Simple and Effective Zero-shot Cross-lingual Phoneme Recognition

Qiantong Xu*, Alexei Baevski, Michael Auli

Meta AI

{qiantong, abaevski, michaelauli}@fb.com

Abstract

Recent progress in self-training, self-supervised pretraining and unsupervised learning enabled well performing speech recognition systems without any labeled data. However, in many cases there is labeled data available for related languages which is not utilized by these methods. This paper extends previous work on zero-shot cross-lingual transfer learning by fine-tuning a multilingually pretrained wav2vec 2.0 model to transcribe unseen languages. This is done by mapping phonemes of the training languages to the target language using articulatory features. Experiments show that this simple method significantly outperforms prior work which introduced task-specific architectures and used only part of a monolingually pretrained model.

Index Terms zero-shot transfer learning, cross-lingual, phoneme recognition, multilingual ASR

1. Introduction

There is a large number of languages spoken around the world of which only a small fraction is served by speech technology. A large barrier to making speech technology more accessible is the requirement for large amounts of transcribed speech audio by current models which is simply not available for the vast majority of languages. Speech recognition accuracy has been steadily improving by recent advances in supervised multilingual modeling [1, 2], self-supervised learning [3, 4, 5, 6, 7], and semi-supervised learning [8, 9, 10, 11, 12], particularly for low-resource languages. This recently led to good speech recognition performance in settings where no labeled data exists at all [13, 14, 15]. One downside of these approaches is that they require training a separate unsupervised model for each language while ignoring the presence of labeled data in related languages.

Zero-shot transfer learning addresses this by training a single multilingual model on the labeled data of several languages to enable zero-shot transcription of unseen languages [16, 17, 18, 19, 17, 20, 21]. Models usually have a common encoder that extracts acoustic information from speech audio and then predicts either a shared phoneme vocabulary [17, 16] or language-specific phonemes [1, 20, 22]. The former requires either phonological units that are agnostic to any particular language such as articulatory features [20] or global phones [23, 17].

In this paper, we study a simple zero-shot transfer learning approach which builds a global phoneme recognizer by simply considering all possible phonemes of the training languages and then decodes the model with a language model to generate the final phoneme sequence. The lexicon is built from articulatory features to map the phonemes between the training and target vocabulary. Our method makes no assumption about the relation of training and testing languages, including attributes like phoneme distribution or coverage. We extend prior work by using unsupervised cross-lingually pretrained representations estimated on 53 languages [24] instead of monolingually trained representations [16] and our approach also uses the full pretrained model instead of only the feature-extractor [16].

We conduct experiments on 42 languages of CommonVoice [25], 19 languages of BABEL [26] and six languages of MLS [27]. Results show significant improvements on unseen languages over the approach of [16] and cross-lingual pretrained representations are more effective. Finally, zero-shot transfer learning performs comparably to unsupervised approaches with the benefit of being able to transcribe multiple unseen languages using a single model.

2. Approach

Our approach entails the use of self-supervised representations trained on data in many languages [(24), §2.1]. Next we simultaneously fine-tune the model to perform phoneme recognition on data in multiple training languages. At inference time, we test the fine-tuned model on all unseen languages using a mapping of the phonemes from the training vocabulary to the ones in the target languages (§2.2).

2.1. Self-supervised Model Training

We use XLSR-53, a wav2vec 2.0 model pretrained on data in 53 languages [24, 6]. This model contains a convolutional feature encoder \( f : \mathcal{X} \mapsto \mathcal{Z} \) to map raw audio \( \mathcal{X} \) to latent speech representations \( z_t, \ldots, z_T \) which are input to a Transformer \( g : \mathcal{Z} \mapsto \mathcal{C} \) to output context representations \( c_1, \ldots, c_T \) [28, 29]. Each \( z_t \) represents about 25ms of audio strided by 20ms and the Transformer architecture follows BERT [30, 28].

During training, feature encoder representations are discretized to \( q_1, \ldots, q_T \) with a quantization module \( \mathcal{Z} \mapsto \mathcal{Q} \) to represent the targets in the objective. The quantization module uses a Gumbel softmax to choose entries from the codebooks and the chosen entries are concatenated to obtain \( q \) [31, 32, 29].

The model is trained by solving a contrastive task over masked feature encoder outputs. At training time, spans of ten time steps with random starting indices are masked. The objective requires identifying the true quantized latent \( q_t \) for a masked time-step within a set of \( K = 100 \) distractors \( Q_t \) sampled from other masked time steps:

\[
-\log \frac{\exp(\text{sim}(e_t, q_t))}{\sum_{q \in Q_t} \exp(\text{sim}(e_t, q))}
\]

where \( e_t \) is the output of the Transformer, and \( \text{sim}(a, b) \) denotes cosine similarity. The objective is augmented by a codebook diversity penalty to encourage the model to use all codebook entries [33].

2.2. Phoneme Mapping

We use phonemes as modeling units and in particular, the symbols of the standard International Phonetic Alphabet (IPA).
However, the vocabulary estimated from the training languages may not cover the full vocabulary of the target languages which results in out-of-vocabulary (OOV) phonemes at test time. We address this by mapping between the training and target vocabularies based on articulatory/phonological features [34]. Articulatory feature is a set of global attributes to describe any sound or phone. There are four groups of attributes: major class (syllabic, vocalic, approximant, sonorant), manner (continuant, lateral, nasal, strident), place (labial, coronal, dorsal, pharyngeal), and laryngeal (voiced, aspirated, glottalized). Each attribute can be either positive or negative.

We compute the distance between each pair of phonemes using the Hamming edit distance between the articulatory feature vectors⁴, and then generate two types of simple many-to-one mapping lexicons:

- **tr2tgt lexicon** maps each phoneme in the training vocabulary to its closest one in the target vocabulary. Then for the remaining uncovered phonemes in the target vocabulary, it maps the closest ones in the training vocabulary to them.

- **tgt2tr lexicon** that maps for each phoneme in the target vocabulary, the phonemes in the training vocabulary that have 0 distance to it.

We compare both below (§4.3.2) and use tr2tgt unless otherwise mentioned.

### 3. Experimental setup

#### 3.1. Datasets

We consider three multilingual corpora and a variety of languages to evaluate our approach. All the audios are up-down-sampled to 16kHz.

**Multilingual LibriSpeech (MLS)** is a large corpus of read audiobooks from Librivox and we experiment with the same six languages as [15]: Dutch (du), French (fr), German (de), Italian (it), Portuguese (pt), Spanish (es). We use the same split as [15] for validation and test.

**CommonVoice (CV)** is a multilingual corpus of read speech comprising more than two thousand hours of speech data in 76 languages [25]. We use the December 2020 release (v6.1) for training and fine-tuning models. We select 42 languages in total that are supported by our phonemizer (see §3.2) as well as their official train, dev and test splits. Italian (it) serves as validation and test.

**BABEL** is a multilingual corpus of conversational telephone speech from IARPA, which includes Asian and African language [26]. We include 21 languages from it (Table 1). We include Cantonese and Lao in the test set to compare with [16] and the remaining 19 languages in the training set. Italian serves for validation.

#### 3.2. Pre-processing and Phonemization

We first normalize all transcriptions for CommonVoice and BABEL by removing punctuation and rare characters. Rare characters are usually numbers or characters from other languages. We then obtain the phonemic annotations from the word transcriptions using ESpeak⁶, as well as [35] based on Phonetisaurus⁷ to compare with [16]. Specifically, we use Espeak on MLS, Phonetisaurus on BABEL.

#### 3.3. Model Training

Models are implemented in fairseq [36] and we use the pre-trained XLSR-53 model [24] which has 24 Transformer blocks, model dimension 1024, inner dimension 4096 and 16 attention heads. It is pretrained on the joint training set of MLS, CommonVoice and BABEL, which consists of about 56K hours of speech data.

To fine-tune the model we add a classifier representing the joint vocabulary of the training languages on top of the model and train on the labeled data with a Connectionist Temporal Classification (CTC) loss [37]. Weights of the feature encoder are not updated at fine-tuning time, while the Transformer weights are finetuned after 10k updates. We determine the best transformer final dropout in [0, 0.3], learning rates setting in [5e-6, 5e-4].

The learning rate schedule has three phases: warm up for the first 10% of updates, keep constant for 40% and then linearly decay for the remainder. The models were finetuned for

---

⁴https://github.com/dmort27/panphon. In this repository, each feature articulatory vector contains 21 attributes

⁶https://github.com/espeak-ng/espeak-ng

⁷https://github.com/AdolfVonKleist/Phonetisaurus
Table 2: Comparison to prior zero-shot work [16] in terms of phonetic token error rate (PTER) on the test sets of a subset of BABEL languages. Cantonese and Lao are the unseen languages. Models are trained on 6 or 19 languages of BABEL (BB-6/19), 21 languages of CommonVoice (CV-21), Globalphone (GP) and the Spoken Dutch Corpus (CGN).

<table>
<thead>
<tr>
<th></th>
<th>Gao et al. [16]</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB Data</td>
<td>BB-6 [16]</td>
<td>BB-6 BB-19 BB-19 BB-19</td>
</tr>
<tr>
<td>Other Data</td>
<td>CGN+GP</td>
<td>- CV-21 CV-21</td>
</tr>
<tr>
<td># hours / lang</td>
<td>all</td>
<td>all 10 10</td>
</tr>
<tr>
<td># hours total</td>
<td>1,492</td>
<td>317 935 118 298</td>
</tr>
</tbody>
</table>

Supervised

- Bengali: 38.2 -> 36.1 -> 35.4 -> 53.2 -> 40.7
- Vietnamese: 32.0 -> 40.7 -> 42.1 -> 71.0 -> 63.3
- Zulu: 35.2 -> 34.6 -> 34.8 -> 61.0 -> 44.1
- Amharic: 38.0 -> 35.5 -> 35.5 -> 63.2 -> 42.8
- Javanese: 44.2 -> 40.2 -> 40.8 -> 57.4 -> 49.1
- Georgian: 38.6 -> 27.6 -> 43.8 -> 51.6 -> 43.2

Zero-shot

- Cantonese: 73.1 -> 73.6 -> 72.6 -> 70.9 -> 63.6
- Lao: 69.3 -> 70.3 -> 70.2 -> 72.1 -> 63.7

Table 3: Unsupervised ASR (w2v-U) vs. zero-shot ASR (This work). Results are reported in phoneme error rate (PER) on MLS.

<table>
<thead>
<tr>
<th>de</th>
<th>nl</th>
<th>fr</th>
<th>es</th>
<th>it</th>
<th>pt</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>w2v-U [15]</td>
<td>21.6</td>
<td>25.0</td>
<td>27.7</td>
<td>20.2</td>
<td>31.2</td>
<td>36.0</td>
</tr>
<tr>
<td>+ n-gram LM</td>
<td>16.2</td>
<td>17.8</td>
<td>26.5</td>
<td>18.1</td>
<td>28.6</td>
<td>30.6</td>
</tr>
<tr>
<td>This work</td>
<td>23.8</td>
<td>38.0</td>
<td>31.0</td>
<td>28.7</td>
<td>33.5</td>
<td>45.0</td>
</tr>
<tr>
<td>+ n-gram LM</td>
<td>14.8</td>
<td>26.0</td>
<td>26.4</td>
<td>12.3</td>
<td>21.7</td>
<td>36.5</td>
</tr>
</tbody>
</table>

25k updates on 4 GPUs. The best checkpoints are selected by the validation error on the validations set for BABEL and CommonVoice; while for MLS, it is selected using the unsupervised cross validation metric of [15] to enable a direct comparison.

3.4. Decoding

The wav2letter beam-search decoder [38] is used to generate the final transcriptions with the lexicon and an external 6-gram language model trained on the phoneme annotations of the labeled training data. Beam size is set to 50 in all the inference experiments. The lexicons mentioned above limits the search space to only the valid phones in the training vocabulary and ensures the decoder predicts only phones in the target dictionary.

4. Results

4.1. Comparison to other zero-shot work

In our first experiment, we compare performance to the zero-shot transfer learning approach of [16] which used only the feature extractor of a wav2vec 2.0 model trained on English. The training data on CommonVoice and BABEL is prepared in the same way as [16] and we report the same phonetic token error rate (PTER) metric, in which each IPA token is treated as separate suprasegmentals (such as long vowels, and primary stress symbol), tones, diphthongs and affricates.

Table 2 shows that finetuning on only 6 languages of BABEL (BB-6) with our method can outperform [16] on the supervised languages while using only 317 hours of labeled data compared to nearly 1.5K hours. This shows that using the full pretraining model is beneficial. Using more languages (BB-19) improves performance on the zero-shot Cantonese setting.

The CV-21 setting performs significantly less well on the supervised languages because of domain-mismatch since the test languages are BABEL, however, on the zero-shot languages CV-21 performs similarly to BB-19. This indicates that domain is less of a challenge in the challenging zero-shot setting where error rates are relatively higher.

Finally, our approach can outperform [16] on the zero-shot directions when using both the CommonVoice and BABEL data while restricting the amount of labeled data to 10 hours for each language. This results in fewer than 300 hours of labeled data since some languages do not even have 10 hours of labeled data.

4.2. Comparison to unsupervised learning

Next, we compare zero-shot transfer learning to wav2vec-U [15], both of which use the same pretrained representations (XLSR-53). We use 10 hours of labeled data for each MLS language as prepared in [24] and measure the performance when fine-tuning XLSR-53 on five of the six languages and then evaluate on the held-out language. Table 3 shows that the performance of zero-shot transfer learning is on par to wav2vec-U [15] while using a simpler training and inference pipeline.

4.3. Ablations

In this section, we analyze the importance of pretraining, cross-lingual pretraining, lexicon construction strategies as well as the impact of different phonemizers. We use the CommonVoice benchmark for these experiments (Table 1).

4.3.1. Effect of multilingual pretraining

Multilingual pretraining plays an important role for the model to perform well on unseen languages. To get a better sense of this, we compare the performance without pretraining to pretraining using either up to 10h or up to 200h of labeled training data per language for a subset of the CommonVoice languages.

For pretraining we consider an English-only pretrained model, wav2vec 2.0 pretrained on the full Libri-light training data, as
Table 4: Comparison of PER on the test sets of a subset of Common Voice languages when using either at most 10h or 200h of labeled data per language without pretraining (No pretrain), English-only pretraining of wav2vec 2.0 on 60K hours of Libri-Light data [6, 39], or cross-lingual pretraining using the XLSR-53 model which was trained on 53 different languages [24]. We show the maximum number of labeled training hours per language and the number of training hours in total for each setting. Results are based on Viterbi decoding from the acoustic model without a language model but with a lexicon and with a 6-gram language model.

<table>
<thead>
<tr>
<th>Model</th>
<th>No pretrain</th>
<th>No pretrain</th>
<th>w2v LV-60K</th>
<th>XLSR-53</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># hours / language</td>
<td>10</td>
<td># hours in total</td>
<td>149</td>
</tr>
<tr>
<td></td>
<td>viterbi</td>
<td>n-gram</td>
<td>viterbi</td>
<td>n-gram</td>
</tr>
<tr>
<td>it</td>
<td>56.6</td>
<td>47.5</td>
<td>50.1</td>
<td>41.8</td>
</tr>
<tr>
<td>eu</td>
<td>51.2</td>
<td>45.6</td>
<td>39.7</td>
<td>32.1</td>
</tr>
<tr>
<td>ia</td>
<td>38.9</td>
<td>27.8</td>
<td>30.9</td>
<td>23.0</td>
</tr>
<tr>
<td>lv</td>
<td>65.2</td>
<td>59.8</td>
<td>62.7</td>
<td>56.5</td>
</tr>
<tr>
<td>ka</td>
<td>61.8</td>
<td>56.1</td>
<td>54.6</td>
<td>48.9</td>
</tr>
<tr>
<td>nl</td>
<td>66.3</td>
<td>56.8</td>
<td>63.0</td>
<td>56.1</td>
</tr>
<tr>
<td>el</td>
<td>49.5</td>
<td>40.6</td>
<td>42.1</td>
<td>33.7</td>
</tr>
<tr>
<td>ro</td>
<td>45.7</td>
<td>34.7</td>
<td>46.7</td>
<td>36.9</td>
</tr>
<tr>
<td>mt</td>
<td>66.3</td>
<td>60.2</td>
<td>62.0</td>
<td>56.0</td>
</tr>
<tr>
<td>tt</td>
<td>68.2</td>
<td>63.9</td>
<td>65.1</td>
<td>60.8</td>
</tr>
<tr>
<td>fi</td>
<td>58.8</td>
<td>55.6</td>
<td>53.8</td>
<td>48.3</td>
</tr>
<tr>
<td>sl</td>
<td>62.9</td>
<td>56.0</td>
<td>60.5</td>
<td>54.6</td>
</tr>
<tr>
<td>pl</td>
<td>62.3</td>
<td>59.3</td>
<td>60.2</td>
<td>56.0</td>
</tr>
<tr>
<td>Avg</td>
<td>58.0</td>
<td>51.1</td>
<td>53.2</td>
<td>46.5</td>
</tr>
</tbody>
</table>

Table 5: Effect of lexicon construction strategies (§2.2) and different phonemizers (§3.2) on CommonVoice in terms of PER: tr2tgt denotes a lexicon constructed by mapping training language phonemes to target language phonemes and tgt2tr denotes the reverse strategy. Average PER excludes “eu” and “ia” since they are not supported by Phonetisaurus.

<table>
<thead>
<tr>
<th>Phonemizer</th>
<th>Espeak</th>
<th>Phonetisaurus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexicon</td>
<td>tr2tgt</td>
<td>tgt2tr</td>
</tr>
<tr>
<td>Avg</td>
<td>24.5</td>
<td>24.6</td>
</tr>
</tbody>
</table>

5. Conclusion

In this work, we investigate zero-shot transfer learning on cross-lingual phoneme recognition using a cross-lingually pretrained self-supervised model. Pretraining vastly improves accuracy over no pretraining, even when a moderate amount of labeled data is used, and cross-lingual pretraining performs better than monolingual pretraining. Our simple approach of fine-tuning a large pretrained model performs better than prior work which only used the feature extractor of a monolingually pre-trained wav2vec 2.0 model and which relied on task-specific architectures such as language embeddings. We also show that our approach performs on par to the recently introduced unsupervised speech recognition work of [15] which does not use labeled data from related languages and requires training separate models for each target language.

To better understand the impact of this, we use both Espeak and Phonetisaurus (§3.2) and evaluate them on both types of lexicon construction techniques. Table 5 indicates that both phonemizers show the same trend in performance for tr2tgt/tgt2tr.
6. References


