Field NLP - Automatic Speech Recognition of Low-Resource Languages Based on Chukchi

Cydnie Davenport (sdevenport@edu.hse.ru), Tatiana Udina (tayudina_2@edu.hse.ru), Emil Nadimanov (einadimanov@edu.hse.ru), Anastasia Safonova (aasafonova@edu.hse.ru)

National Research University Higher School of Economics, July 2021

1. INTRODUCTION

The following paper presents a project focused on the research and creation of a new Automatic Speech Recognition (ASR) and Text to Speech (TTS) system based in the Chukchi language. The aim of this is to develop a system that makes the language more accessible to speakers of Chukchi - such as annotating subtitles on videos and movies, providing more accessible data for research and analysis, or the creation of chat-bots for online users. This system should consist of; an acoustic model for receiving an audio signal fragment and which gives the probability of various phonemes based on the fragment analyzed; a language model for determining which suggestions are more or less likely; and a decoder which will determine the most likely prediction. Predictive automatic speech recognition models already exist and are a popular focus in the realm of Natural Language Processing, however, the most challenging adversaries are low-resource languages due to extreme data deficits.

This project is centered around a multi-step research process. Initially, we began by analyzing the Chukchi language from a linguistic perspective, but for the sake of clarity regarding motivations for making this system, it also must be looked at from a cultural and sociolinguistic perspective. What is known about the language, are there any cultural influences within the language, why is such a system necessary, and so on. Once there was extensive understanding of the subject at hand, the next step would be finding data that is usable. For this project, this included broadcasts from a Russian-based Chukchi radio station, videos and lessons from YouTube, written translations of the Bible, and the Higher School of Economics' set of Chukchi-based corpora known as Chuklang. Once enough data is collected, there then comes the task of cleaning it. This included labeling and

segmenting audio data for training, cleaning and filtering out unnecessary symbols (mainly Russian) from text, and determining which data would be used for pre-training and which would be used for testing the resultant model. Once enough data has been collected *and* cleaned for our model, we sample and train various models to understand how they process data. Additionally, we must try various encoders to understand how well they clean out noise and extra acoustic audio. Extra research must be conducted in order to compare models designed for both high- and low-resource languages. Various designs and tools for training ASR models include VQ-VAE, XLSR, the toolkit Kaldi, wav2vec, and more. The intended result of this project is an automatic speech-recognition system that can seamlessly work with Chukchi and provides us with the potential to be used for other low-resource languages.

2. BACKGROUND

2.1 The Chukchi Language

The Chukotko-Kamchatkan family of languages is said to contain two branches by default. The northern branch is referred to as the Chukotian branch (or “Luorevetlan”, based on the Chukchi ethnonym) and consists of Chukchi, Koryak, Alutor and Kerek (now extinct). The second branch is known as Itelmen, and contains the language Western Itelmen, which itself consists of two dialects: Khajrjusovo and Sedanka (Fortescue, 2005). The language of focus for this paper is Chukchi, a polysynthetic language spoken primarily within the Chukotka Autonomous Okrug, which is located in the easternmost part of Siberia. Chukchi itself is an endangered indigenous language with less than 10,000 speakers at present, and most speakers are bilingual with a primary language of Russian. There are only less than 100 speakers who don’t speak Russian at all. Instances and usages of this language are difficult to come by, and is not a language taught in schools. The decreasing use of this language in general everyday life, as well prominence of Russian within the community demonstrates the necessity for an automatic speech recognition system, so that we may provide more accessibility to such an endangered and very low-resource language and its community.

2.2 What is a Low-Resource Language?

In the field of NLP, research tends to have a large focus on languages where data and native speakers are easily accessible, and the language is relatively well-

known. These are referred to as high-resource languages, and as such, produce a large quantity of data. On the other hand, low-resource languages (occasionally referred to as LRLs) are usually "..less studied, resource scarce, less computerized, less privileged, less commonly taught, or low density.." (Magueresse, et al. 2020) and therefore are not prioritized in the realm of NLP research. However, this is actually one of the more major motivating factors for our project. Chukchi is an incredibly low-resource language, an example of which is that most of the up-to-date information regarding the language and its speakers is most easily accessed from a detailed article found on Wikipedia[1]. The low-resourcedness of Chukchi is what inspired this project, as it is an endangered language, and one that is not particularly accessible in terms of media, education, and history. By creating a new automatic speech recognition system, not only can accessibility be provided for this language, but it also creates new opportunities for the same achievements in other low-resource languages.

2.3. What is an ASR System?

Traditionally, modern automatic speech recognition systems are typically made up of three different parts: a lexicon, an acoustic model, and a language model.[2] The lexicon contains the information that an ASR system needs to be able to understand the input it receives on the base level. This includes things such as phonetic transcription codes that are used for the target language's phonemes. For English, ARPABET[3] and TIMIT[4] are the most commonly used codes and transcriptions, developed by the Defense Advanced Research Projects Agency (DARPA).

The second component of an ASR system is the acoustic model, which is responsible for forging the relationships between the phonemes of a language (such as the ones provided in the lexicon) and an audio signal. This interaction is

[1] Link to the article in question: https://en.wikipedia.org/wiki/Chukchi_language#:~:text=Chukchi%20%2F%CB%88t%CA%83%CA%8Ak,mainly%20in%20Chukotka%20Autonomous%20Okrug.&text=In%20the%20UNESCO%20Red%20Book,the%20list%20of%20endangered%20languages.

[2] Information about the basic components of an automatic speech recognition system is widely available, one of the more easily understandable sources can be found here: https://voximplant.com/blog/what-is-automatic-speech-recognition

[3] https://en.wikipedia.org/wiki/ARPABET

[4] https://en.wikipedia.org/wiki/TIMIT

supported by the use of transcripts along with their respective audio files,[5] and are thus supposed to be able to map statistical representations for feature vector sequences of a particular phoneme (or sound unit) and classify it (Sarma and Sarma, 2015). This allows the system to recognize and distinguish this particular sound unit from the rest of the phonemes that it may encounter in both training data and experimental data.

Finally, there is the language model, which helps to provide clearer contexts and allows the model to view the language in a naturally occurring form. Thus, this is where training comes in. By training the language model, contexts become more comprehensive and coherent when interacting with the system, and are thus understandable. By design, the system, with the help of all of these aforementioned components, is then supposed to be able to predict speech patterns.

2.4. Previous Research

Regarding previous research focused on low-resource languages and ASR, there have been multiple approaches to finding the most efficient and effective model for processing such a limited amount of available data. The basic framework for processing speech was typically based on a few components. For example, these components could have included an autoencoder (denoising or otherwise), dual transformation for both text and speech, bidirectional sequence modeling, typically with a major focus on unsupervised pre-training (Ren, et al., 2020). In addition to this, many approaches also included a Transformer-based unified model structure (Ren, et al., 2020, Baevksi, et al., 2020, Kriman, et al., 2019). The goal was to have a system that could sample the language evenly and return feedback to the model, learning as it continued to sample more data (Yubei, et al. 2021). These components are crucial to the creation of our model, and will be utilized in this project.

Both universities and major corporations alike (e.g., Google with Strope et al, 2011) have also researched the most effective ways to implement the most ideal features for training both acoustic models as well as language models. In many cases, it is incredibly difficult to create a pre-training environment that is entirely unsupervised, but the key here is that it is *almost* unsupervised. The benefit of

[5] Microsoft conducts extensive research regarding acoustic models, more information as well as links to other sources and publications can be found here: https://www.microsoft.com/en-us/research/project/acoustic-modeling/

unsupervised data pre-training is that it makes data much more usable. Without the need for supervision, the amount of usable data increases significantly, which gives us much more accessibility to languages that lack a significant amount of data (i.e., being able to utilize data in a more efficient way). The key to much of the unsupervised training that already exists is the technique implemented. Discriminative (Strope et al, 2011), in which a dual unigram and trigram language model was used to interpret relative truth, active or passive (Riccardi, Hakkani-Tür, 2005). Passive learning was a technique and algorithm that dominated the realm of automatic speech recognition for much of its lifespan.

Passive learning was the initial algorithm used for training language models. This meant that a model was trained based on a single implementation of a set of data, fixed in time (Riccardi, Hakkani-Tür, 2005). As a result, there was no room allowed for the model to improve. Additionally, all the data being used was usually transcribed under human supervision, and training a model to work with language data ended up being a very time consuming process. By taking the workload off of the researchers and volunteers who transcribe this data and manually check the model, more effective and efficient means of training an ASR system can be developed. Active learning mechanisms, as a result, are particularly useful in cases like these. With a feedback system that allows for the model to learn from itself and ultimately use less data. This can prove invaluable in developing an ASR system for Chukchi and other low-resource languages.

3. DATA COLLECTION

Given that Chukchi is a very low-resource language with very few speakers, finding usable data proved difficult, as was discussed above. Samples of both spoken and written Chukchi were selected from any source that could be found. This included the Charles Weinstein website of Chukchi with translations and descriptions in both French and Russian, recent news broadcasts in Chukchi (December 2020 and January 2021) from the Anadyr'-based radio station *Radio "Purga",* videos from Youtube, corpora from Chuklang.ru, as well as translated parts of the Bible from Bible.is.

3.1 Radio 'Purga'

One of the main sources of high-quality annotated data was the Chukchi radio station Radio “Purga,” which has a special feature at their station in which they report news in Chukchi on a regular (almost daily) basis. A representative of this radio station provided our research team with a total of 2.53 hours of audio data from 30 episodes of morning news. These audio files then had to be manually split into shorter chunks of both audio recording and text pairs in order to be used further. Fortunately, each broadcast came with its own script. However, there were a few issues with this data in that the real recording and the script would sometimes differ. Additionally, the script also contained several sentences of pure Russian speech, which makes some part of the audio file unusable. Both of these issues were solved by excluding such recording-script pairs from the dataset.

3.2 YouTube Videos

Youtube was another primary resource to find any instances of Chukchi audio samples. All videos found were then converted into WAV format. This portion of the corpus contains:

- stories;
- Chukchi online language lessons;
- interviews with native speakers;
- lessons from the project “Vetgav. Chuckhi lessons”;
- cartoons;
- news;
- and more.

All links to the video data used for this project and statistical sources can be found at the project’s GitHub[6]. The video data of the corpus totaled at 14 hours, 12 minutes and 13 seconds.

3.3 Bible Audio

The resource Bible.is[7] contains chapters from the Bible in a variety of languages from around the world, including Chukchi. There we found audio recordings totaled at 3 hours, 36 minutes and 52 seconds in wav format. Some Bible chapters do have a text annotation (for example, *The Gospel of Luke*), and some don’t (*The Book of Jonah*).

[6] Project GitHub: https://github.com/ftyers/fieldasr/blob/main/DATA.md
[7] Bible.is: https://www.faithcomesbyhearing.com/audio-bible-resources/bible-is

3.4 Chuklang Corpora

The Chuklang Corpora[8] was created by professors and students of the National Research University - Higher School of Economics in Moscow, Russia, during the linguistic expeditions to the village of Amguema in the Iultinsky District of the Chukotka Autonomous Okrug. It consists of annotated audio recordings with a total length of 1 hour, 14 minutes and 18 seconds.

3.5 All audio data

Table 1 displays information about the duration of all audio data, broken down by resource type.

Table 1. Distribution of audio

Resource	Duration	Transcription availability
Radio 'Purga'	2:32:00	+
Youtube	14:11:13	-
Bible	3:36:52	-
Chuklang	1:14:18	+

Duration of all unannotated data - 17:48:05
Duration of all annotated data - 3:46:18
Total duration of all audio data - 21:34:23

3.6 Text

For a linguistic model in an automatic speech recognition system, any kind of texts are useful, even if they do not have their own audio annotations. The most commonly used lexical items (*ынкъам, гивик, ымы, ытлён, ынан, Чукоткакэн, гатвален, ынқэн, лыги, вальыт, вагыргыттитэ, гатваленат*) were used to assist in finding written examples of Chukchi. In addition to this, Yandex.XML[9], which is a service that allows a user to send a search query to the Yandex search engine and receive any answers found in XML format, was also used. No more than 200

[8] Chuklang corpora: https://chuklang.ru
[9] Yandex.XML: https://yandex.ru/dev/xml/

results per query were allowed per its own limitations, however, by using various filters and sorting mechanisms, 1800 URLs were found and 462 unique links were extracted. From these selected URLs, the largest and most useful sources for parsing were determined to be:

- news outlets
- stories and riddles
- fictional literature
- grammar and thematic dictionary of the Chukchi language, a collection of Chukchi literary texts

The full text corpus is composed of 112,719 sentences, and 2,068,273 words. The obtained URL links and text corpus can be found on the GitHub repository for this project[10]. Table 2 below displays the distribution of texts according to subcorpus. From these numbers, the necessity for automatic deletion of Russian from a few sub corpora becomes quite noticeable.

Table 2. Distribution of texts by subcorpus (before the removal of the Russian language)

Subcorpus	Number of texts/pages	Number of sentences	Number of words	% of words	The presence of the Russian language
Internet newspaper *"Extreme North"*	118	6187	82661	4,00%	+
A special supplement to the newspaper *"Extreme North"*	11	303	27466	1,33%	+
"Portal of National Literatures"	4	569	4729	0,23%	-
The book *"По аргишному пути канчаланского чаучу"*	1	444	3936	0,19%	+
Lingvoforum: fairy tales, stories	8	251	1297	0,06%	-
VK: fairy tales	11	573	2840	0,14%	-
Puzzles	1	1234	8112	0,39%	+

[10] Project GitHub: https://github.com/ftyers/fieldasr/blob/main/DATA.md

The Wayback Machine	47	3079	16952	0,82%	-
Charles Weinstein	10	96765	1891705	91,46%	+
Radio *'Purga'*	29	836	7742	0,37%	+
Chuklang Corpora	1006	1006	4414	0,21%	-
Bible.is	25	1472	16419	0,79%	-
All data	**1271**	**112719**	**2068273**	**100%**	

4. DATA PROCESSING

4.1 Preprocessing: Text

The Chukchi alphabet contains a special symbol (‘) which was implemented in different forms in writing (‘, ’, `, “, ”, ' , etc) across different sources of Chukchi. Therefore, the first step for preprocessing was to identify all instances of this apostrophe and determine a single variant for its representation, with the result being: (’). Following this were substitutions of letters *к’ - ӄ, К’- Ӄ, н’ - с, Н’ - Ӈ, ль - ӆь, Ль - Ӆь*, where the rightmost form is the final representation needed.

Additionally, we cleaned up the data collected from internet-sources on a more global scale: that is, deletion of hyperlinks, illegible symbols, email addresses (these can frequently be found in news articles), and occasionally, double spaces between words. All data preprocessing was carried out with the help of the standard Python library *re*[11], which is used for work with regular expressions.

4.2 Removing Russian from the Text Corpus

Due to the writing system of both Chukchi and Russian being incredibly similar, it became necessary to clean the Russian out of the text corpus. This is because the similarities between the visible forms of the languages can be problematic to the model, and thus skew our results. Binary sentence classification with the help of the most frequently used words in both Russian and Chukchi. Based on the collected text data that only contains Chukchi, we calculated the most frequently occurring words. The “New Russian Lexical Frequency Dictionary”[12] was used to create a list of the

[11] Documentation on the Python library *re* can be found here: https://docs.python.org/3/library/re.html
[12] Новый частотный словарь русской лексики: http://dict.ruslang.ru/freq.php

most frequently occurring Russian words. In order to predict which class a sentence belonged to, each word in the text, along with its normal form determined by the morphological analyzer *pymorphy2*[13], were checked for entries in the frequency list. The corresponding variable-counters were increased and then compared. If a Russian word was found more than a Chukchi word, then that sentence was considered Russian, and vice versa.

The pre-training model "lid.176.bin" was used for fastText[14] for language recognition. This model is supported for 176 languages. With the assistance of this model, the likelihoods of sentences belonging to certain languages were isolated, followed by the use of the model KMeans [15]for segmenting data into 2 clusters: Chukchi and not-Chukchi. For the given model the f1-score value was 0.91. After deletion of Russian from the corpus, the size of the resulting corpus was 15309 sentences and 117567 words. This corpus can be found on the GitHub repository for this project[16].

Table 3. Distribution of texts by subcorpus (after the removal of the Russian language)

Subcorpus	Number of texts/pages	Number of sentences	Number of words	% of words
Internet newspaper *"Extreme North"*	118	3063	33219	28,26%
A special supplement to the newspaper *"Extreme North"*	11	3236	25160	21,40%
"Portal of National Literatures"	4	574	4732	4,02%
The book *"The Argysh route of a Canchalan Chukchi"*	1	208	1425	1,21%
Lingvoforum: fairy tales, stories	8	304	1297	1,10%

[13] Documentation for *pymorphy2:* https://pymorphy2.readthedocs.io/en/stable/
[14] Documentation for fastText: https://fasttext.cc/
[15] Documentation for KMeans: https://pymorphy2.readthedocs.io/en/stable/
[16] Project GitHub: https://github.com/ftyers/fieldasr/blob/main/data/text_corpus.txt

VK[17]: fairy tales	11	755	2840	2,42%
Puzzles	1	358	1511	1,29%
The Wayback Machine	47	3127	16904	14,38%
Charles Weinstein, chukchi lessons	10	332	1989	1,69%
Radio *'Purga'*	29	863	7654	6,51%
Chuklang Corpora	1006	1006	4416	3,76%
Bible	25	1483	16420	13,97%
All data	**1271**	**15309**	**117567**	**100%**

4.3 Attempts at Denoising

Due to Chukchi being a very low-resource language, the amount of high quality audio recordings in our dataset were minimal. Most of the audio that dominated the collected data were those of fairly poor quality. In order to achieve better results with the data on hand, a denoising autoencoder was applied to the data to filter out extra noise.

However, either due to flaws in denoising methodology or due to the high level of noise pollution in the collected audio data, the application of a denoising autoencoder was fruitless, and did not appear to aid our system. In the future, we would like to find something that will clean our audio in a more efficient and effective way.

4.4 Segmenting and Labeling Radio Data

The samples from Radio "Purga" were collected for the months of December 2020 and January 2021. These files typically consisted of a short news segment, usually no longer than 5 minutes each. With about 15 of these samples collected for each month, we had 200 minutes of audio. However, these audio files needed to be segmented, as it was unfortunately not all Chukchi. These audio recordings were segmented by sentences, and labeled using the software ELAN[18]. Any instances of

[17] Vk.com is a Russian social networking site.
[18] ELAN is a linguistic annotator that allows for work with audio files, including segmentation, and labeling.

Russian were to be extracted and removed from the files. In addition to this, the audio broadcasts were also provided with personal scripts from the speaker's notes. As a result, we were able to take these scripts, and segment the audio by sentences, which was both necessary for processing data and training the model. However, some of these scripts appeared to be used more as notes as opposed to actual scripts, with some portions of the broadcast entirely missing from the documents we had been provided with. In these cases, the audio was not usable, and was removed from the dataset.

4.5 Audio Data Slicing

All received audio was cut into small fragments. Sometimes it was possible to cut by pauses, so as not to cut off parts of words that may be important for ASR model training. In this case, the Python libraries *pyAudioAnalysis*[19] and *pydub*[20] were used. Sometimes it was impossible to cut by pauses (for example, if there was music playing in the background), if this was the case, then the files were cut by 5 minutes.

5. METHODOLOGY

This project was centered around 3 experiments with 3 different models: VQ-VAE, XLSR, and wav2vec. These models were selected specifically for their previously displayed use for low-resource languages. Each model was pre-trained with a portion of the cleaned, segmented, annotated and unannotated Chukchi data, and then tested with the remaining portion.

5.1 VQ-VAE

5.1.1 Theory

One approach used in creation of ASR systems for low-resource languages is to use a model that can learn from latent representations in language. For this task the best option would be to use VQ-VAE (Oord, Vinyals, et al. 2017). This model has two important components: VAE (Variational Auto-Encoders) (Kingma and Welling, 2013) and VQ (Vector Quantized). VAE is a type of NN architecture which does inference by retrieving statistics (mean and variance) for latent random variables. Model learns

[19] https://github.com/tyiannak/pyAudioAnalysis
[20] https://github.com/jiaaro/pydub

separate $\mu_i$and $\sigma_i$for a random variable. VAE works for continuous data, a вот VQ-VAE learns discrete representations (embeddings). It is assumed that the model VQ-VAE will learn embeddings by constructing its own inputs. Therefore, only audio data is required for learning. Thus, if the learned latent representations are learned well, then they can be used for unsupervised or semi-supervised models.

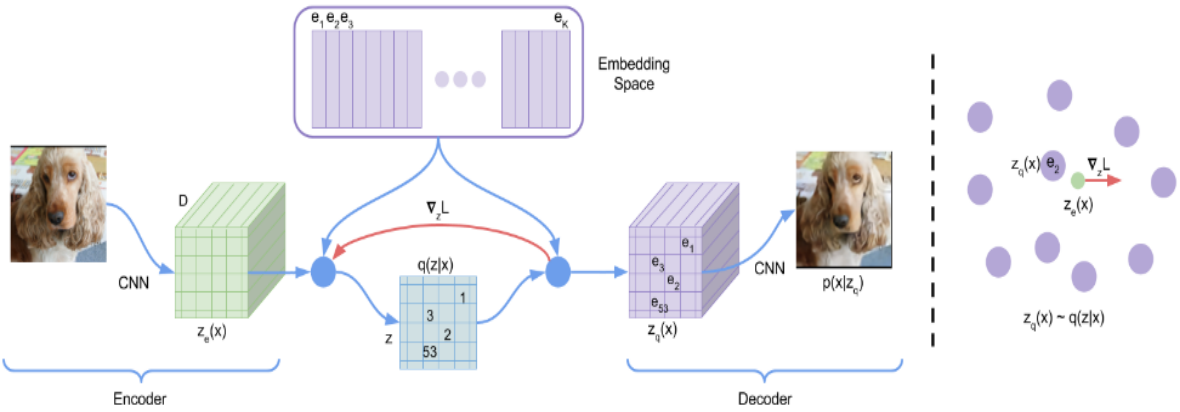

Figure 1. Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. (Oord, Vinyals, et al. 2017)

5.1.2 Experiment

Two experiments were carried out for the collected audio data: Chuklang, Bible.is, Radio “Purga,” and YouTube. The data corpus used in the first experiment contained audio with a length of 4-5 minutes (as a result a large part of this audio was cut down to 5-20 seconds), therefore a decision was made to conduct a second experiment in which the audio that had lengths of longer than 20 seconds were converted into smaller audio fragments, segmented by pauses. This was done as the length of the largest piece of audio in the dataset influences the padding parameter value of the model: the model can have a significant bias in the data, as it will fill all the audio-recording vectors with zeros to fit the size of the largest audio piece.

The second experiment that was carried out focused on the use of data augmentation. We attempted to apply various effects from the first experiment to the audiodata, thereby doubling the size of the training dataset. Regarding effects, the following were used:

- single-pole lowpass filter (["lowpass", "-1", "300"]),
- reduce the speed (["speed", "0.8"]),

- after that it was necessary to add the `rate` effect with the original sample rate (["rate", f"{sample_rate1}"]),
- reverberation (["reverb", "-w"]).

In the present work we used the model VQ-VAE for *PyTorch.* The following parameters were used for the VQ-VAE model:

- input_dim=39,
- hid_dim=256,
- enc_dim=64,
- K=512.

The VQ-VAE model was trained for 1000 epochs. The following values were selected as the batch-size for the data: 128 (train), 10 (validation), 16 (test). *PyTorch's* Adam Optimizer was also implemented with a learning rate of $2e^{-4}$.

5.1.3 Results

The validation loss function was used in order to rate the quality of the VQ-VAE model. The results from the change in the function can be observed below.

Table 4. Validation loss values for experiments with VQ-VAE

Experiment	Validation loss (1 epoch)	Validation loss (1000 epoch)
Experiment 1.1 (without augmentation)	3249.4058	3132.565
Experiment 1.2 (with augmentation)	3247.8926	3131.7913
Experiment 2	3249.5176	3128.0374

It is visible from this table that even after 1000 epochs, the value of the validation loss was left practically unchanged. As a result, we have determined that this model was unsuccessful for our research purposes due to 2 main reasons: the small amount of data available for training, and the poor sound-quality of these audio files. As mentioned previously, the denoising autoencoder proved ineffective, resulting in audio that still contained a significant amount of noise, likely interfering with the model.

5.2 XLSR

5.2.1 Theory

Another approach to this task is using a model trained on a high-resource language and fine-tuning it via low-resource language data, in our case, Chukchi. A model that is suitable for such a cross-language approach is referred to as XLSR or *Cross-Lingual Speech Representations*. XLSR is a transformer-based multilingual model that was trained on 56 thousand hours of speech data shared between 53 languages. It is supposed to represent latent features that are shared between languages. This means that XLSR can be used as a solid base for fine-tuning for low-resource languages. Therefore, Chukchi, being extremely low-resource, is a great candidate for this experiment. For a broader description of the model and this approach we suggest addressing the original article by Facebook AI team (Conneau, et al., 2020).

We hypothesized that through using a model that has learned generalized natural language representation, it will be easier to train a model that recognizes Chukchi speech - basically, a big part of the training process is skipped, and the whole training process is essentially condensed to fine-tuning. Moreover, the authors of the mentioned article have proven that multilingual models may even outperform monolingual ones for some languages.

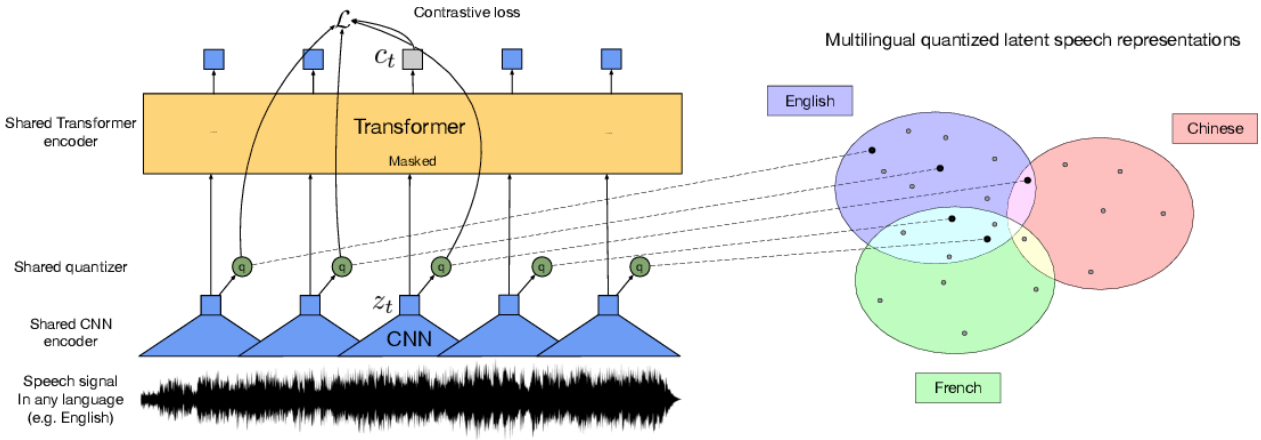

Figure 2. A shared quantization module over feature encoder representations produces multilingual embeddings, through which the model learns to share discrete tokens across languages, creating bridges across languages (Conneau, et al., 2020).

5.2.2 Experiment

After the necessary initial steps, such as feature extraction, normalization, sampling rate adjustment, etc., the model was further trained on the annotated Chukchi data at hand. The CNN part from Figure 2 in section 5.2.1 does not need fine-tuning, because it was trained by the authors of the model where the multilingual features are encoded. Moreover, the fine-tuning process is relatively short, and only lasts for 6 epochs, 400 steps each, because the model has shown to be prone to overfitting on such a small dataset. The training rate was set to a static value of $3e^{-4}$, which fits in the range suggested by the authors of the original paper. It seems to make little sense to change learning rate dynamically, given the low amount of training data.

The model will be publicly available at our team's GitHub repository.[21]

5.2.3 Results

The data that was used for fine-tuning consisted of annotated data that had the following structure:

Table 4: Training and Testing times for the XLSR model

TRAIN	02:49:36.35
TEST	00:17:09.25
TOTAL	**03:06:45.00**

This allowed the fine-tuned model to achieve a WER[22] of 0.758395 and CER[23] 0.186895 - surprisingly low for such a small dataset. We plan to prepare more data and further improve this result, as it has appeared to be the most fruitful approach among others. Additionally, it appears to be useful to compare these metrics with the results achieved in the original article.

Table 5: Word Error Rate

[21] Project GitHub: https://github.com/ftyers/fieldasr/blob/main/DATA.md

[22] WER is a metric that is commonly used for speech modelling and speech recognition. It is computed at word-level as **(Deletions + Insertions + Substitutions) / N**, where N is the number of words in a text, and the three components of the dividend are the numbers of corresponding operations, applied to words, that are needed to reconstruct the original text.

[23] CER is computed the same way as WER, but on character level. That means that the same formula is used and the same operations are counted, but in appliance to characters in the original text.

Language	*Assamese*	*Tagalog*	*Swahili*	*Georgian*	*Our: Chukchi*
WER	44.1	33.2	26.5	31.1	75.8
Fine-tuning data, hrs	55	76	30	46	3

WER proved to not be the best metric for training in Chukchi, as it accounts for any mistake in a word. Considering the nature of this language is polysynthetic (e.g. it contains long “sentence-words”), it is easy to achieve a high error rate due to a large number of substitutions. CER is, in our opinion, a more suitable metric in these circumstances. Expectedly, in comparison with the results in the table above, our model does not perform satisfactorily on the word-level.

Table 6: Character Error Rate

Language	*Assamese*	*Tagalog*	*Swahili*	*Lao*	*Ours: Chukchi*
CER	17.9	13.1	21.3	22.4	18.7
Fine-tuning data, hrs	55	76	30	59	3

Character error rate accounts for character-level mistakes. Our model has achieved a comparable result, given much less data. This result is very satisfying, however, low WER definitely means that this model is not suitable for production use. However, if one takes into consideration the morphological complexity of Chukchi and its incredibly scarce data, one can conclude that XLSR has proven to be an astoundingly powerful base for low-resource ASR.

As a demonstration, we would like to finish this part of the article with an example of the model’s output:

Table 7: Output from the XLSR model

Original	Recognised

ӈутингивик ымыӆьычукоткак ӈыраӄ аӄатвагыргын гатваленат яама нымытваӆьа милгэрти ыннэнчьэн о'равэтӆьан егтэлытваркын ӄутти ӈыроӄ гэвъилинэт	*ӈутингивик м чукуткак ӈыаӄалтвагыргыт гатваленат яма нымытваӆьа милгэри ынэчьэннорав эт лан егтэлверкын отиӈыргэвилиэт*

5.3 Wav2Vec Unsupervised

5.3.1 Theory

Another model suitable for our task, wav2vec Unsupervised (wav2vec-U), was presented in May 2021 by Facebook AI (Baevski, et al. 2021). The main feature of this model is that only separate audio and texts that are not related to each other are needed (not annotated ones), which makes it possible to apply it to low-resource languages that lack text and audio pairs. A good result (about 85% for low-resource languages, such as Tatar) is obtained due to the self-supervised model (wav2vec 2.0), a simple k-means clustering method and a Generative Adversarial Network (GAN). The illustration of wav2vec-U can be found in Figure 3. The wav2vec-U training procedure consists of three main steps:

- Preparation of speech representations and text data;
- Generative Adversarial Training (GAN);
- Iterative self-training + Kaldi LM-decoding.

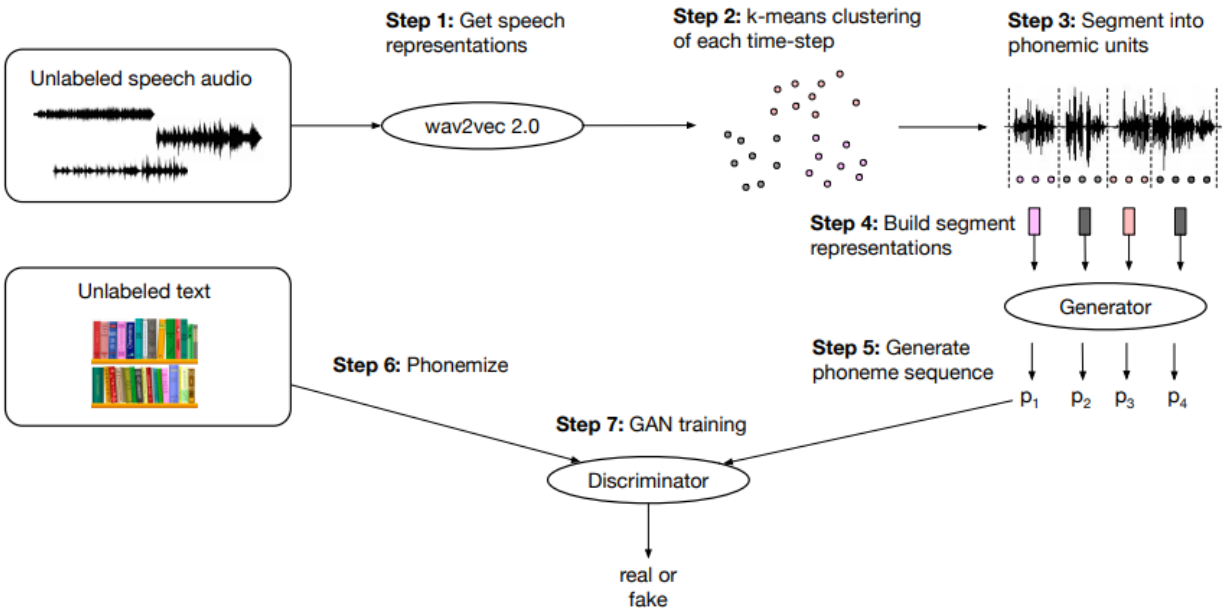

Figure 3. Illustration of wav2vec Unsupervised (Baevski, et al. 2021)

5.3.2 Experiment

As mentioned above, using the model requires specific preprocessing of the data, which requires wav2vec 2.0 (open-source framework for self-supervised learning of representations from raw audio data), eSpeak (a compact open source software speech synthesizer for English and other languages) and fastText (a library for efficient learning of word representations and sentence classification) pre-trained models for Chukchi, which do not exist. It is possible to train these models from scratch, however, it takes a lot of time, and at the moment of writing this article it was not possible to preprocess the data properly for the wav2vec-U model.

5.3.3 Results

No results for wav2vec-U have been obtained at this time, this will be a task for the next study.

6. CONCLUSION

In this work we introduced our project, focused on creating a new automatic speech recognition (ASR) system for low-resource languages, with a focus on Chukchi. Our goal was to create a system as unsupervised as possible. In order to run experiments and train our selected models to work for Chukchi, we collected a sizable corpus of both audio and text data in Chukchi, a feat that was rather unexpected for such a low-resource language. In total, there were 15,309 sentences, and 117,567 words for the full text corpus, and 21 hours, 34 minutes, 23 seconds worth of audio data. We then proceeded to conduct 3 different experiments centered around VQ-VAE, XLSR, and wav2vec models.

The first experiment, regarding Vector Quantized - Variational Auto-Encoder (VQ-VAE), demonstrated very little change in validation loss between the first and thousandth epoch that had been executed. This result was found in all three sub-experiments that were run with both augmented and unaugmented data. This minimal change is likely a result of how little data we had for this model, as well as the poor sound quality of the audio files. The second experiment, focused on XLSR, was shown to be more promising than the first, with a surprisingly low WER of 0.758395 and CER 0.186895. XLSR actually proved to be the most powerful base for development in low-resource ASR. These results on their own may not appear

particularly noteworthy, however, we must take into account the fact that Chukchi is a polysynthetic language, meaning that words and sentences tend to have more of an overlapping appearance when manifested in written language. With this in mind, our interpretation of the CER result becomes much more significant. Finally, the third experiment conducted through wav2vec, unfortunately, proved fruitless. There are no results to interpret at this time, as preprocessing of the data was unsuccessful. This experiment will be pursued in further studies.

In summary, we were able to train 2 separate models for work in Chukchi, and will be attempting to train the third in the near future. Despite some unfortunate setbacks, we have made an incredible amount of progress in our understanding of low-resource automatic speech recognition, have learned what works for this particular set of data and what proves ineffective, and set tasks that can be completed in future research. Ultimately, we consider this to be a successful project with a promising outlook on the future of low-resource automatic speech recognition.

Works Cited

- Baevski, Alexei, et al. “Unsupervised Speech Recognition.” *ArXiv.org*, 24 May 2021, arxiv.org/abs/2105.11084.
- Baevski, Alexei, et al. “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.” *ArXiv.org*, 22 Oct. 2020, arxiv.org/abs/2006.11477v3.
- Chorowski, Jan, et al. “Unsupervised Speech Representation Learning Using WaveNet Autoencoders.” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, 2019, pp. 2041–2053., doi:10.1109/taslp.2019.2938863.
- “Chukchi (Ԓыгъоравэтԓьэн Йиԓыйиԓ).” *Chukchi Language, Alphabet and Pronunciation*, omniglot.com/writing/chukchi.htm.
- Conneau, Alexis, et al. “Unsupervised Cross-Lingual Representation Learning for Speech Recognition.” *ArXiv.org*, 15 Dec. 2020, arxiv.org/abs/2006.13979v2.
- Fortescue, Michael. *Comparative Chukotko-Kamchatkan Dictionary*, Mouton De Gruyter, 2005.
- Kingma, Diederik P, and Max Welling. “Auto-Encoding Variational Bayes.” *ArXiv.org*, 9 Jan. 2014, arxiv.org/abs/1312.6114v5.
- Kriman, Samuel, et al. “Quartznet: Deep Automatic Speech Recognition with 1D Time-Channel Separable Convolutions.” *ICASSP 2020 - 2020 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, doi:10.1109/icassp40776.2020.9053889.

- Magueresse, Alexandre, et al. “Low-Resource Languages: A Review of Past Work and Future Challenges.” *ArXiv.org*, 12 June 2020, arxiv.org/abs/2006.07264v1.
- Oord, Aaron van den, et al. “Neural Discrete Representation Learning.” *ArXiv.org*, 30 May 2018, arxiv.org/abs/1711.00937v2.
- Ren, Yi, et al. “Almost Unsupervised Text to Speech and Automatic Speech Recognition.” *ArXiv.org*, 26 July 2020, arxiv.org/abs/1905.06791.
- Sarma, Mousmita, and Kandarpa Kumar Sarma. “Acoustic Modeling of Speech Signal Using Artificial Neural Network.” *Advances in Systems Analysis, Software Engineering, and High Performance Computing*, 2015, pp. 282–299., doi:10.4018/978-1-4666-8493-5.ch012.
- Schneider, Steffen, et al. “wav2vec: Unsupervised Pre-Training for Speech Recognition.” *ArXiv.org*, 2 July 2019, arxiv.org/abs/1904.05862v3.
- Xiao, Yubei, et al. “Adversarial Meta Sampling for Multilingual Low-Resource Speech Recognition.” *ArXiv.org*, 12 Apr. 2021, arxiv.org/abs/2012.11896.