UNIVERSITY OF ALBERTA

# AUTOMATIC SPEECH RECOGNITION OF LOW RESOURCE LANGUAGES

BY

ALLAN PLESNIARSKI

A THESIS SUBMITTED TO THE FACULTY OF ARTS IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ARTS

DEPARTMENT OF LINGUISTICS

EDMONTON, ALBERTA

April, 2024

UNIVERSITY OF ALBERTA FACULTY OF ARTS

# ABSTRACT

Automatic Speech Recognition (ASR) can play a significant role in the documentation of low-resource languages. There is typically a lack of labeled data for low-resource languages to build accurate ASR systems. We will evaluate some of the current approaches for neural network models to enhance ASR performance for such languages. Current ASR approaches span from building a language model from scratch such as ELPIS to fine-tuning existing multi-lingual models such as Meta Research MMS. ASR performance error rates can range widely with greater amounts of annotated training data resulting in lower error rates. Self-supervised learning with multi-lingual models is explored as a practical solution for advancing ASR technology in low-resource linguistic contexts. By harnessing the capabilities of neural network models, we pave the way for more inclusive and accurate speech recognition systems. This study focused on exploring ASR systems primarily for the transcription of the Totonac languages of Coatepec and Upper Necaxa. Best ASR transcription results were achieved using Meta Research MMS multilingual model with the Wave2vec ASR framework. The Totonac languages were transcribed with a reasonable Phoneme Error rate based on the Highland Totonac language trained into the Meta MMS model. The transcription accuracy of consonants is higher than vowels, giving a linguistic researcher an automatically transcribed template that can serve as the basis for manual fine-tuning of phonemes and word boundaries.

# ACKNOWLEDGEMENTS

I would like to acknowledge my supervisor, Dr. Timothy Mills, for his guidance in the phonology context of this paper, and in the overall structure of the paper. Acknowledgements also go to Dr. David Beck for providing access to Totonac corpora, and to the individuals who originally gathered and transcribed the corpora. Their combined focus facilitated the production of a working ASR system for an endangered low-resource language.

TABLE OF CONTENTS

# 1. AUTOMATIC SPEECH RECOGNITION METHODOLOGY

A number of successive approaches have been used over the past few years for Automatic Speech Recognition (ASR) of Low-Resource Languages (LRL). Typically, there are not enough voice recordings of LRL to train a model for ASR based solely on the available recordings. Therefore, most ASR methods for LRL employ models trained from High-Resource languages (HRL). Each successive approach has used a more advanced computational method to train an ASR model. For the purposes of this paper, the overall methodology will be briefly described, but essentially treated as a black box for the remainder of this paper.

## 1.1 Speech Stream Segmentation

The first step in the ASR process is to segment the speech stream into consistent time steps called frames to feed forward into the process. A typical time step duration is 10 milliseconds, as shown in Figure 1 below.

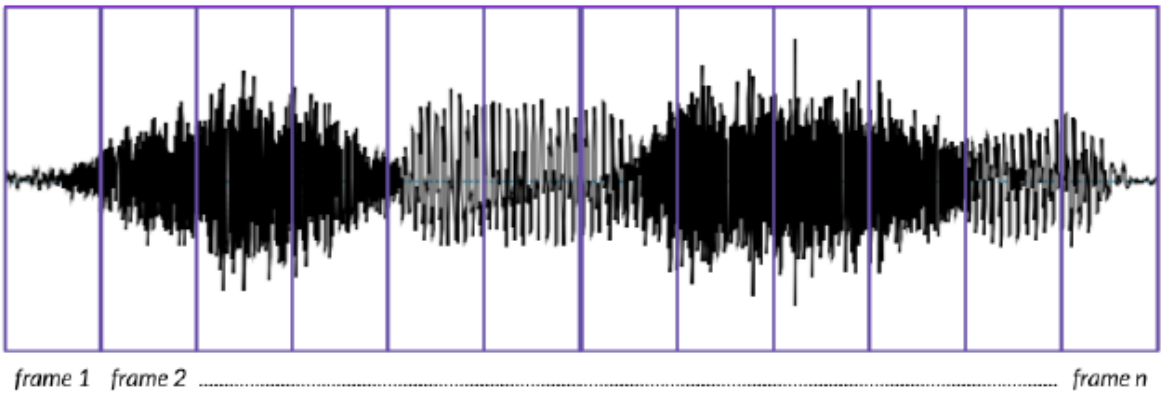


Figure 1. An audio waveform diagram of a speech stream splitting process (Macaire, 2021)

## 1.2 Feature Extraction

The next second step in the ASR process is to characterize frequencies of the

speech frames into a matrix of numbers that can be used in computational processes. Figure 2 shows how a spectrogram of frequency intensity along the time axis is converted into a logarithmic scale of numbers.

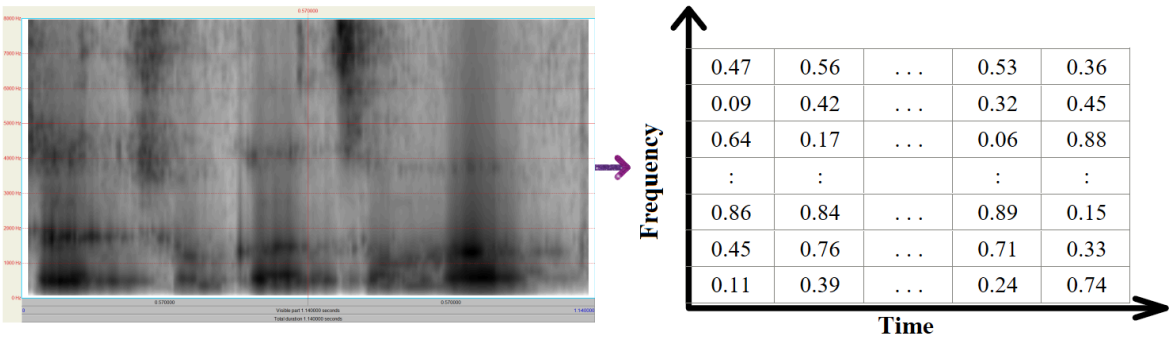


Figure 2. Speech spectrogram frame to frequency pattern extraction

### 1.3 Neural Network

The third step in the ASR process is the pre-processing stage of a neural network that groups the frequency pattern matrix frames into similar patterns encompassing time steps larger than each frame. This step is typically done by a Convolutional layer employing an algorithm called Connectionist Temporal Classification (CTC). Figure 3 shows an example of CTC output for the spoken word “hubble” (Tjandra et al., 2022). The speech frames along the time steps t1..t10 are a figurative representation of the spoken letters, or phones, in the word. In computational terms, the speech frames are numeric matrices as described earlier. Note that the letter “e” at the end of the word is silent and has no representation. Also there are relatively silent speech frames within the word, represented here by the letter epsilon. Speech frames that are consecutively similar are merged into one representation, as in the letter “h”. However the epsilon frame in this case prevents the two sequential, yet not unified, “b” representations from being merged. Finally the empty placeholder frames are removed and the grouped output, in numeric matrix form,

is passed on to the next stage.

| Time step | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 |
|---|---|---|---|---|---|---|---|---|---|---|
| speech frame | **h** | **h** | **u** | **u** | **ϵ** | **b** | **ϵ** | **b** | **l** | **l** |
| merge | h | | u | | ϵ | b | ϵ | b | l | |
| delete ϵ | h | | u | | | b | | b | l | |
| output | h | u | b | b | l | | | | | |

Figure 3. Convolutional Layer processing of the word “hubble”

### 1.4 Encoder - Decoder Architecture

The fourth step in the ASR neural network is the encoder-decoder function. The encoder function can be described as adding probabilities based on the sequence of grouped speech frame representations. This combination of properties creates an interim probability state of each representation that is then passed on to the decoder, as shown in Figure 4.

The decoder function is to initially train a neural network based on the probability states of the input speech signal in combination with a pre-transcribed speech signal. Thus the decoder output is trained in a supervised way by comparison to actual known output. The decoder passes the most probable speech representations to an output layer that emits the ASR transcription.

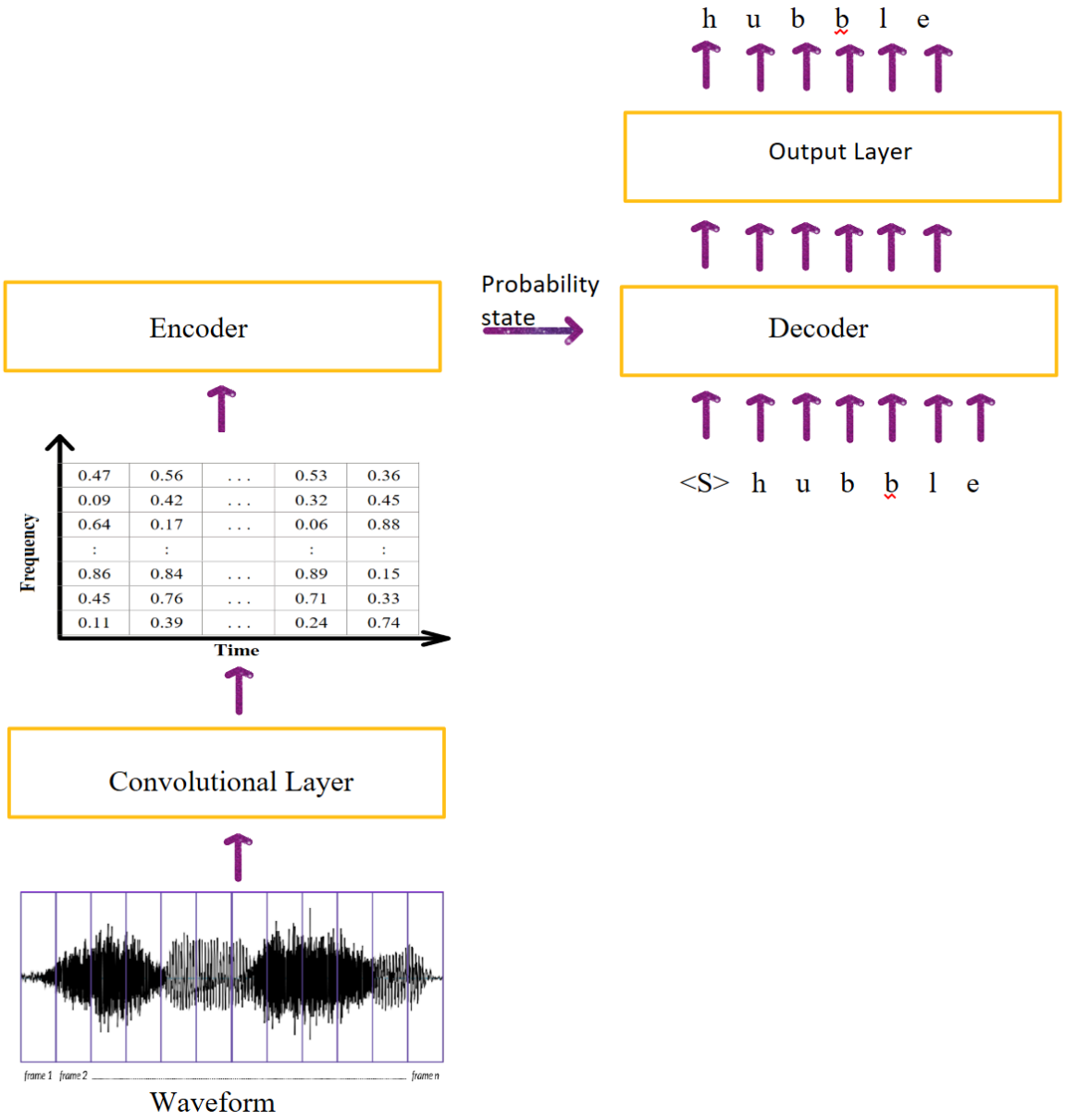


Figure 4. Overall ASR architecture

### 1.5 ASR Training Methodology

Initially an ASR neural network model is "trained" by passing transcribed text into the decoder, as shown in Figure 4. Note that the first character is a placeholder token <S> that indicates the start of an utterance. The decoder calculates the probability of the character in the next probability state based on the previously transcribed character. So for

example the probability of the second character “u” in “hubble” will be calculated based on the input of the previously transcribed character “h”. Note that the previously transcribed character “e” is passed into the model to combine with the prior character “l”. The sequential probabilities are combined to form a word representation at the output layer.

### 1.6 ASR Inference Methodology

Once an ASR model has been sufficiently trained to provide consistent phone and word representations of speech for a particular language, the model can be used to infer speech outside of that used to train and test the model. At inference time, the transcription passed into the decoder comes from the output of the model itself, not a pre-existing transcription. So for example in this case the output of the character “u” in “hubble” becomes the input to calculate the probability of the next character “b”. Note that the silent ‘e’ is not in the speech stream, yet the model generates it in the output based on its probability with the previous sequence of characters.

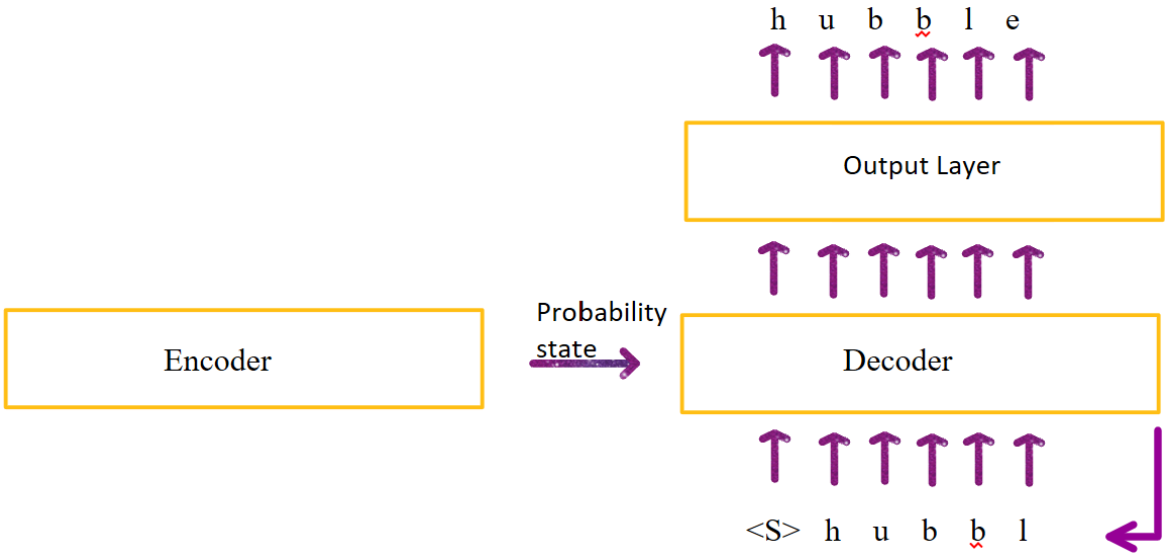


Figure 5. ASR Inference

# 2. HISTORY OF ASR SYSTEMS

Speech recognition accuracy has been measured at different levels depending on the ASR method used: word, token, character and phoneme. Word Error Rate (WER) is based on each individual word in the transcription. Token error rate (TER%) is measured by the ratio of different words, including special characters like capitals, and punctuation marks from the intended text. Character error rate (CER) is based on each individual character in the transcription. Phoneme Error Rate (PER) is based on each distinct phoneme in the transcription. The Error Rate calculation is the same for all:

$$ER = (S + D + I) / N$$

where Error Rate (ER) is based on the sum of the number of substitutions (S) plus the number of deletions (D) plus the number of insertions (I), divided by the number of correct items (N).

## 2.1 ASR for a Low Resource Language based on a High Resource Language

Scharenborg et al. (2017) used Dutch as a HRL and English as a LRL, since there are phones in English like /æ/ and /θ/ which are absent in Dutch. Their method consisted of three-steps: (1) build an ASR on the HRL Dutch, (2) transfer the phone inventory of the LRL English to the HRL ASR, and (3) iteratively train the model using the self-labeled phone sequences. The model was trained using Connectionist Temporal Classification (CTC) which maps the speech signal into most probable phones.

The results were measured in the results achieved a TER between 71.67 and 72.52, which is comparable to the phone error rates of cross-language ASR systems. Note that a measure by TER is typically higher than Character error rate (CER%), because CER considers each successive character to be a token without considering spaces.

### 2.2 ESPnet - End-to-End Speech Processing Toolkit

Watanabe et al. (2018) created an "End-to-End Speech Processing Toolkit" called ESPnet. They used a software package called Kaldi to perform acoustic feature extraction of audio files that were fed into a model training method using a more advanced combination of CTC and attention based decoder network. An attention based decoder network essentially focuses on the probability of each word of the input in a serial manner. The flow of the ESPnet process stages is shown in Figure 6.

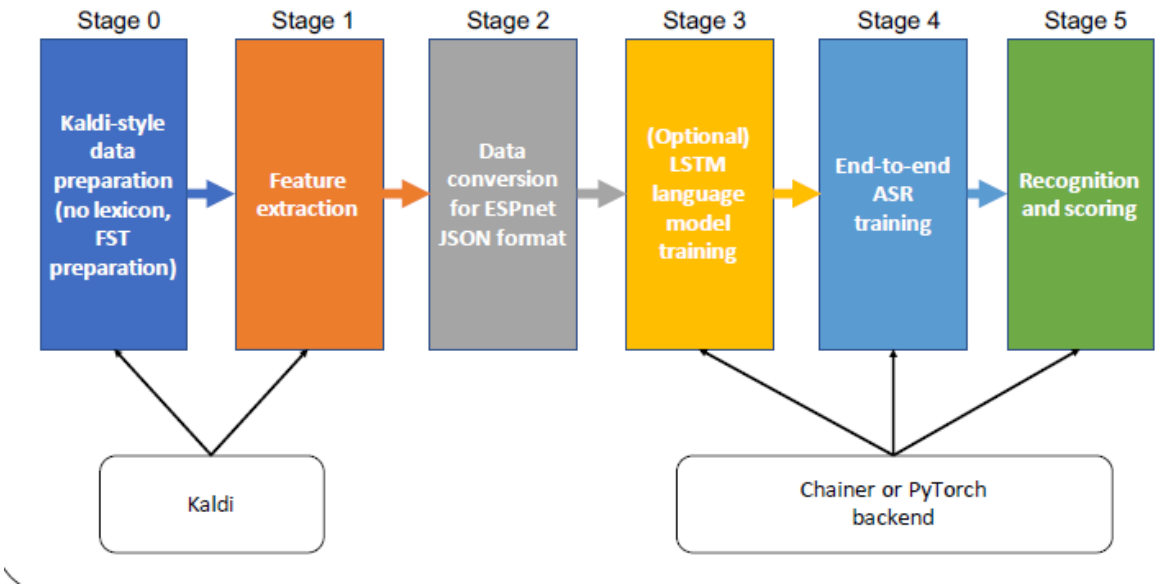


Figure 6. ESPnet ASR Stages (Watanabe et al., 2018)

At Stage 0, the raw audio is input into Kaldi to extract audio features for Stage 1, then converted in Stage 2 into a JSON format containing all the audio properties necessary for ESPnet training. The ESPnet program backend encompasses the remaining stages. Stage 3 is an optional Long-Short Term Memory (LSTM) network for character based training. Stage 4 is the ESPnet main neural network consisting of a combination of CTC and encoder-decoder transformer. End-to-end ASR training is a machine learning technique where a single neural network is trained to transform input data directly into output using a

single model. In Stage 5 the output of the previous two stages is measured.

ESPnet was benchmarked on a standardized ASR benchmark called the Wall Street Journal (WSJ) task. ESPnet achieved results of a CER between 3.6 to 10.1 depending on the number of iterations, and a WER between 8.9 and 12.4.

### 2.3 ELPIS - Endangered Language Pipeline and Inference System

An ASR system called the Endangered Language Pipeline and Inference System (ELPIS) was designed by Foley et al. (2018) to encapsulate the various stages an ASR system goes through as shown above. ELPIS uses a virtual machine called a Docker container to instantiate a website tied to the various backend functions like Kaldi. ELPIS is targeted at linguistic researchers that are beginning to work with a volume of audio recordings of their studied languages. ELPIS is capable of using ELAN transcriptions with audio files to build either word level or phoneme level transcriptions. The word level models are built from empty, so the greater the training data, the lower the WER. The WER for less than one hour of training data can range from approximately 40 to 80 depending on the language recordings (Foley et al., 2018).

### 2.4 Wav2vec - Waveform-to-Vector

Wav2vec is an ASR system developed by the AI team at Meta (Alexei Baevski et al., 2020). Unlike Kaldi and other ASR frameworks that rely on pre-transcribed training and test data, wav2vec uses a self-supervised approach to training data. Wav2vec trains models by generating self-modified versions of speech audio and comparing them to the original audio. This self-supervised fine tuning is iteratively repeated several times for each time stepped representation of audio as shown in Figure 7.

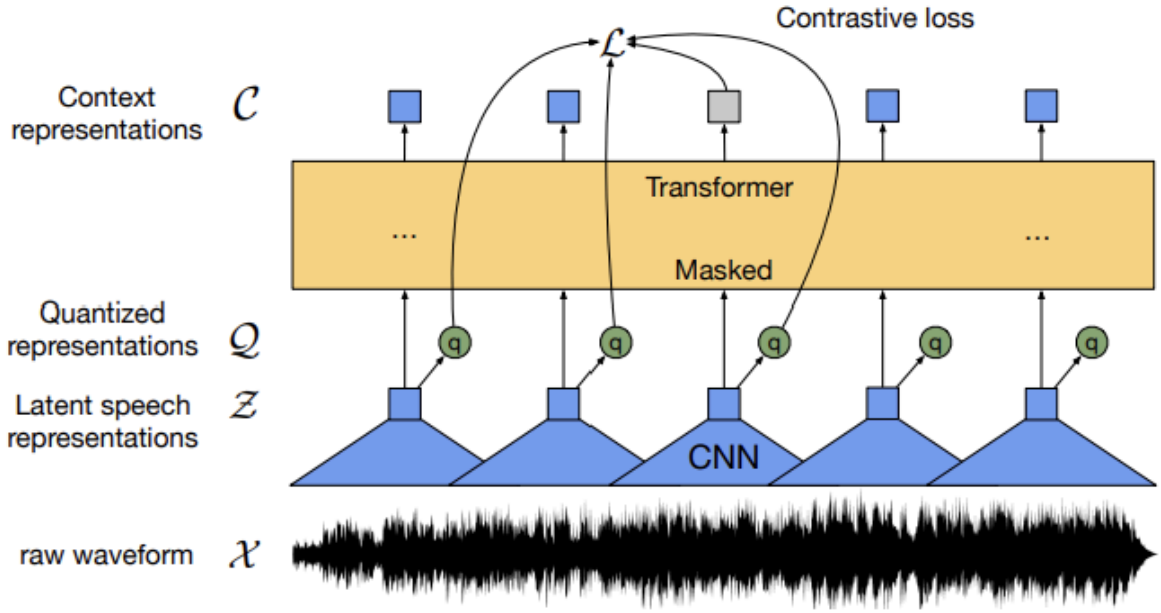


Figure 7. The raw audio waveform is separated into time steps that follow a series of representations from raw waveform X to Context representations C (Alexei Baevski et al., 2020).

The wav2vec framework separates raw audio waveform into a series of discrete samples 25 ms long. These audio samples are represented as X. The audio samples are then passed through a CNN layer. The CNN layer determines speech features based on the frequency spectrogram of each audio unit, and adds the features to each representation, converting them into latent speech representation units Z. The Z units are then passed through a Transformer. The Transformer passes the series of Z units through an encoder / decoder function that outputs the final transcribed context representations C.

Some of the Z units are set aside as quantized representation units Q that are later compared to the transformer model output C. The comparison is used to compute a transcription measure called Contrastive loss L. The Contrastive loss L is used to iteratively refine self-supervised training. The algorithm then predicts the audio transcriptions at further points in time to compare to generated transcriptions and thereby

progressively refine the ASR model.

### 2.5 XLSR - Cross-Lingual Speech Representation

XLSR represents a cross-lingual speech representation ASR framework that incorporates wav2vec version 2. XLSR focuses on creating a multilingual model from raw speech audio combined from several different languages. Models are built from a wide cross section of languages that contain a wide range of phonetic features. The models are initially built in the self-supervised approach of wav2vec. Later a model can be fine-tuned with labeled data for a low resource language not included in the model. Pretrained XLSR models have been created that range between 53 and 128 languages.

## 3. METHODS

Speech audio is typically sampled at a 44 Khz sampling rate. All ASR toolkits employed required the conversion of audio to a 16 Khz sampling rate. Audio clips were shortened as required by the ASR toolkit in use by using Audacity for conversational level clips up to 3 minutes and Elan for utterance level clips up to 10 seconds. Using noise reduction features provided by Audacity did not significantly alter the recognized speech output.

### 3.1 Totonac Corpora

There were corpora for two Totonac varieties, Coatepec (McQuown, 2013) and Upper Necaxa (Beck, 2004), used for ASR transcriptions. In the ASR method using the Meta MMS model (Meta Research, 2024) described below, the model produced results based on the Highland group of Totonac languages having the language code [tos], to which Coatepec belongs as a subset. The language code format stems from an internationally recognized standard designated ISO 639 for representing individual

languages and language groups (*ISO - ISO 639 — Language Code*, 2007). A comparison of the phonological inventories of the three Totonac varieties is shown for consonants in Table 1 and vowels in Table 2.

| | Labial | Alveolar | | Post alveolar | Velar | Uvular | Glottal |
|---|---|---|---|---|---|---|---|
| | | central | lateral | | | | |
| Nasal | m | n | | (ɲ) | (ŋ) | | |
| Plosive | p | t | | | k | q q | ʔ |
| Affricate | | ts | tɬ tɬ | tʃ | | | |
| Fricative | | s | ɬ | ʃ | x | | h h |
| Approximant | | | l<br>(ɾ) ɾ | y | w | | |

Table 1. Consonantal inventory of three Totonac varieties

Legend: black - all 3 varieties, red - Upper Necaxa, blue - Coatepec, purple - Highland

| | Front | | Central | | Back | |
|---|---|---|---|---|---|---|
| | plain | laryngeal | plain | laryngeal | plain | laryngeal |
| High | i iː | ḭ ḭː | | | u uː | ṵ ṵː |
| Mid | | | e eː | ḛ ḛː | o oː | o̰ o̰ː |
| Low | | | a aː | a̰ a̰ː | | |

Table 2. Vowel inventory of three Totonac varieties

Legend: black - all 3 varieties, red - Upper Necaxa, purple - Highland and Upper Necaxa

Exceptions to note among the three varieties are that Upper Necaxa lacks the lateral affricate /tɬ/ and uvular stop /q/ present in both Coatepec and Highland. Coatepec lacks the laryngeal vowels present in Upper Necaxa and Highland. Only Upper Necaxa

has the mid vowels /e/ and /o/. Thus, Coatepec may be considered a subset of Highland, while Upper Necaxa is phonologically distinct from the other two.

### 3.1 Method 1 - ELPIS

ELPIS ASR software was used to provide word level transcription of Highland Totonac utterances. A letter to sound mapping file (Appendix A) was created from transcriptions previously provided for the utterances. Since the utterance line transcriptions appeared polysynthetic in nature, both Line and Prosody annotation levels were used to generate separate sets of word level transcriptions.

To set up ELPIS for ASR transcriptions, follow the instructions on the ELPIS documentation website (*Welcome to the Elpis ASR Documentation! — Elpis 1.0.6 Documentation*, n.d.). To prepare recordings, approximately one minute of audio was split among 12 utterances. This version of Elpis requires the audio files to be in 16kHz WAV format. Recorded audio in 44kHz format was opened in a program like Audacity and exported in 16kHz format. The corresponding .eaf ELAN transcription file was split to match the utterance audio files, thus resulting in 12 audio wav files and 12 .eaf files. Ten of the utterances were used for training and two for testing. After loading the files into the ELPIS website, the user can select which annotation tier to use for recorded transcription. The tier transcriptions should be as clean and time aligned as possible. There is an option to remove words and tags that were entered into the annotation for note taking, but it is better to remove them completely from the annotation in advance if possible.

ELPIS reads the audio and transcription files and produces a Word list it has recognized, along with a frequency for each word. This list is beneficial in perceiving the

relative distribution of words in the transcription. A sample Word list is shown in Table 1 of Appendix A. A letter to sound mapping file manually composed by the user is then uploaded to create what ELPIS calls a Pronunciation dictionary. ELPIS cross checks that the characters uploaded in the letter to sound file match the characters in the transcriptions and flags any characters not found in the mapping file. The corresponding letter to sound mapping file is shown in Table 2 of Appendix A.

In the training phase, there is a selection of the number of n-grams to use for model training. The default is one n-gram which corresponds to learning a single word, but best results were achieved with an n-gram setting of three, which learns a combination of a word along with two of its neighbors.

### 3.2 Method 2 - Meta Research MMS

Pre-trained models from the Massively Multilingual Speech (MMS) project were used to transcribe Totonac varieties of Coatepec and Upper Necaxa using the [tos] Highland Totonac language code. The steps to perform these transcription tasks are recorded in a document called “Totonac Transcription Procedure” stored on github (Allan Plesniarski, 2023).

Note that the Highland Totonac does not match the phonology of the two transcribed Totonac varieties exactly. The orthography produced from the Meta model for Highland Totonac was adjusted to match the orthography originally transcribed for Coatepec and Upper Necaxa where the matching patterns were consistent. The orthography matching patterns are listed in Appendix B.

Other language codes that include the voiceless alveolar lateral fricative /ɬ/ in the Meta MMS model were also tested to extract ASR transcriptions, including [nhi] for

Nahuatl. However, the transcriptions produced with non-Totonac language codes did not reasonably match the Totonac corpora, likely due to word level differences.

### 3.3 Method 3 - XLS-R

A pre-trained XLS-R multi-lingual model Wav2Vec2-XLs-R-300M (Chaumond & Davaadorj, 2023) was fine-tuned to create a custom model for Turkish utterance transcription. Turkish speech was sourced from the "mozilla-foundation/common_voice_16_1" dataset with 123 hours of recorded audio from 1599 speakers available on HuggingFace (Mozilla Foundation, 2024).

The Mozilla dataset contains a number of other low resource languages, including Western Sierra Puebla Nahuatl with the language code [nhi]. Nahuatl is one of the languages that contains lateral fricatives similar to Upper Necaxa Totonac. A fine-tuned model was then created for Nahuatl based on 1 hour of recorded audio from 2 speakers.

## 4. RESULTS

### 4.1 ELPIS Totonac ASR Results

The WER for Upper Necaxa training ranged from 60 to 80%, depending on the choice of 3 or 2 n-gram training models respectively. The model inference error rate after training was nearly 100% in two subsequent inference utterances, with the word "tzamá:" being most commonly recognized.

### 4.2 Meta MMS Totonac ASR Results

Totonac transcription results were generated using the large MMS model with the "tos" Highland Totonac language code. Detailed ASR transcription results for two varieties of Totonac, Coatepec and Upper Necaxa, are shown in Appendix B. The ASR

transcribed Phoneme Error Rate comparison between Coatepec and Upper Necaxa vowels and consonants is shown in Figure 8.

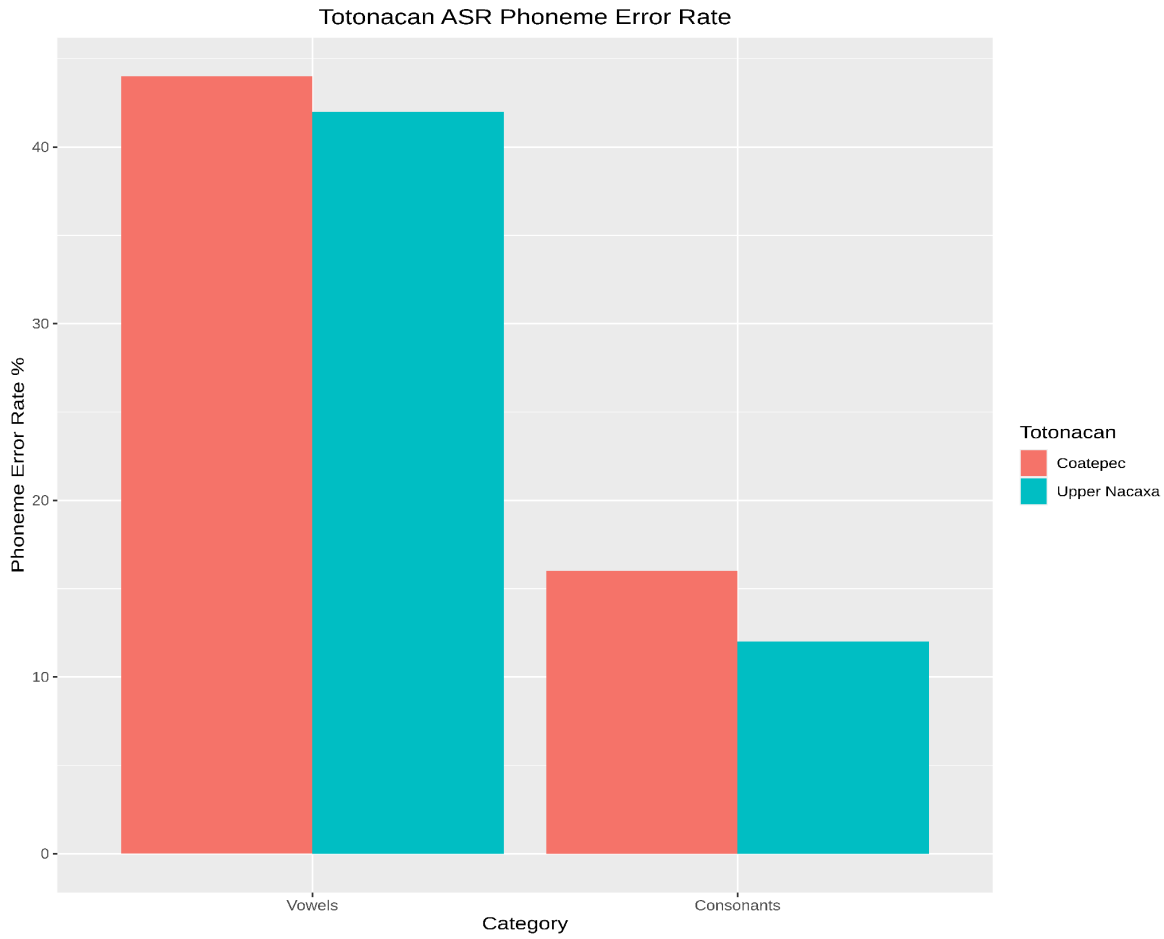


Figure 8. Phoneme Error Rates for Coatepec vs Upper Necaxa

Note that vowels have a higher PER than consonants for both Totonac varieties. Coatepec ASR resulted in a slightly higher PER for both vowels and consonants than ASR for Upper Necaxa. Tables 4 and 5 in Appendix B list both the ASR PER and the Phoneme distribution percentage P% for both varieties. Note there is a difference in the transcribed vowel and consonant inventories between the two varieties, which may be as a result of the corpora used. The overall PER for Coatepec was 29% and the overall PER for Upper Necaxa was 20%.

Example ASR transcriptions for both Totonac varieties are shown in Table 3.

| | **Coatepec** |
|---|---|
| Orig. | na:ɬmíɬ šli:qawhwaá quštaá taqaɬí:n ni:šlakapa:stákaa papi:čanaá |
| ASR | na: ɬ miɬ šli:qajua qošta: taqaɬí:n ni: šlakapa:staka pa: pi: čuná: |
| | **Upper Necaxa** |
| Orig. | a:li:stá:n pus mat puwán tu: tzej tale:ma:nálh tu: tale:ma:nálh nakixkilhnikán? |
| ASR | ali:stá:n pus matpuwaán tu: tze: talemanal tu: talemana naixquilhnikán |

Table 3. Example Original vs. ASR transcriptions for Coatepec and Upper Necaxa

Note that the consonant ASR transcription matches reasonably inline with the original despite the word boundaries being somewhat offset.

### 4.3 XLSR

The WER for Turkish ranged from 69% to 32% during training. The training data volume for Nahuatl was not sufficient to determine a WER for training.

## 5. DISCUSSION & CONCLUSIONS

### 5.1 Discussion

Although ELPIS encapsulates lower level ASR functions to make the system more user accessible, it can be challenging to set up a suitable computer runtime environment to run ELPIS on Docker or ELPIS on a Cloud platform. The advantage of using ELPIS for Low Resource Language ASR is that it helps a linguistic researcher to quickly familiarize themselves with the word inventory frequency and letter to sound mappings of available audio recordings. The challenge in using ELPIS is supplying a sufficient

volume of audio recordings and transcription files in order to obtain a reasonable Word Error Rate.

The Meta MMS ASR transcription method produces a relatively low Phoneme Error Rate for the Totonac varieties transcribed. Although the Highland Totonac variety previously trained into the MMS model by Meta does not match the phoneme and word inventory of the Totonac varieties tested exactly, it is sufficient to produce results that show differences between the two varieties. Differences between the vowel and consonant inventories between the two varieties could be determined. The Upper Necaxa variety had a greater vowel phoneme inventory than the Coatepec variety. The overall PER for Upper Necaxa was lower than Coatepec, with Consonants providing the bulk of the difference by having a lower PER in Upper Necaxa. There was also a greater percentage of Upper Necaxa consonants in the corpus as compared to vowels, with nearly a 3:1 ratio. The Coatepec ratio of consonant to vowel phonemes was nearly 1:1.

Despite the PER for the Meta MMS ASR method being relatively low, the WER for this method was significantly higher. This is because there is a greater difference in Highland Totonac word inventory trained into the Meta MMS model as compared to the Coatepec and Upper Necaxa utterances tested. Using Beyond Compare software to compare the annotated words to the ASR word output, the WER of the Meta MMS ASR method was estimated to be 50% or more depending on the transcribed word length. The longer the transcribed word length, the greater the chance of an ASR transcription error.

The XLSR results produced high WER rates at inference time. The results were affected by lack of training and testing data available of the low resource language for fine tuning of the multilingual language model used. Further model fine tuning would produce inference results with a lower WER.

### 5.2 Conclusions

For linguistic researchers with audio recordings and matching annotated transcription files, ELPIS is an excellent tool to understand word and phoneme inventory and frequency. ELPIS may initially produce inaccurate results for word level transcription, considering that each time ELPIS is instantiated, the model needs to be learned from scratch.

The ASR method with the highest accuracy for producing transcriptions of Totonac languages was Meta MMS. Linguistic researchers need only to find the language code already trained into available MMS models that most closely matches the low resource language that they are transcribing. The transcription output provides a substantially correct base from which to manually adjust the transcriptions, as opposed solely to manual transcription by listening to the audio. There may also be auditory biases that may skew what someone perceives as hearing when transcribing audio manually. Employing an ASR transcription method eliminates any perceptual biases. Since the transcription accuracy of consonants is higher than vowels, the consonants give a transcribed framework from which the embedded vowels can be adjusted to match the target language phonological inventory. The consonant transcription can also be used to manually adjust word boundaries to match the target language morphology.

XLSR holds promise in producing lower WER with the availability of a sufficient volume of audio and annotated transcriptions for model fine-tuning. Over time larger XLSR models may be released incorporating more low resource languages. Further research could focus on fine tuning the latest multi-lingual XLSR model with a specific Totonac language. Integrating a distinct Totonac language that is mutually unintelligible from other Totonac languages already trained into existing models would in theory

produce lower ASR Phoneme and Word error recognition rates.

**REFERENCES**

Alexei Baevski, Zhou, Y., Mohamed, A., & Auli, M. (2020). *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. *33*, 12449–12460.

Allan Plesniarski. (2023, December 6). *LING402-ASR*. GitHub. https://github.com/plesniar/LING402-ASR

Beck, D. (2004). *Upper Necaxa Totonac*. Munich: LINCOM Europa.

Chaumond, J., & Davaadorj, M. (2023, October 3). *blog/fine-tune-xlsr-wav2vec2.md at main · huggingface/blog*. GitHub. https://github.com/huggingface/blog/blob/main/fine-tune-xlsr-wav2vec2.md

Foley, B., Arnold, J., Coto‐Solano, R., Gautier Durantin, T. Mark Ellison, Daan van Esch, Heath, S., František Kratochvíl, Zara Maxwell-Smith, Nash, D. B., Olsson, O., Richards, M., San, N., Stoakes, H., Thieberger, N., & Wiles, J. (2018). Building Speech Recognition Systems for Language Documentation: The CoEDL Endangered Language Pipeline and Inference System (ELPIS). *Minerva Access (University of Melbourne)*. https://doi.org/10.21437/sltu.2018-43

*ISO - ISO 639 — Language code*. (2007). ISO. https://www.iso.org/cms/%20render/live/en/sites/isoorg/home/standards/popular-standards/iso-639-language-code.html

Macaire, C. (2021). *Recognizing lexical units in low-resource language contexts with supervised and unsupervised neural networks*. HAL Archives Ouvertes. https://hal.science/hal-03429051

Meta Research. (2024, January 24). *fairseq/examples/mms at main · facebookresearch/fairseq*. GitHub. https://github.com/facebookresearch/fairseq/tree/main/examples/mms#asr

McQuown, Norman A. and Manuel Oropeza Castro. [1949] 2013. Audio of the recording of the Textos Totonacos by Manuel Oropeza Castro. [Digitalization by the Linguistic Laboratory at the University of Chicago.]

McQuown, Norman A. 1971. [1939–1968]. Textos totonacos. Microfilm Collection of Manuscripts on Cultural Anthropology 100. Joseph Regenstein Library, The University of Chicago.

Mozilla Foundation. (2024, January 16). *mozilla-foundation/common_voice_16_1 · Datasets at Hugging Face*. Huggingface.co. https://huggingface.co/datasets/mozilla-foundation/common_voice_16_1

Scharenborg, O. E., Ciannella, F., Palaskar, S., Black, A., Metze, F., Ondel, L., & Hasegawa-Johnson, M. (2017). Building an ASR system for a low-resource language through the adaptation of a high-resource language ASR system. Preliminary results [in press]. *Proceedings of the International Conference on Natural Language, Signal and Speech Processing*. https://repository.ubn.ru.nl/handle/2066/244342

Tjandra, A., Singhal, N., Zhang, D., Kalinli, O., Mohamed, A., Le, D., & Seltzer, M. L. (2022). *Massively Multilingual ASR on 70 Languages: Tokenization, Architecture, and Generalization Capabilities*. arXiv. https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.2211.05756&site=edslive&scope=site

Wang, C., Tang, Y., Ma, X., Wu, A., Dmytro Okhonko, & Juan Miguel Pino. (2020). fairseq S2T: Fast Speech-to-Text Modeling with fairseq. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2010.05171

Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Soplin, N. E. Y.,

Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., & Ochiai, T. (2018). *ESPNet: End-to-end speech processing toolkit*. Waseda.pure.elsevier.com. https://doi.org/10.21437/Interspeech.2018-1456

*Welcome to the Elpis ASR documentation! — Elpis 1.0.6 documentation*. (n.d.). Elpis.readthedocs.io. https://elpis.readthedocs.io/en/latest/

# APPENDIX A

**Table 4. ELPIS Word frequency in sample Totonac Training Files**

| Word | Frequency | Word | Frequency |
|---|---|---|---|
| mat | 5 | chú | 1 |
| chi | 4 | cuento | 1 |
| ka | 4 | hka | 1 |
| lhtzá | 4 | hnina | 1 |
| cha | 3 | ix | 1 |
| he | 3 | ixá | 1 |
| kí | 3 | kawa | 1 |
| má | 3 | kwan | 1 |
| tu | 3 | lakaní | 1 |
| tzamá | 3 | munka | 1 |
| wi | 3 | n | 1 |
| xla | 3 | naka | 1 |
| a | 2 | nama | 1 |
| j | 2 | ne | 1 |
| mpalá | 2 | skujá | 1 |
| palaho | 2 | skúja | 1 |
| sájna | 2 | takúxtu | 1 |

| | | | |
|---|---|---|---|
| tzi | 2 | tanka | 1 |
| u | 2 | ti | 1 |
| xcha | 2 | tín | 1 |
| ya | 2 | wa | 1 |
| aa | 1 | xkú | 1 |
| akxní | 1 | xlho | 1 |
| chi'xkú | 1 | xni | 1 |
| chu | 1 | | |

**Table 5. ELPIS Letter to Sound Mapping file for Totonac Training Files**

| Consonants | Vowels | |
|---|---|---|
| ch t͡ɕ | aa aa | u◌́ U |
| c c | | u◌́ U |
| tz t͡s | e: e: | ú U |
| h h | a: a: | |
| kk k | i: i: | i◌́: I: |
| k k | o: o: | i◌́ I |
| m m | u: u: | í: I: |
| nn nn | | í I |
| n n | a`: A: | |
| t t | á: A: | a a |
| x x | | e e |
| l l | u◌́: U: | i i |
| w w | ú: U: | o o |
| j J | | u u |
| s s | a◌́ A | |
| y j | á A | |
| p p | | |
| w w | | |

# APPENDIX B

**Table 6. Meta MMS Highland Totonac Orthography Mapping**

| | Corpora Match | MMS Model Output |
|---|---|---|
| **Vowels** | a: | a̲ |
| | á: | á̲ |
| | e: | e̲ |
| | é: | é̲ |
| | i: | i̲ |
| | í: | í̲ |
| | o: | o̲ |
| | ó: | ó̲ |
| | u: | u̲ |
| | ú: | ú̲ |
| | | |
| **Coatepec** | č | ch |
| | ł | lh |
| | š | x |
| | | |
| **Upper Necaxa** | wa | hu |

# APPENDIX C

**Table 7: Coatepec Totonac Transcription Phoneme Error Rate**

| Phoneme | PER | Phoneme% |
|---|---|---|
| a | 26% | 14% |
| á | 77% | 8% |
| a: | 0% | 2% |
| i | 38% | 5% |
| í | 60% | 4% |
| i: | 14% | 5% |
| u | 67% | 4% |
| ú | 86% | 2% |
| u: | 33% | 2% |
| **Vowels** | 44% | 45% |
| | | |
| y | 0% | 1% |
| w | 50% | 6% |
| ʔ | 100% | 1% |
| č | 0% | 4% |
| l | 0% | 3% |
| p | 0% | 5% |
| t | 0% | 7% |
| q | 0% | 3% |
| s | 0% | 2% |
| š | 0% | 5% |
| k | 25% | 4% |
| ɬ | 0% | 1% |
| m | 14% | 2% |
| n | 17% | 8% |
| h | 100% | 1% |
| **Consonants** | 16% | 55% |

| **Total** | 29% | 100% |
|---|---|---|

**Table 8: Upper Necaxa Totonac Transcription Phoneme Error Rate**

| **Phoneme** | **PER** | **Phoneme%** |
|---|---|---|
| a | 43% | 12% |
| á | 0% | 0% |
| á: | 100% | 0% |
| a: | 50% | 2% |
| e | 75% | 2% |
| i | 0% | 4% |
| í | 100% | 0% |
| i: | 33% | 2% |
| o | 0% | 0% |
| o: | 100% | 0% |
| u | 83% | 2% |
| ú | 0% | 0% |
| u: | 0% | 0% |
| **Vowels** | 42% | 27% |
| | | |
| j | 20% | 2% |
| w | 20% | 2% |
| x | 10% | 9% |
| l | 5% | 9% |
| p | 33% | 2% |
| t | 4% | 10% |
| r | 0% | 0% |
| s | 25% | 2% |
| k | 31% | 7% |
| m | 0% | 9% |
| n | 8% | 10% |
| h | 26% | 8% |

| | | |
|---|---|---|
| z | 0% | 4% |
| c | 100% | 1% |
| tz | 100% | 2% |
| **Consonants** | 12% | 73% |
| **Total** | 20% | 100% |