SIMPLE AND EFFECTIVE ZERO-SHOT CROSS-LINGUAL PHONEME RECOGNITION

Qiantong Xu, Alexei Baevski, Michael Auli
Facebook AI Research

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]. Models usually have a common encoder that extracts acoustic information from speech audio and then predict either a shared phoneme vocabulary [17, 16] or language-specific phonemes [11, 20, 21]. The former requires either phonological units that are agnostic to any particular language such as articulatory features [20] or global phones [22, 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 [23] 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 Common-Voice [24], 19 languages of BABEL [25] and six languages of MLS [26]. 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 [23, §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 [22].

2.1. Self-supervised Model Training

We use XLSR-53, a wav2vec 2.0 model pretrained on data in 53 languages [23, 6]. This model contains a convolutional feature encoder \( f : \mathcal{X} \rightarrow \mathcal{Z} \) to map raw audio \( \mathcal{X} \) to latent speech representations \( z_1, \ldots, z_T \) which are input to a Transformer \( g : \mathcal{Z} \rightarrow \mathcal{C} \) to output context representations \( c_1, \ldots, c_T \) [27, 28]. Each \( z_t \) represents about 25ms of audio strided by 20ms and the Transformer architecture follows BERT [29, 27]. During training, feature encoder representa-
Articulatory vector contains 21 attributes using the hamming edit distance between the articulatory features. Each attribute can be either positive or negative. 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, sonorant). We compute the distance between each pair of phonemes to describe any sound or phone.

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. 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 vectors 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. 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.

### Table 1. Splits of CommonVoice (CV) and BABEL (BB). The 6 BABEL languages of [16] are bolded.

| Split     | Languages                                      |
|-----------|------------------------------------------------|
| **CommonVoice (CV)** | Esperanto(eo), Lithuanian(lt), Welsh(cy), Tamil(ta), Swedish(sv-SE), German(de), English(en), Oriya(or), Hindi(hi), Persian(fa), Japanese(ja), Assamese(as), Indonesian(id), Catalan(ca), Spanish(es), French(fr), Portuguese(pt), Arabic(ar), Chinese(zh-CN), Chinese(zh-TW), Turkish(tr), Estonian(et), Hungarian(hu), Russian(ru), Czech(cs) |
| **Train** | Italian(it)                                    |
|           | Basque(eu), Interlingua(ia), Latvian(lv), Georgian(ka), Irish(ga-ie), Dutch(nl), Greek(el), Punjabi(pa-in), Romanian(ro), Maltese(mt), Chinese(zh-HK), Tatar(tt), Finnish(fi), Slovenian(sl), Polish(pl), Kirghiz(ky)                                      |
| **Dev**   | Amharic(am), Bengali(bn), Cebuano(ceb), Igbo(ig), Haitian(ht), Javanese(jv), Mongolian(mn), Swahili(sw), Tamil(ta), Vietnamese(vi), Assamese(as), Dholuo(luo), Guarani(gn), Kazakh(kk), Pashto(ps), Georgian(ka), Tagalog( tl), Telugu(te), Turkish(tr), Zulu(zu) |
| **Test**  | CV-Italian(it)                                 |
|           | Italian(it)                                    |
|           | Cantonese(yue), Lao(lo)                        |

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. 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 language for development, for training we use a total of 26 languages and the remaining 13 languages are for testing (Table 1). For each language in the test set, we also make sure that there is at least one language that belongs to the same language family as in the training set. Compared to other datasets such as BABEL or MLS, CommonVoice is well suited for zero-shot transfer learning, since it covers a larger number of languages.

**BABEL** is a multilingual corpus of conversational telephone speech from IARPA, which includes Asian and African language. We include 21 languages from it (Table 1). 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. 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 vectors 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.

---

1. https://github.com/dmort27/panphon. In this repository, each feature articulatory vector contains 21 attributes.
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 ESpeak\(^2\) as well as \(^3\) based on Phonetisaurus\(^3\) to compare with [16]. Specifically, we use ESpeak on MLS, Phonetisaurus on BABEL.

3.3. Model Training

Models are implemented in fairseq \(^4\) and we use the pretrained XLSR-53 model \(^5\) 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 \(^6\). We determine the best transformer final dropout in \([0, 0.3]\) and the learning rate 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 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 \(^7\) 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 with unsupervised method

In our first experiment, 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 \([23]\) and measure the phoneme error rate (PER) on MLS.

|                  | de | nl | fr | es | it | pt | Avg |
|------------------|----|----|----|----|----|----|-----|
| w2v-U \([15]\)   | 21.6 | 25.0 | 27.7 | 20.2 | 31.2 | 36.0 | 27.0 |
| + n-gram LM      | 16.2 | 17.8 | 26.5 | 18.1 | 28.6 | 30.6 | 23.0 |
| This work        | 23.8 | 38.0 | 31.0 | 28.7 | 33.5 | 45.0 | 33.3 |
| + n-gram LM      | 14.8 | 26.0 | 26.4 | 12.3 | 21.7 | 36.5 | 22.9 |

Table 2. Unsupervised ASR (w2v-U) vs. zero-shot ASR (This work). Results are in terms of phoneme error rate (PER) on MLS.

Table 3 shows that finetuning on only 6 languages of BABEL and CommonVoice outperforms wav2vec-U \([15]\) by 13.6% 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).

4.2. Comparison to other zero-shot work

Next, 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 3. 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).

|                  | Gao et al. \([16]\) | This work |
|------------------|-------------------|-----------|
| BB Data | BB-6 | BB-6 | BB-19 |
| Other Data | CGN+GP | - | CV-21 |
| # hours / lang | all | all | 10 |
| # hours total | 1,492 | 317 | 298 |
| Supervised | | | |
| Bengali | 38.2 | 36.1 | 40.7 |
| Vietnamese | 32.0 | 40.7 | 63.3 |
| Zulu | 35.2 | 34.6 | 44.1 |
| Amharic | 38.0 | 35.5 | 42.8 |
| Javanese | 44.2 | 40.2 | 49.1 |
| Georgian | 38.6 | 27.6 | 43.2 |
| Zero-shot | | | |
| Cantonese | 73.1 | 73.6 | 63.6 |
| Lao | 69.3 | 70.3 | 63.7 |

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.
**Table 4.** Effect of no pretraining, monolingual pre-training (w2v LV-60K) and multilingual pretraining (XLSR-53) in terms of PER on the test languages of CommonVoice.

| # hours / lang | No pretrain | w2v LV-60K | XLSR-53 |
|----------------|-------------|------------|---------|
| # hours total  | 149         | 1156       | 149     |

| Language | 10 | 200 | 10 | 10 | 149 |
|----------|----|-----|----|----|-----|
| it       | 47.5 | 41.8 | 16.9 | 13.9 |
| eu       | 45.6 | 32.1 | 16.3 | 13.7 |
| ia       | 27.8 | 23.0 | 6.7 | 6.1 |
| lv       | 59.8 | 56.5 | 33.5 | 32.3 |
| ka       | 56.1 | 48.9 | 24.0 | 23.8 |
| nl       | 56.8 | 56.1 | 30.5 | 19.8 |
| el       | 40.6 | 33.7 | 10.7 | 10.4 |
| ro       | 34.7 | 36.9 | 15.0 | 14.8 |
| mt       | 60.2 | 56.0 | 36.1 | 35.9 |
| tt       | 63.9 | 60.8 | 34.7 | 37.4 |
| fi       | 55.6 | 48.3 | 29.9 | 29.0 |
| sl       | 56.0 | 54.6 | 29.0 | 26.1 |
| pl       | 59.3 | 56.0 | 27.3 | 25.7 |
| Avg      | 51.1 | 46.5 | 23.9 | 22.2 |

BEL (BB-6) with our method can outperform [16] on the supervised languages while using 317 hours of labeled data compared to nearly 1.5K hours. This shows that using the full pretraining model is beneficial. Our approach can outperform [16] on the zero-shot directions when we add CommonVoice 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.

### 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. Table 1 shows that accuracy without pretraining performs vastly less well than pretraining-based approaches, even when the amount of labeled data is increased by up to a factor of 20. This is inline with prior work on automatic speech recognition [6]. Furthermore, multilingual pre-training (XLSR-53) performs better than monolingual pretraining on English data (w2v LV-60K) on every single language.

---

**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.

| Phonemizer   | Espeak | Phonetisaurus |
|--------------|--------|---------------|
| Lexicon      | tr2tgt | tgt2tr        |
| Avg          | 24.5   | 24.6          |
|              | 31.7   | 32.4          |

#### 4.3.2. Comparison of lexicon and phonemizers

Next, we compare the decoding performance with different lexicons. Table 5 shows that tr2tgt is slightly better than tgt2tr on average for different phonemizers. Different phonemizers can generate fairly different phoneme sequences given the same word transcriptions which may impact the final performance of our models. 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.

### 5. Conclusions

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.
6. REFERENCES

[1] S. Dalmia, R. Sanabria, F. Metze, and A. Black, “Sequence-based multi-lingual low resource speech recognition,” in Proc. of ICASSP. IEEE, 2018.

[2] V. Pratap, A. Sriram, et al., “Massively multilingual asr: 50 languages, 1 model, 1 billion parameters,” arXiv preprint arXiv:2007.03001, 2020.

[3] Aä. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” in Proc. of NeurIPS, 2018.

[4] Y.-A. Chung and J. Glass, “Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech,” in Proc. of Interspeech, 2018.

[5] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, “An unsupervised autoregressive model for speech representation learning,” in Proc. of Interspeech, 2019.

[6] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” arXiv preprint arXiv:2006.11477, 2020.

[7] W.-N. Hsu et al., “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” arXiv preprint arXiv:2106.07447, 2021.

[8] G. Synnaeve, Q. Xu, et al., “End-to-end ASR: from Supervised to Semi-Supervised Learning with Modern Architectures,” arXiv, vol. abs/1911.08460, 2019.

[9] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, “Iterative pseudo-labeling for speech recognition,” arXiv, 2020.

[10] T. Likhomanenko, Q. Xu, J. Kahn, G. Synnaeve, and R. Collobert, “slimipl: Language-model-free iterative pseudo-labeling,” arXiv preprint arXiv:2010.11524, 2020.

[11] Qiantong X., Alexei B., et al., “Self-training and pre-training are complementary for speech recognition,” in Proc. of ICASSP. IEEE, 2021.

[12] D. Park, Y. Zhang, Y. Jia, et al., “Improved noisy student training for automatic speech recognition,” arXiv preprint arXiv:2005.09629, 2020.

[13] D. Liu, K.-Y. Chen, H.-Y. Lee, and L. s. Lee, “Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings,” in Proc. of Interspeech, 2018.

[14] K.-Y. Chen, C.-P. Tsai, D.-R. Liu, et al., “Completely unsupervised speech recognition by a generative adversarial network harmonized with iteratively refined hidden markov models,” in Proc. of Interspeech, 2019.

[15] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, “Unsupervised speech recognition,” arXiv preprint arXiv:2105.11084, 2021.

[16] H. Gao, J. Ni, Y. Zhang, K. Qian, et al., “Zero-shot cross-lingual phonetic recognition with external language embedding,” in Proc. of Interspeech, 2021.

[17] X. Li, S. Dalmia, J. Li, et al., “Universal phone recognition with a multilingual allophone system,” in Proc. of ICASSP. IEEE, 2020.

[18] C. Jacobs and H. Kamper, “Multilingual transfer of acoustic word embeddings improves when training on languages related to the target zero-resource language,” arXiv preprint arXiv:2106.12834, 2021.

[19] B. Yan, S. Dalmia, D. Mortensen, F. Metze, and S. Watanabe, “Differentiable allophone graphs for language-universal speech recognition,” arXiv preprint arXiv:2107.11628, 2021.

[20] X. Li, S. Dalmia, D. Mortensen, et al., “Towards zero-shot learning for automatic phonemic transcriptions,” in Proc. of AAAI, 2020.

[21] G. Winata, G. Wang, C. Xiong, and S. Hoi, “Adapt-and-adjust: Overcoming the long-tail problem of multilingual speech recognition,” arXiv preprint arXiv:2012.01687, 2020.

[22] T. Schultz, “Globalphone: a multilingual speech and text database developed at karlsruhe university,” in ICSLP, 2002.

[23] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, “Unsupervised cross-lingual representation learning for speech recognition,” arXiv preprint arXiv:2006.13979, 2020.

[24] R. Ardila et al., “Common voice: A massively-multilingual speech corpus,” arXiv preprint arXiv:1912.06670, 2019.

[25] M. Gales, K. M Knill, A. Ragni, and S. Rath, “Speech recognition and keyword spotting for low-resource languages: Babel project research at cued,” in SLTU, 2014.

[26] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, “Mls: A large-scale multilingual dataset for speech research,” arXiv preprint arXiv:2012.03411, 2020.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” arXiv, 2018.

[28] A. Baevski, S. Schneider, and M. Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” in Proc. of ICLR, 2020.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, and et al., “Attention is all you need,” in Proc. of NIPS, 2017.

[30] H. Jegou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” IEEE Trans. Pattern Anal. Mach. Intell., 2011.

[31] Eric Jang, Shixiang Gu, and Ben Poole, “Categorical reparameterization with gumbel-softmax,” arxiv, 2016.

[32] David R. M., Patrick L., et al., “Panphon: A resource for mapping IPA segments to articulatory feature vectors,” in Proc. of COLING. 2016, ACL.

[33] M. Hasegawa-Johnson et al., “Grapheme-to-phoneme transduction for cross-language asr,” in SLSP. Springer, 2020.

[34] M. Ott, S. Edunov, A. Baevski, et al., “fairseq: A fast, extensible toolkit for sequence modeling,” arXiv preprint arXiv:1904.01038, 2019.

[35] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proc. of ICML, 2006.

[36] V. Pratap, A. Hannun, Q. Xu, et al., “Wav2letter+: A fast open-source speech recognition system,” in Proc. of ICASSP. IEEE, 2019.
Table 6. Statistics of languages from CommonVoice and the ones that are supported in Espeak and Phonetisaurus phonemizers. The languages denoted with * are potentially not well supported by Espeak phonemizer, so we manually removed them in either train or test set.

| Dataset | Code   | Lang         | Family       | Split | Hours | Phonemizer            |
|---------|--------|--------------|--------------|-------|-------|-----------------------|
|         |        |              |              | train | valid | test                  |
|         | eo     | esperanto    | Constructed  | train  | 34.0  | 13.3 14.3 | eo espeak     |
|         | lt     | lithuanian   | Baltic       | train  | 1.2   | 0.4 0.7  | lt lithuanian |
|         | cy     | welsh        | Celtic       | train  | 9.1   | 6.8 7.0  | cy            |
|         | ta     | tamil        | Dravidian    | train  | 2.4   | 2.2 2.3  | ta tamil      |
|         | sv-SE  | swedish      | North Germanic | train  | 2.1   | 1.7 1.8  | sv-SE swedish |
|         | de     | german       | West Germanic | train  | 392.7 | 25.0 25.5 | de german     |
|         | en     | english      | West Germanic | train  | 893.5 | 27.2 26.0 | en english    |
|         | as     | assamese     | Indic        | train  | 0.4   | 0.2 0.2  | as assamese   |
|         | hi     | hindi        | Indic        | train  | 0.2   | 0.2 0.2  | hi hindi      |
|         | or     | oriya        | Indic        | train  | 0.6   | 0.2 0.2  | or oriya      |
|         | fa     | persian      | Iranian      | train  | 7.7   | 6.5 7.2  | fa persian    |
|         | ja*    | japanese     | Japonic      | train  | 0.9   | 0.8 0.9  | ja japanese   |
|         | id     | indonesian   | Austronesian | train  | 2.1   | 1.9 2.0  | id indonesian |
|         | ca     | catalan      | Romance      | train  | 441.5 | 24.0 24.9 | ca            |
|         | es     | spanish      | Romance      | train  | 235.1 | 25.0 25.7 | es spanish    |
|         | fr     | french       | Romance      | train  | 424.5 | 24.0 24.5 | fr french     |
|         | pt     | portuguese   | Romance      | train  | 7.8   | 5.6 6.1  | pt portuguese |
|         | ar     | arabic       | Semitic      | train  | 16.0  | 8.8 9.1  | ar arabic     |
|         | zh-CN  | chinese      | Sino-Tibetan | train  | 26.6  | 13.3 14.1 | zh-CN mandarin |
|         | zh-TW  | chinese      | Sino-Tibetan | train  | 3.0   | 2.4 2.6  | zh-TW mandarin |
|         | tr     | turkish      | Turkic       | train  | 2.0   | 1.9 2.1  | tr turkish    |
|         | ky     | kirghiz     | Turkic       | train  | 2.6   | 2.1 1.9  | ky kirghiz   |
|         | et     | estonian     | Uralic       | train  | 5.5   | 4.7 4.6  | et estonian   |
|         | hu     | hungarian    | Uralic       | train  | 4.3   | 1.7 1.9  | hu hungarian  |
|         | ru     | russian      | East Slavic  | train  | 23.5  | 12.3 13.2 | ru russian    |
|         | cs     | czech        | West Slavic  | train  | 7.3   | 5.0 5.0  | cs czech     |
|         | it     | italian      | Romance      | dev    | 86.2  | 21.0 22.1 | it italian    |
|         | eu     | basque       | Language isolate | test  | 10.9  | 7.8 8.2  | eu            |
|         | ia     | interlingua  | Constructed  | test   | 2.2   | 1.5 0.8  | ia            |
|         | lv     | latvian      | Baltic       | test   | 1.9   | 1.6 1.6  | lv latvian   |
|         | ka     | georgian     | South Caucasian | test | 1.6   | 0.9 1.0  | ka georgian  |
|         | nl     | dutch        | West Germanic | test   | 11.5  | 6.4 7.0  | nl dutch     |
|         | el     | greek        | Hellenic     | test   | 2.8   | 1.5 1.8  | el greek     |
|         | ro     | romanian     | Romance      | test   | 3.6   | 1.0 2.0  | ro romanian   |
|         | mt     | maltese      | Semitic      | test   | 2.3   | 1.8 2.1  | mt maltese   |
|         | tt     | tatar        | Turkic       | test   | 11.5  | 2.0 4.4  | tt tatar     |
|         | fi     | finnish      | Uralic       | test   | 0.5   | 0.5 0.6  | fi finnish   |
|         | sl     | slovenian    | South Slavic | test   | 1.9   | 0.5 0.7  | sl slovenian |
|         | pl     | polish       | West Slavic  | test   | 9.3   | 6.6 7.0  | pl polish    |
|         | ga-IE* | irish        | Celtic       | test   | 0.5   | 0.4 0.5  | ga irish     |
|         | zh-HK* | chinese      | Sino-Tibetan | test   | 3.9   | 3.1 3.6  | zh-HK yue    |
|         | pa-IN* | punjabi      | Indic        | test   | 0.2   | 0.1 0.1  | pa punjabi   |

CV
Table 7. Statistics of languages from Babel and the ones that are supported in Espeak and Phonetisaurus phonemizers.

| Dataset | Code | Lang     | Family      | Hours | Phonemizer |
|---------|------|----------|-------------|-------|------------|
|         |      |          |             | train | valid | test     | Espeak | Phonetisaurus |
| Babel   | 307  | Amharic  | Semitic     | 39.4  | 4.4   | 11.7     | am     | amharic.8_2.4.fst |
|         | 103  | Bengali  | Indic       | 56.4  | 6.3   | 10.0     | bn     | bengali.4_3_2.fst  |
|         | 301  | Cebuano  |             | 37.4  | 4.2   | 10.4     | cebuano.4_3_2.fst |
|         | 201  | Haitian  | Creole      | 61.0  | 6.7   | 10.8     | ht     | haitian.8_3_3.fst  |
|         | 402  | Javanese | Austronesian| 41.1  | 4.6   | 11.4     | javanese.4_2_2.fst|
|         | 202  | Swahili  | Bantu       | 40.1  | 4.5   | 10.7     | sw     | swahili.4_2_2.fst  |
|         | 204  | Tamil    | Dravidian   | 62.6  | 7.0   | 11.6     | ta     | tamil.2_3_3.fst   |
|         | 107  | Vietnamese| Austroasiatic| 78.8  | 8.8   | 11.0     | vi     | vietnamese.2_2_2.fst|
|         | 102  | Assamese | Indic       | 54.8  | 6.1   | 10.0     | as     | assamese.4_2_3.fst |
|         | 403  | Dholuo   |             | 37.6  | 4.1   | 10.1     | luo    | luo.4_2_2.fst     |
|         | 305  | Guaraní  | South American Indian | 38.9  | 4.3   | 10.6     | gn     | guaraní.4_2_2.fst |
|         | 306  | Igbo     | Niger–Congo | 39.7  | 4.4   | 10.9     | igbo   | igbo.4_3_4.fst    |
|         | 302  | Kazakh   | Turkic      | 36.1  | 4.0   | 9.8      | kk     | kazakh.2_3_2.fst  |
|         | 104  | Pashto   | Indo-European| 70.7  | 7.8   | 10.0     | pushto | pushto.8_3_2.fst  |
|         | 106  | Tagalog  | Austronesian| 76.2  | 8.6   | 10.7     | tagalog| tagalog.4_2_3.fst |
|         | 303  | Telugu   | Dravidian   | 38.1  | 4.3   | 9.9      | te     | telugu.4_4_4.fst  |
|         | 105  | Turkish  | Turkic      | 70.0  | 7.8   | 9.9      | tr     | turkish.download.fst|
|         | 206  | Zulu     | Niger–Congo | 56.4  | 6.2   | 10.5     | zulu   | zulu.4_2_4.fst    |
|         | 404  | Georgian | South Caucasian| 45.5  | 5.1   | 12.4     | ka     | georgian.4_2_3.fst|
|         | 101  | Cantonese| Sino-Tibetan| 120.3 | 13.5  | 17.0     | yue    | yue.2_2_4.fst     |
|         | 203  | Lao      | Tai–Kadai  | 59.2  | 6.5   | 10.6     | luo    | luo.2_2_2.fst     |

Table 8. Comparison of PER on the test set of a subset of Common Voice languages. tr2tgt lexicon is used in beam-search decoding by default, while tgt2tr lexicon is used only for the columns denoted with *. The numbers in the parenthesis next to each pre-trained model is the maximum number of hours per language in the training set.

| Pretrain Phonemizer | - (10) | - (200) | EN - LV (10) | XLSR - 53 (10) | XLSR - 53 (10) |
|---------------------|--------|---------|--------------|----------------|----------------|
|                     | Espeak| n-gram  | Espeak       | Espeak         | Espeak         |
|                     | viterbi| n-gram  | viterbi      | viterbi        | viterbi        |
| it                  | 56.6  | 47.5    | 50.1         | 41.8           | 31.8           |
| eu                  | 51.2  | 45.6    | 39.7         | 32.1           | 24.8           |
| ia                  | 38.9  | 27.8    | 30.9         | 23.0           | 12.7           |
| lv                  | 65.2  | 59.8    | 62.7         | 56.5           | 41.9           |
| ka                  | 61.8  | 56.1    | 54.6         | 48.9           | 29.1           |
| nl                  | 66.3  | 56.8    | 63.0         | 56.1           | 46.8           |
| el                  | 49.5  | 40.6    | 42.1         | 33.7           | 18.9           |
| ro                  | 45.7  | 34.7    | 46.7         | 36.9           | 21.3           |
| mt                  | 66.3  | 60.2    | 62.0         | 56.0           | 47.4           |
| tt                  | 68.2  | 63.9    | 65.1         | 60.8           | 46.2           |
| fi                  | 58.8  | 55.6    | 53.8         | 48.3           | 36.8           |
| sl                  | 62.9  | 56.0    | 60.5         | 54.6           | 43.5           |
| pl                  | 62.3  | 59.3    | 60.2         | 56.0           | 36.1           |
| Avg                 | 58.0  | 51.1    | 53.2         | 46.5           | 33.6           | 23.9 |
|                     | 31.4  | 22.2    | 22.3         | 39.5           | 31.7           | 32.4 |
A. DATASET DETAILS

In this section, we summarize the details of CommonVoice and BABEL datasets. Specifically we list the code and name of each language together with the family they belong to. We also show the duration in hours of each split of each language. The amount of training data varies a lot in CommonVoice dataset. We subsample the training data for high resource languages included in the training set. Cantonese and Lao are the unseen languages. BB and CV represents BABEL and CommonVoice dataset and the following numbers are the number of the languages and it is good for zero-shot transfer learning. Besides, Vietnamese (vi) and Chinese family (zh-CN, zh-TW, zh-HK) also scatter around the whole plot, meaning that IPA phoneme symbols are commonly shared across different languages and it is good for zero-shot transfer learning. Besides, Vietnamese (vi) and Chinese family (zh-CN, zh-TW, zh-HK) seems isolated to others, as their phoneme symbols include tones. Specifically, vowels like ‘ou’ can be denoted as one of the following: ‘ou1’, ‘ou2’, ‘ou3’, ‘ou4’, ‘ou5’ and ‘ou6’. They are intrinsically both hard to learn and hard to predict.

C. FULL COMPARISON ON COMMONVOICE

We summarize all the results on CommonVoice in Table 8. Apart from the analysis in the ablation section, we can also find that beam-search decoding consistently helps to improve the model performance for all languages in all the settings. The results in Table 8 shows that the trend of accuracy on unseen languages is similar across phonemizers.

D. FULL RESULTS ON BABEL

As shown in Table 9 with finetuning on only the BABEL subset of 16’s training data, our method performs better on the supervised languages already, indicating that the wav2vec Transformer blocks, that are not included in 16, benefit the model learning a lot. Additionally, models trained with mixed CommonVoice and BABEL data generalize better than the ones trained on either one of them on the unseen languages. It also surpasses 16 with using an extra learned language encoder.

Table 9. 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. BB and CV represents BABEL and CommonVoice dataset and the following numbers are the number of the languages included in the training set.

| BB Data | BB-6[16] | BB-6 | BB-19 | - | BB-19 |
|---------|----------|------|-------|---|-------|
| Other Data | CGN+GP | - | - | CV-21 | CV-21 |
| # hours / lang | all | all | all | 10 | 10 |
| # hours total | 1,492 | 317 | 935 | 118 | 298 |
| 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 |
| Cantonese | 73.1 | 73.6 | 72.6 | 70.9 | **63.6** |
| Lao | 69.3 | 70.3 | 70.2 | 72.1 | **63.7** |

Fig. 1. Correlation between each pair of languages in CommonVoice dataset.

B. LANGUAGE CORRELATION

We simply denote the correlation between each pair of languages by \( \text{cor}(l_1, l_2) = \frac{|\text{vocab}(l_1) \cup \text{vocab}(l_2)|}{|\text{vocab}(l_1)|} \), where \( l_1 \) and \( l_2 \) are two languages and \( \text{vocab}() \) denotes the phoneme vocabulary of a given language. Figure 1 shows the correlations between pairs of CommonVoice languages. Since languages are ordered purely by family, it is reasonable to see high correlations on the diagonal blocks. However, this high correlation also scatter around the whole plot, meaning that IPA phoneme symbols are commonly shared across different languages and it is good for zero-shot transfer learning. Besides, Vietnamese (vi) and Chinese family (zh-CN, zh-TW, zh-HK) seems isolated to others, as their phoneme symbols include tones. Specifically, vowels like ‘ou’ can be denoted as one of the following: ‘ou1’, ‘ou2’, ‘ou3’, ‘ou4’, ‘ou5’ and ‘ou6’. They are intrinsically both hard to learn and hard to predict.

https://github.com/uiuc-sst/g2ps

---

4https://github.com/uiuc-sst/g2ps