectives. Jimerson et al [74] studied a data augmentation strategy for Seneca language that used 2 hours and 35 minutes of audio data. They were able to achieve a WER of 59.11% using traditional model with synthetic verb forms. Thai et al [78] extended the research, in order to achieve a WER of 57% using deep speech with transfer learning. A similar approach was also applied to the endangered Tujia Language by Yu et al [115]. This work reported a WER of 46.19% from 2 hour and 54 minutes of data. Another study on the Yoloxochitl Mixtec language by Jiatong et al [68] used a self-supervised model that resulted in a WER of 39.2%, but this required a rather extensive 10-hour subset of their data. They also reported evaluating their model on Puebla Nahuatl language with a 10-hour subset that resulted an WER of 43.7%. By contrast, our proposed model uses a much more limited 1 hour and 9 minutes of data, achieving a WER of 45.06%. Therefore, compared to the existing literature on low-resource endangered languages, our proposed model is highly competitive.

### 4.7.2 Comparison over data size:

We next compare the effect of different data size with the traditional GMM-HMM, hybrid TDNN-LSTM and Deep Speech 2. We train all the models using 10 min, 20 min, 40 min and the full original data. From Figure 4.6, it can be seen that for all the data sizes, the proposed model outperforms the other models. By leveraging pretrained weights from large multilingual datasets, even with only 10 minutes of data, the proposed model still performs better than the others. This


Figure 4.6 Comparison over different data size for different traditional and E2E models.

also provides a practical lower-bound for the minimal amount of data required for such models to train effectively for the Upper Tanana language. As mentioned previously, the Deep Speech 2 is trained from scratch. Therefore, using less than 40 minutes of data shows little to no impact on the training for this approach. It is evident that, for all models, generally better results are achieved with more data. In particular, the performance of the traditional models can be quite close to the proposed model with sufficient data. However, a major difference is that traditional models typically require a lot of preprocessing, along with a pronunciation lexicon and a language model to achieve a similar result.

### 4.7.3 Comparison of ASR systems in terms of model complexity:

We also calculated the training and testing time for each traditional and E2E models. Since we trained the traditional and E2E models on different GPU system the training time is not directly comparable with each other. However, we tested all the models on the same machine using same computational resources. Since for the traditional models we used a different number of utterances in the test set, thus for a fair comparison an average decoding time is also calculated including common utterances for both test set. From Table 4.4, it shows that the training time is much higher for any Deep-learning based model. Although the average decoding time per utterance for Deep Speech 2 model is only 14.89 ms which is significantly less than the others, however in terms of

Table 4.4 : Comparison of training time, testing time of Traditional and E2E models

| Model | # of Utt for test | Train time | Test time | Average Decoding time per Utt. |
|---|---|---|---|---|
| GMM-HMM | 118 | 1 min 30 s | 33s | 279.66 ms |
| TDNN-LSTM | 118 | 52 min 19 s | 40 s | 338.98 ms |
| Deep Speech 2.0 | 182 | 13 hrs 9 min | 2.71s | 14.89 ms |
| DA-XLSR | 182 | 11 hrs 38 min | 17.51s | 96 ms |

performance the model is still far behind than the others, which makes the model practically less efficient. The proposed model requires a decoding time of 96 ms per utterance which is less than half of the traditional models. However, this is notable that we only used greedy search in this research as no language model were used.

### 4.7.4 Consistency of the Augmentation Strategy:

We verify the consistency of the proposed augmentation strategy applying over all the E2E models used in this study. The results are presented in Figure 4.**7**. The figure shows a similar improvement trend for all the wav2vec2 models. As such our previous results the XLSR-53

performs better than other models after applying the augmentation to all the models. For all three wav2vec2 models a relative improvement 6.29% to 8.75% were possible to achieve from 70 epoch of training. In contrast, it shows a big improvement for the deep Speech 2 model attaining around 21.75% relatively. Although the performance is still not satisfactory for the Deep Speech 2 model, however it verifies the augmentation strategy is consistently useful for most E2E models.

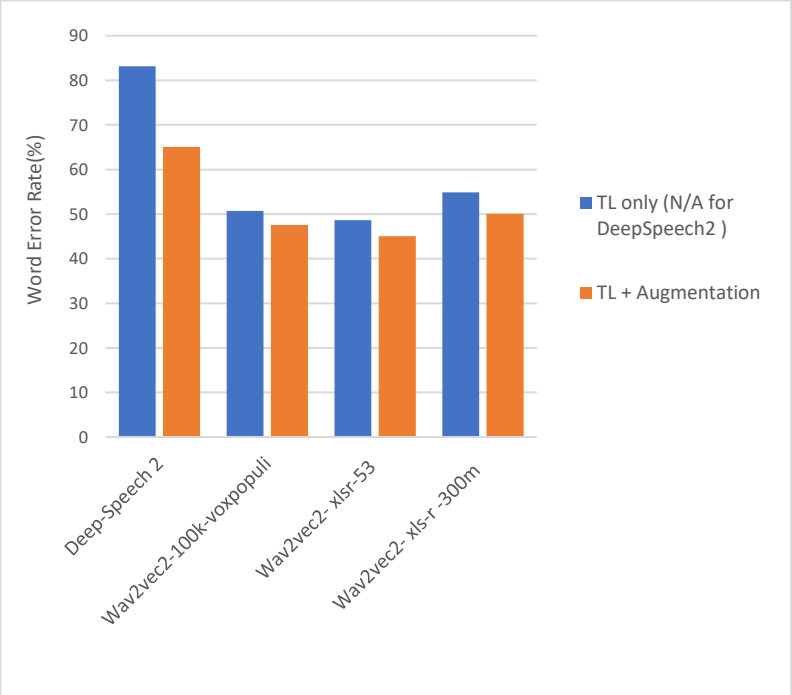


Figure 4.7 The results (WER) of different E2E models before and after applying the augmentation. TL stands for transfer learning.

## 4.8 Conclusion

In this work, an ASR system is proposed for critically endangered language Upper Tanana. This research aims to utilize transfer learning from cross lingual data and data augmentation to provide an effective result for extremely low resource languages when there is no pronunciation or language model directly available. Our experiments demonstrate that such model can exceed the performance of best-known traditional models using only 1 hour and 9 minutes of data without using any language model. It also validates the effectiveness of cross lingual pretrained models for

out of language training. We also focus on further improvement of the result from data augmentation that shows consistency over most state-of-the-art E2E models. Our analysis verifies that larger pretrained models or models including more language does not necessarily perform better. We were able to achieve the best result for Upper Tanana from XLSR-53 that was pretrained on 53 languages. We anticipate the result depends more on the morphological correlation between the languages in the pretraining and the target language. Although the traditional models take much shorter time to train, the decoding time is significantly larger compared to the E2E models. Since this research is motivated by the documentation of the Endangered Languages like Upper Tanana, the shorter decoding time can be immensely helpful for the linguists in their fieldwork. As potential future work, we can consider improving our model, investigating different strategies and data augmentation technique such as GAN, multimodal network etc as well as study other low resource endangered languages in future.

# 5 Conclusion

This final chapter first describes the major contributions of the thesis. Then, some potential future works for research and applications are discussed.

## 5.1 Summary of Contribution

The contributions of the two manuscript forming the body of this thesis are provided in Section 3.3 and 4.3 respectively. Here, we briefly review the major highlights once again on account of completeness:

- This is the first time an ASR system is built for Upper Tanana addressing the challenges of extremely small size of labelled data and no phonetic dictionary.
- An investigation on feature selection and model development for low resource speech recognition is presented based on Upper Tanana corpus.
- An E2E ASR modeling strategy is proposed that delivers better performance compared to other traditional and E2E models without any phonetic dictionary or language model.
- An effective data augmentation technique is provided which can potentially improve the performance of any E2E model.

## 5.2 Future Works

Before we discuss the future direction, it is important to highlight some remaining challenges. Firstly, the experiments of this thesis were conducted on a single low resource language. Although the performance is justified with the size of labelled data, but we don't know how well it is generalized to other critically endangered or morphologically complex languages. Besides, most endangered languages are not easily accessible for ASR training and every language has its own performance curve. Secondly, although during the development we consulted with a field linguist for necessary directions. However, the outputs are not examined at each step by her, rather by an NLP expert. Hence, for future works we need to cross validate the updates in depth by a field linguist first.

While some areas of exploration were investigated into this thesis, some of the future extensions of this work can potentially include:

- Identifying closely related language from large archives of languages and pretrain cross lingual self-supervised model using those selected languages. Although identifying closely related language is a long term burdensome task for endangered languages, but it can potentially improve the performance to a large difference.
- Validate the proposed model on closely related languages for a generalized evaluation. It might be possible to limit the amount of data from the major languages to mimic a similar constrain, however a lot of endangered language does not follow a well-structured grammar or orthography similar to the commonly spoken languages.
- A thorough investigation on generating synthetic data from both acoustic and text data can potentially benefit the acoustic as well as language modeling.

# References


[1] R. Jimerson and E. Prud'hommeaux, "ASR for Documenting Acutely Under-Resourced Indigenous Languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018, pp. 4161–4166. Accessed: Sep. 08, 2022. [Online]. Available: https://aclanthology.org/L18-1657

[2] C. MACAIRE, "Recognizing lexical units in low-resource language contexts with supervised and unsupervised neural networks," LACITO (UMR 7107), Research Report, 2021. Accessed: Mar. 20, 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03429051

[3] B. Foley *et al.*, "Building Speech Recognition Systems for Language Documentation: The CoEDL Endangered Language Pipeline and Inference System (ELPIS)," in *6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, Aug. 2018, pp. 205–209. doi: 10.21437/SLTU.2018-43.

[4] M. Sperber, G. Neubig, J. Niehues, S. Nakamura, and A. Waibel, "Transcribing against time," *Speech Commun.*, vol. 93, pp. 20–30, Oct. 2017, doi: 10.1016/j.specom.2017.07.006.

[5] O. Adams, T. Cohn, G. Neubig, H. Cruz, S. Bird, and A. Michaud, "Evaluating phonemic transcription of low-resource tonal languages for language documentation," p. 11.

[6] K. Precoda, "Non-mainstream Languages and Speech Recognition: Some Challenges," *CALICO J.*, vol. 21, no. 2, p. 15.

[7] "Upper Tanana language, Dialects." https://www.wikizero.com/en/Upper_Tanana_language

[8] B. H. Juang and L. R. Rabiner, "Hidden Markov Models for Speech Recognition," *Technometrics*, vol. 33, no. 3, pp. 251–272, 1991, doi: 10.2307/1268779.

[9] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989, doi: 10.1109/5.18626.

[10] S. Romdhani, “Implementation of DNN-HMM Acoustic Models for Phoneme Recognition,” p. 98.

[11] A. Mohamed, G. E. Dahl, and G. Hinton, “Acoustic Modeling Using Deep Belief Networks,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012, doi: 10.1109/TASL.2011.2109382.

[12] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A Fast Learning Algorithm for Deep Belief Nets,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, doi: 10.1162/neco.2006.18.7.1527.

[13] J. Oruh, S. Viriri, and A. Adegun, “Long Short-Term Memory Recurrent Neural Network for Automatic Speech Recognition,” *IEEE Access*, vol. 10, pp. 30069–30079, 2022, doi: 10.1109/ACCESS.2022.3159339.

[14] H. Sak, A. Senior, and F. Beaufays, “Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition,” *ArXiv14021128 Cs Stat*, Feb. 2014, Accessed: Apr. 24, 2022. [Online]. Available: http://arxiv.org/abs/1402.1128

[15] D. bukhari, Y. Wang, and H. Wang, “Multilingual Convolutional, Long Short-Term Memory, Deep Neural Networks for Low Resource Speech Recognition,” *Procedia Comput. Sci.*, vol. 107, pp. 842–847, Dec. 2017, doi: 10.1016/j.procs.2017.03.179.

[16] H. Sak, A. Senior, K. Rao, and F. Beaufays, “Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition,” *ArXiv150706947 Cs Stat*, Jul. 2015, Accessed: Apr. 25, 2022. [Online]. Available: http://arxiv.org/abs/1507.06947

[17] A. Graves, A. Mohamed, and G. Hinton, “Speech Recognition with Deep Recurrent Neural Networks,” *ArXiv13035778 Cs*, Mar. 2013, Accessed: Apr. 25, 2022. [Online]. Available: http://arxiv.org/abs/1303.5778

[18] M. Karafiát *et al.*, “BUT OpenSAT 2017 Speech Recognition System,” in *Interspeech 2018*, Sep. 2018, pp. 2638–2642. doi: 10.21437/Interspeech.2018-2457.

[19] H. Krishna, K. Gurugubelli, V. Vegesna, and A. Vuppala, *An Exploration towards Joint Acoustic Modeling for Indian Languages: IIIT-H Submission for Low Resource Speech Recognition Challenge for Indian Languages, INTERSPEECH 2018*. 2018, p. 3196. doi: 10.21437/Interspeech.2018-1584.

[20] M. Karafidt, M. K. Baskar, K. Vesely, F. Grezl, L. Burget, and J. Cernocky, "Analysis of Multilingual Blstm Acoustic Model on Low and High Resource Languages," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Apr. 2018, pp. 5789–5793. doi: 10.1109/ICASSP.2018.8462083.

[21] P. Doetsch, M. Kozielski, and H. Ney, "Fast and Robust Training of Recurrent Neural Networks for Offline Handwriting Recognition," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, Greece, Sep. 2014, pp. 279–284. doi: 10.1109/ICFHR.2014.54.

[22] K. Chen, "Training Deep Bidirectional LSTM Acoustic Model for LVCSR by a Context-Sensitive-Chunk BPTT Approach," vol. 24, no. 7, p. 9, 2016.

[23] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Interspeech 2015*, Sep. 2015, pp. 3214–3218. doi: 10.21437/Interspeech.2015-647.

[24] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan, "An Exploration of Dropout with LSTMs," in *Interspeech 2017*, Aug. 2017, pp. 1586–1590. doi: 10.21437/Interspeech.2017-129.

[25] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low Latency Acoustic Modeling Using Temporal Convolution and LSTMs," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 373–377, Mar. 2018, doi: 10.1109/LSP.2017.2723507.

[26] V. Bataev, M. Korenevsky, I. Medennikov, and A. Zatvornitskiy, "Exploring End-to-End Techniques for Low-Resource Speech Recognition," *ArXiv180700868 Cs Eess*, Jul. 2018, Accessed: Feb. 27, 2022. [Online]. Available: http://arxiv.org/abs/1807.00868

[27] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, “Applying Wav2vec2.0 to Speech Recognition in Various Low-resource Languages,” *ArXiv201212121 Cs*, pp. 1–5, Jan. 2021.

[28] R. Al-Ghezi, Y. Getman, A. Rouhe, R. Hildén, and M. Kurimo, “Self-Supervised End-to-End ASR for Low Resource L2 Swedish,” in *Interspeech 2021*, Aug. 2021, pp. 1429–1433. doi: 10.21437/Interspeech.2021-1710.

[29] R. Rasipuram, “Grapheme-based Automatic Speech Recognition using Probabilistic Lexical Modeling,” EPFL, Lausanne, 2014. doi: 10.5075/epfl-thesis-6280.

[30] J. Zuluaga-Gomez *et al.*, “How Does Pre-trained Wav2Vec2.0 Perform on Domain Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications,” *ArXiv220316822 Cs Eess*, pp. 1–5, Mar. 2022.

[31] N. Indurkhya and F. J. Damerau, Eds., “An Overview of Modern Speech Recognition Xuedong Huang and Li Deng,” in *Handbook of Natural Language Processing*, 0 ed., Chapman and Hall/CRC, 2010, pp. 363–390. doi: 10.1201/9781420085938-24.

[32] H. F. Pardede, V. Zilvan, D. Krisnandi, A. Heryana, and R. B. S. Kusumo, “Generalized Filter-bank Features for Robust Speech Recognition Against Reverberation,” in *2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, Oct. 2019, pp. 19–24. doi: 10.1109/IC3INA48034.2019.8949593.

[33] H. Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” *J. Acoust. Soc. Am.*, vol. 87, no. 4, pp. 1–15, 1990, doi: 10.1121/1.399423.

[34] X. Huang, A. Acero, and H.-W. Hon, *Spoken language processing: a guide to theory, algorithm, and system development*. Upper Saddle River, NJ: Prentice Hall PTR, 2001.

[35] E. Chuangsuwanich, “Multilingual Techniques for Low Resource Automatic Speech Recognition,” p. 143.

[36] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, “A pitch extraction algorithm tuned for automatic speech recognition,” in *2014 IEEE*

*International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 2494–2498. doi: 10.1109/ICASSP.2014.6854049.

[37] S. R. Madikeri, S. Dey, P. Motlícek, and M. Ferras, "Implementation of the Standard I-vector System for the Kaldi Speech Recognition Toolkit," 2016.

[38] N. V. Prasad and S. Umesh, "Improved cepstral mean and variance normalization using Bayesian framework," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec. 2013, pp. 156–161. doi: 10.1109/ASRU.2013.6707722.

[39] K. P. Simha, "Improving Automatic Speech Recognition on Endangered Languages," p. 89.

[40] H. Yadav and S. Sitaram, "A Survey of Multilingual Models for Automatic Speech Recognition." arXiv, Feb. 25, 2022. Accessed: Nov. 13, 2022. [Online]. Available: http://arxiv.org/abs/2202.12576

[41] P. A. Devijver, "Baum's forward-backward algorithm revisited," *Pattern Recognit. Lett.*, vol. 3, no. 6, pp. 369–373, Dec. 1985, doi: 10.1016/0167-8655(85)90023-6.

[42] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967, doi: 10.1109/TIT.1967.1054010.

[43] E. Morris, "Automatic Speech Recognition for Low-Resource and Morphologically Complex Languages," p. 76.

[44] W. Wang, X. Yang, and H. Yang, "End-to-End Low-Resource Speech Recognition with a Deep CNN-LSTM Encoder," in *2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP)*, Sep. 2020, pp. 158–162. doi: 10.1109/ICICSP50920.2020.9232119.

[45] C. Liu, S. Mallard, and R. Silva, "Improving Conversational Forced Alignment with Lexicon Expansion," p. 8.

[46] A. Graves, “Sequence Transduction with Recurrent Neural Networks,” *ArXiv12113711 Cs Stat*, Nov. 2012, Accessed: Apr. 21, 2022. [Online]. Available: http://arxiv.org/abs/1211.3711

[47] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, “Listen, Attend and Spell.” arXiv, Aug. 19, 2015. Accessed: Nov. 14, 2022. [Online]. Available: http://arxiv.org/abs/1508.01211

[48] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-Based Models for Speech Recognition.” arXiv, Jun. 24, 2015. Accessed: Nov. 14, 2022. [Online]. Available: http://arxiv.org/abs/1506.07503

[49] Y. Zhang, W. Chan, and N. Jaitly, “Very Deep Convolutional Networks for End-to-End Speech Recognition.” arXiv, Oct. 10, 2016. Accessed: Nov. 14, 2022. [Online]. Available: http://arxiv.org/abs/1610.03022

[50] H. Soltau, H. Liao, and H. Sak, “Neural Speech Recognizer: Acoustic-to-Word LSTM Model for Large Vocabulary Speech Recognition.” arXiv, Oct. 31, 2016. Accessed: Nov. 14, 2022. [Online]. Available: http://arxiv.org/abs/1610.09975

[51] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, “Self-Supervised Representation Learning: Introduction, advances, and challenges,” *IEEE Signal Process. Mag.*, vol. 39, no. 3, pp. 42–62, May 2022, doi: 10.1109/MSP.2021.3134634.

[52] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks,” p. 8.

[53] Y. Getman, “End-to-End Low-Resource Automatic Speech Recognition for Second Language Learners,” p. 67.

[54] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate.” arXiv, May 19, 2016. Accessed: Nov. 14, 2022. [Online]. Available: http://arxiv.org/abs/1409.0473

[55] F. Cong, W. Hu, Q. Huo, and L. Guo, “A Comparative Study of Attention-Based Encoder-Decoder Approaches to Natural Scene Text Recognition,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sep. 2019, pp. 916–921. doi: 10.1109/ICDAR.2019.00151.

[56] A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 2017, vol. 30, pp. 1–15. Accessed: Sep. 10, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[57] A. Hannun, “Sequence Modeling with CTC,” *Distill*, vol. 2, no. 11, p. e8, Nov. 2017, doi: 10.23915/distill.00008.

[58] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural Discrete Representation Learning.” arXiv, May 30, 2018. Accessed: Nov. 15, 2022. [Online]. Available: http://arxiv.org/abs/1711.00937

[59] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised Pre-training for Speech Recognition.” arXiv, Sep. 11, 2019. Accessed: Nov. 15, 2022. [Online]. Available: http://arxiv.org/abs/1904.05862

[60] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” *ArXiv200611477 Cs Eess*, pp. 1–19, Oct. 2020.

[61] A. T. Liu, S. Yang, P.-H. Chi, P. Hsu, and H. Lee, “Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 6419–6423. doi: 10.1109/ICASSP40776.2020.9054458.

[62] P.-H. Chi *et al.*, “Audio Albert: A Lite Bert for Self-Supervised Learning of Audio Representation,” in *2021 IEEE Spoken Language Technology Workshop (SLT)*, Shenzhen, China, Jan. 2021, pp. 344–350. doi: 10.1109/SLT48900.2021.9383575.

[63] A. Hannun *et al.*, “Deep Speech: Scaling up end-to-end speech recognition,” *ArXiv14125567 Cs*, pp. 1–12, Dec. 2014.

[64] D. Amodei *et al.*, “Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin,” in *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA, 2016, vol. 48, pp. 173–182.

[65] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[66] A. Anastasopoulos and D. Chiang, “A case study on using speech-to-translation alignments for language documentation,” in *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, Honolulu, Mar. 2017, pp. 170–178. doi: 10.18653/v1/W17-0123.

[67] T.-N.-D. Do, A. Michaud, and E. Castelli, “Towards the automatic processing of Yongning Na (Sino-Tibetan): developing a ‘light’ acoustic model of the target language and testing ‘heavyweight’ models from five national languages,” p. 11.

[68] J. Shi, J. D. Amith, R. Castillo García, E. Guadalupe Sierra, K. Duh, and S. Watanabe, “Leveraging End-to-End ASR for Endangered Language Documentation: An Empirical Study on Yolóxochitl Mixtec,” in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online, 2021, pp. 1134–1145. doi: 10.18653/v1/2021.eacl-main.96.

[69] G. Saon, H.-K. J. Kuo, S. Rennie, and M. Picheny, “The IBM 2015 English Conversational Telephone Speech Recognition System.” arXiv, May 21, 2015. Accessed: May 23, 2022. [Online]. Available: http://arxiv.org/abs/1505.05899

[70] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, “The Microsoft 2017 Conversational Speech Recognition System,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5934–5938. doi: 10.1109/ICASSP.2018.8461870.

[71] C.-C. Chiu *et al.*, “State-of-the-art Speech Recognition With Sequence-to-Sequence Models.” arXiv, Feb. 23, 2018. Accessed: May 23, 2022. [Online]. Available: http://arxiv.org/abs/1712.01769

[72] M. Ćavar, D. Ćavar, and H. Cruz, “Endangered Language Documentation: Bootstrapping a Chatino Speech Corpus, Forced Aligner, ASR,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, Portorož, Slovenia, May 2016, pp. 4004–4011. Accessed: Oct. 11, 2022. [Online]. Available: https://aclanthology.org/L16-1632

[73] A. Zahrer, A. Zgank, and B. Schuppler, “Towards Building an Automatic Transcription System for Language Documentation: Experiences from Muyu,” in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Marseille, France, May 2020, pp. 2893–2900. Accessed: Oct. 11, 2022. [Online]. Available: https://aclanthology.org/2020.lrec-1.353

[74] R. Jimerson, K. Simha, R. Ptucha, and E. Prudhommeaux, “Improving ASR Output for Endangered Language Documentation,” in *6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, Gurugram, India, Aug. 2018, pp. 187–191. doi: 10.21437/SLTU.2018-39.

[75] O. Lovick, *A Grammar of Upper Tanana, Volume 1: Phonology, Lexical Classes, Morphology*, vol. 1. University of Nebraska Press, 2020. Accessed: Oct. 11, 2022. [Online]. Available: https://www.jstor.org/stable/j.ctvvh8522

[76] S. Thomas, M. L. Seltzer, K. Church, and H. Hermansky, “Deep neural network features and semi-supervised training for low resource speech recognition,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6704–6708. doi: 10.1109/ICASSP.2013.6638959.

[77] D. Povey *et al.*, Eds., “The Kaldi Speech Recognition Toolkit,” in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, 2011, pp. 1–4. [Online]. Available: https://www.danielpovey.com/files/2011_asru_kaldi.pdf

[78] B. Thai, R. Jimerson, D. Arcoraci, E. Prud’hommeaux, and R. Ptucha, “Synthetic Data Augmentation for Improving Low-Resource ASR,” in *2019 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, Rochester, NY, USA, Oct. 2019, pp. 1–9. doi: 10.1109/WNYIPW.2019.8923082.

[79] E. Prud’hommeaux, R. Jimerson, R. Hatcher, and K. Michelson, “Automatic Speech Recognition for Supporting Endangered Language Documentation,” *Lang. Doc.*, vol. 15, pp. 491–513, 2021.

[80] “UNESCO Atlas of the World’s Languages in danger.” http://www.unesco.org/languages-atlas/en/atlasmap.html (accessed Apr. 20, 2022).

[81] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, “Automatic speech recognition for under-resourced languages: A survey,” *Speech Commun.*, vol. 56, pp. 85–100, Jan. 2014, doi: 10.1016/j.specom.2013.07.008.

[82] M. A. Hasegawa-Johnson *et al.*, “ASR for Under-Resourced Languages From Probabilistic Transcription,” *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 1, pp. 50–63, Jan. 2017, doi: 10.1109/TASLP.2016.2621659.

[83] L. Meng, J. Xu, X. Tan, J. Wang, T. Qin, and B. Xu, “MixSpeech: Data Augmentation for Low-Resource Automatic Speech Recognition,” in *ICASSP 2021 - 2021 IEEE International*

*Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, Jun. 2021, pp. 7008–7012. doi: 10.1109/ICASSP39728.2021.9414483.

[84] K. Matsuura, M. Mimura, S. Sakai, and T. Kawahara, "Generative Adversarial Training Data Adaptation for Very Low-resource Automatic Speech Recognition," *ArXiv200509256 Cs Eess*, Jul. 2020, Accessed: Feb. 27, 2022. [Online]. Available: http://arxiv.org/abs/2005.09256

[85] N. Fathima, T. Patel, M. C, and A. Iyengar, "TDNN-based Multilingual Speech Recognition System for Low Resource Indian Languages," in *Interspeech 2018*, Sep. 2018, pp. 3197–3201. doi: 10.21437/Interspeech.2018-2117.

[86] "Upper Tanana | Alaska Native Language Archive | Alaska Native Language Archive." https://www.uaf.edu/anla/collections/upper_tanana/ (accessed Oct. 20, 2022).

[87] tracy, "About the Upper Tanana language - Yukon Native Language Centre," May 05, 2022. https://ynlc.ca/about-the-upper-tanana-language/ (accessed Oct. 20, 2022).

[88] O. Lovick and S. G. Tuttle, "Conversation in Upper Tanana Athabascan: syntactic and prosodic patterns," p. 46.

[89] O. Lovick, "The identification of narrative genres in Upper Tanana Athabascan: a preliminary study," *NORTHWEST J. Linguist.*, vol. 6, no. 1, pp. 1–29, 2012.

[90] O. Lovick, *Teedląy t'iin naholndak niign: Stories of the Tetlin people By Cora H. David*, 2nd, expanded edition ed. Alaska Native Language Center: University of Alaska Fairbanks, 2017.

[91] O. Lovick, "Documentation of Upper Tanana (Curated dataset), part of Olga Lovick Collection." Alaska Native Languages Archive, Present 2006. [Online]. Available: https://www.uaf.edu/anla/

[92] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes, "ELAN: a Professional Framework for Multimodality Research," p. 4.

[93] W. D. Basson and M. H. Davel, “Comparing grapheme-based and phoneme-based speech recognition for Afrikaans,” p. 5.

[94] A. Stan, P. Bell, and S. King, “A grapheme-based method for automatic alignment of speech and text data,” in *2012 IEEE Spoken Language Technology Workshop (SLT)*, Miami, FL, USA, Dec. 2012, pp. 286–290. doi: 10.1109/SLT.2012.6424237.

[95] M. Bisani and H. Ney, “Joint-sequence models for grapheme-to-phoneme conversion,” *Speech Commun.*, vol. 50, no. 5, pp. 434–451, May 2008, doi: 10.1016/j.specom.2008.01.002.

[96] S. G. Tuttle, O. Lovick, and I. Núñez-Ortiz, “Vowels of Upper Tanana Athabascan,” *J. Int. Phon. Assoc.*, vol. 41, no. 3, pp. 283–312, Dec. 2011, doi: 10.1017/S002510031100034X.

[97] A.-L. Georgescu, H. Cucu, and C. Burileanu, “Kaldi-based DNN Architectures for Speech Recognition in Romanian,” in *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, Oct. 2019, pp. 1–6. doi: 10.1109/SPED.2019.8906555.

[98] D. Wang, X. Wang, and S. Lv, “An Overview of End-to-End Automatic Speech Recognition,” *Symmetry*, vol. 11, no. 8, pp. 1–27, Aug. 2019, doi: 10.3390/sym11081018.

[99] C. Yi, S. Zhou, and B. Xu, “Efficiently Fusing Pretrained Acoustic and Linguistic Encoders for Low-Resource Speech Recognition,” *IEEE Signal Process. Lett.*, vol. 28, pp. 788–792, 2021, doi: 10.1109/LSP.2021.3071668.

[100] J. Zhao and W.-Q. Zhang, “Improving Automatic Speech Recognition Performance for Low-Resource Languages with Self-Supervised Models,” *IEEE J. Sel. Top. Signal Process.*, pp. 1–8, 2022, doi: 10.1109/JSTSP.2022.3184480.

[101] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, “S4L: Self-Supervised Semi-Supervised Learning,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), Oct. 2019, pp. 1476–1485. doi: 10.1109/ICCV.2019.00156.

[102] X. Chang *et al.*, “An Exploration of Self-Supervised Pretrained Representations for End-to-End Speech Recognition,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Cartagena, Colombia, Dec. 2021, pp. 228–235. doi: 10.1109/ASRU51503.2021.9688137.

[103] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, “Unsupervised Cross-lingual Representation Learning for Speech Recognition,” *ArXiv200613979 Cs Eess*, pp. 1–12, Dec. 2020.

[104] A. Babu *et al.*, “XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale,” *ArXiv211109296 Cs Eess*, pp. 1–23, Dec. 2021.

[105] O. Lovick, C. Cox, M. Silfverberg, A. Arppe, and M. Hulden, “A Computational Architecture for the Morphology of Upper Tanana,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018, pp. 1874–1879. Accessed: Sep. 08, 2022. [Online]. Available: https://aclanthology.org/L18-1294

[106] S. Qin, L. Wang, S. Li, J. Dang, and L. Pan, “Improving low-resource Tibetan end-to-end ASR by multilingual and multilevel unit modeling,” *EURASIP J. Audio Speech Music Process.*, vol. 2022, no. 1, pp. 1–10, Dec. 2022, doi: 10.1186/s13636-021-00233-4.

[107] C. Wang *et al.*, “VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online, Aug. 2021, pp. 993–1003. doi: 10.18653/v1/2021.acl-long.80.

[108] C. Yu, M. Kang, Y. Chen, J. Wu, and X. Zhao, “Acoustic Modeling Based on Deep Learning for Low-Resource Speech Recognition: An Overview,” *IEEE Access*, vol. 8, pp. 163829–163843, 2020, doi: 10.1109/ACCESS.2020.3020421.

[109] L. Tóth, G. Kovács, and D. Van Compernolle, “A Perceptually Inspired Data Augmentation Method for Noise Robust CNN Acoustic Models,” in *Speech and Computer*, vol. 11096, A. Karpov, O. Jokisch, and R. Potapova, Eds. Cham: Springer International Publishing, 2018, pp. 697–706. doi: 10.1007/978-3-319-99579-3_71.

[110] S. Shahnawazuddin, A. Dey, and R. Sinha, “Pitch-Adaptive Front-End Features for Robust Children’s ASR,” in *Interspeech 2016*, San Francisco, CA, Sep. 2016, pp. 3459–3463. doi: 10.21437/Interspeech.2016-1020.

[111] D. S. Park *et al.*, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” in *Interspeech 2019*, Graz, Austria, Sep. 2019, pp. 2613–2617. doi: 10.21437/Interspeech.2019-2680.

[112] S. Wei, S. Zou, F. Liao, and weimin lang, “A Comparison on Data Augmentation Methods Based on Deep Learning for Audio Classification,” *J. Phys. Conf. Ser.*, vol. 1453, no. 1, pp. 1–8, Jan. 2020, doi: 10.1088/1742-6596/1453/1/012085.

[113] A. Stolcke, “SRILM - an extensible language modeling toolkit,” in *7th International Conference on Spoken Language Processing (ICSLP 2002)*, Denver, Colorado, USA, Sep. 2002, pp. 901–904. doi: 10.21437/ICSLP.2002-303.

[114] “ARPABET,” *Wikipedia*. Dec. 29, 2021. Accessed: Jun. 06, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=ARPABET&oldid=1062602312

[115] C. Yu, Y. Chen, Y. Li, M. Kang, S. Xu, and X. Liu, “Cross-Language End-to-End Speech Recognition Research Based on Transfer Learning for the Low-Resource Tujia Language,” *Symmetry*, vol. 11, no. 2, pp. 1–14, Feb. 2019, doi: 10.3390/sym11020179.