Simple and Effective Zero-shot Cross-lingual Phoneme Recognition

Qiantong Xu, Alexei Baevski, Michael Auli*

Facebook AI Research

{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 predict 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 $\mathbf{z}_1, \ldots, \mathbf{z}_T$ which are input to a Transformer $g : \mathcal{Z} \mapsto \mathcal{C}$ to output context representations $\mathbf{c}_1, \ldots, \mathbf{c}_T$ [28, 29]. Each $\mathbf{z}_t$ represents about 25ms of audio strided by 20ms and the Transformer architecture follows BERT [30, 28].

During training, feature encoder representations are discretized to $\mathbf{q}_1, \ldots, \mathbf{q}_T$ 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 form the codebooks and the chosen entries are concatenated to obtain $\mathbf{q}$ [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 $\mathbf{q}_t$ for a masked time-step within a set of $K = 100$ distractors $\mathbf{Q}_t$ sampled from other masked time steps:

$$-\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t))}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}}))}$$

where $\mathbf{c}_t$ is the output of the Transformer, and $sim(\mathbf{a}, \mathbf{b})$ 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).

* Now at Sambanova Systems

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 vectors[1], 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 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 [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.

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

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

Split	Languages
CommonVoice (CV)	
train	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)
dev	Italian (it)
test	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)
BABEL (BB)	
train	**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)**
dev	CV-Italian (it)
test	Cantonese (yue), Lao (lo)

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 [35] 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 [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

[2]https://github.com/espeak-ng/espeak-ng

[3]https://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).*

		Gao et al. [16]	This work			
	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
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.*

	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

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.*

Model	No pretrain		No pretrain		w2v LV-60K		XLSR-53	
# hours / language	10		200		10		10	
# hours in total	149		1156		149		149	
	viterbi	n-gram	viterbi	n-gram	viterbi	n-gram	viterbi	n-gram
it	56.6	47.5	50.1	41.8	31.8	16.9	26.0	**13.9**
eu	51.2	45.6	39.7	32.1	24.8	16.3	20.8	**13.7**
ia	38.9	27.8	30.9	23.0	12.7	6.7	10.7	**6.1**
lv	65.2	59.8	62.7	56.5	41.9	33.5	39.9	**32.3**
ka	61.8	56.1	54.6	48.9	29.1	24.0	30.5	**23.8**
nl	66.3	56.8	63.0	56.1	46.8	30.5	37.1	**19.8**
el	49.5	40.6	42.1	33.7	18.9	10.7	17.3	**10.4**
ro	45.7	34.7	46.7	36.9	21.3	15.0	20.1	**14.8**
mt	66.3	60.2	62.0	56.0	47.4	36.1	46.6	**35.9**
tt	68.2	63.9	65.1	60.8	46.2	**34.7**	48.4	37.4
fi	58.8	55.6	53.8	48.3	36.8	29.9	36.8	**29.0**
sl	62.9	56.0	60.5	54.6	43.5	29.0	40.6	**26.1**
pl	62.3	59.3	60.2	56.0	36.1	27.3	32.8	**25.7**
Avg	58.0	51.1	53.2	46.5	33.6	23.9	31.4	**22.2**

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	tr2tgt	tgt2tr
Avg	24.5	24.6	31.7	32.4

well as the cross-lingually pretrained XLSR-53 model, trained on 53 different languages.

Table 4 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 (from 10h to 200h per language). This is in line 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.

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 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. 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.

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] A. 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] Q. X., A. 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 transcription," in *Proc. of AAAI*, 2020.

[21] P. Zelasko, S. Feng, L. M. Velazquez, A. Abavisani, S. Bhati, O. Scharenborg, M. Hasegawa-Johnson, and N. Dehak, "Discovering phonetic inventories with crosslingual automatic speech recognition," *arXiv*, vol. abs/2201.11207, 2022.

[22] 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.

[23] T. Schultz, "Globalphone: a multilingual speech and text database developed at karlsruhe university," in *ICSLP*, 2002.

[24] 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.

[25] R. Ardila *et al.*, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[26] 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.

[27] 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.

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

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

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

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

[32] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv*, 2016.

[33] S. Dieleman, A. v. d. Oord, and K. Simonyan, "The challenge of realistic music generation: modelling raw audio at scale," *arXiv*, 2018.

[34] D. R. M., P. L. *et al.*, "Panphon: A resource for mapping IPA segments to articulatory feature vectors," in *Proc. of COLING*. ACL, 2016.

[35] M. Hasegawa-Johnson *et al.*, "Grapheme-to-phoneme transduction for cross-language asr," in *SLSP*. Springer, 2020.

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

[37] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. of ICML*, 2006.

[38] V. Pratap, A. Hannun, Q. Xu *et al.*, "Wav2letter++: A fast open-source speech recognition system," in *Proc. of ICASSP*. IEEE, 2019.

[39] J. Kahn *et al.*, "Libri-light: A benchmark for asr with limited or no supervision," in *Proc. of ICASSP*. IEEE, 2020.

Appendices

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	Espeak	Phonetisaurus
CV	eo	esperanto	Constructed	train	34.0	13.3	14.3	eo	
	lt	lithuanian	Baltic	train	1.2	0.4	0.7	lt	lithuanian_4_2_2.fst
	cy	welsh	Celtic	train	9.1	6.8	7.0	cy	
	ta	tamil	Dravidian	train	2.4	2.2	2.3	ta	tamil_2_3_3.fst
	sv-SE	swedish	North Germanic	train	2.1	1.7	1.8	sv-SE	swedish_4_4_4.fst
	de	german	West Germanic	train	392.7	25.0	25.5	de	german_download.fst
	en	english	West Germanic	train	893.5	27.2	26.0	en	english_4_2_2.fst
	as	assamese	Indic	train	0.4	0.2	0.2	as	assamese_4_2_3.fst
	hi	hindi	Indic	train	0.2	0.2	0.2	hi	hindi_4_2_2.fst
	or	oriya	Indic	train	0.6	0.2	0.2	or	
	fa	persian	Iranian	train	7.7	6.5	7.2	fa	persian_2_2_2.fst
	ja∗	japanese	Japonic	train	0.9	0.8	0.9	ja	japanese_4_4_4.fst
	id	indonesian	Austronesian	train	2.1	1.9	2.0	id	indonesian_2_4_4.fst
	ca	catalan	Romance	train	441.5	24.0	24.9	ca	
	es	spanish	Romance	train	235.1	25.0	25.7	es	spanish_4_3_2.fst
	fr	french	Romance	train	424.5	24.0	25.1	fr	french_8_4_3.fst
	pt	portuguese	Romance	train	7.8	5.6	6.1	pt	portuguese_download.fst
	ar	arabic	Semitic	train	16.0	8.8	9.1	ar	arabic_download.fst
	zh-CN	chinese	Sino-Tibetan	train	26.6	13.3	14.1	cmn	mandarin_2_4_4.fst
	zh-TW	chinese	Sino-Tibetan	train	3.0	2.4	2.6	cmn	mandarin_2_4_4.fst
	tr	turkish	Turkic	train	2.0	1.9	2.1	tr	turkish_download.fst
	ky	kirghiz	Turkic	train	2.6	2.1	1.9	ky	kirghiz_8_2_2.fst
	et	estonian	Uralic	train	5.5	4.7	4.6	et	estonian_2_4_4.fst
	hu	hungarian	Uralic	train	4.3	1.7	1.9	hu	hungarian_2_4_2.fst
	ru	russian	East Slavic	train	23.5	12.3	13.2	ru	russian_download.fst
	cs	czech	West Slavic	train	7.3	5.0	5.0	cs	czech_4_4_4.fst
	it	italian	Romance	dev	86.2	21.0	22.1	it	italian_8_2_3.fst
	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_2_4_4.fst
	ka	georgian	South Caucasian	test	1.6	0.9	1.0	ka	georgian_4_2_3.fst
	nl	dutch	West Germanic	test	11.5	6.4	7.0	nl	dutch_download.fst
	el	greek	Hellenic	test	2.8	1.5	1.8	el	greek_2_2_2.fst
	ro	romanian	Romance	test	3.6	1.0	2.0	ro	romanian_2_3_3.fst
	mt	maltese	Semitic	test	2.3	1.8	2.1	mt	maltese_2_4_4.fst
	tt	tatar	Turkic	test	11.5	2.0	4.4	tt	tatar_2_2_2.fst
	fi	finnish	Uralic	test	0.5	0.5	0.6	fi	finnish_2_4_4.fst
	sl	slovenian	South Slavic	test	1.9	0.5	0.7	sl	slovenian_2_4_4.fst
	pl	polish	West Slavic	test	9.3	6.6	7.0	pl	polish_2_2_2.fst
	ga-IE∗	irish	Celtic	test	0.5	0.4	0.5	ga	
	zh-HK∗	chinese	Sino-Tibetan	test	3.9	3.1	3.6	yue	yue_2_2_4.fst
	pa-IN∗	punjabi	Indic	test	0.2	0.1	0.1	pa	panjabi_4_4_4.fst

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_4_2_2.fst
	305	Guarani	South American Indian	38.9	4.3	10.6	gn	guarani_4_2_2.fst
	306	Igbo	Niger–Congo	39.7	4.4	10.9		igbo_2_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_8_3_2.fst
	106	Tagalog	Austronesian	76.2	8.6	10.7		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_4_4_3.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		lao_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	No pretrain (10)		No pretrain (200)		w2v LV-60K (10)		XLSR - 53 (10)			XLSR - 53 (10)		
Phonemizer	Espeak		Espeak		Espeak		Espeak			Phonetisaurus		
	viterbi	n-gram	viterbi	n-gram	viterbi	n-gram	viterbi	n-gram	n-gram*	viterbi	n-gram	n-gram*
it	56.6	47.5	50.1	41.8	31.8	16.9	26.0	**13.9**	14.3	26.6	18.1	17.8
eu	51.2	45.6	39.7	32.1	24.8	16.3	20.8	13.7	**12.2**	-	-	-
ia	38.9	27.8	30.9	23.0	12.7	6.7	10.7	6.1	**6.0**	-	-	-
lv	65.2	59.8	62.7	56.5	41.9	33.5	39.9	**32.3**	34.0	50.0	40.5	62.4
ka	61.8	56.1	54.6	48.9	29.1	24.0	30.5	**23.8**	24.3	34.7	26.6	25.5
nl	66.3	56.8	63.0	56.1	46.8	30.5	37.1	**19.8**	22.7	37.3	24.4	27.8
el	49.5	40.6	42.1	33.7	18.9	10.7	17.3	10.4	**9.9**	36.2	32.2	22.9
ro	45.7	34.7	46.7	36.9	21.3	15.0	20.1	14.8	**12.6**	28.5	16.2	17.5
mt	66.3	60.2	62.0	56.0	47.4	36.1	46.6	**35.9**	36.1	43.3	34.7	37.5
tt	68.2	63.9	65.1	60.8	46.2	34.7	48.4	37.4	**35.5**	49.1	45.6	35.3
fi	58.8	55.6	53.8	48.3	36.8	29.9	36.8	29.0	**27.1**	43.1	34.0	37.2
sl	62.9	56.0	60.5	54.6	43.5	29.0	40.6	**26.1**	27.4	33.4	26.1	23.9
pl	62.3	59.3	60.2	56.0	36.1	27.3	32.8	**25.7**	27.1	51.8	49.9	48.1
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

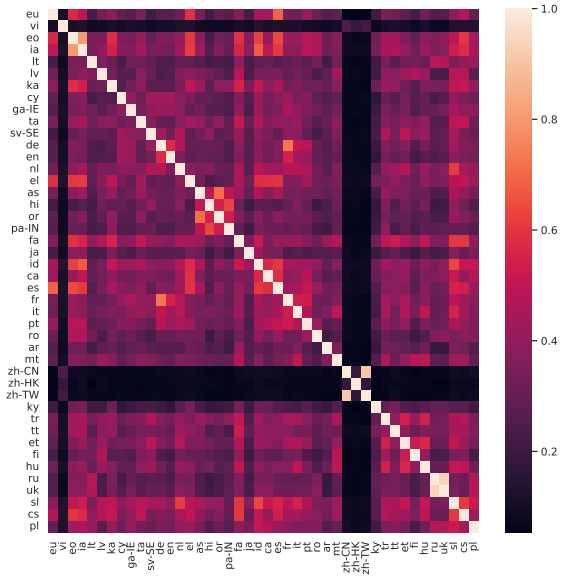

Figure 1: *Correlation between each pair of languages in CommonVoice dataset.*

A. Dataset Details

In this section, we summarize the details of CommonVoice and BABEL datsets. 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 to avoid bias. Additionally, for each language, we also provide the specific language identifier we used in Espeak and the specific finite state transducers[4] in Phonetisaurus. We can see that Espeak covers more languages in CommonVoice, while Phonetisaurus covers more languages in BABEL.

The languages in Table 6 are ordered first by splits (training, validation and test) and then they are grouped by families.

B. Language correlation

We simply denote the correlation between each pair of languages by $cor(l_1, l_2) = \frac{|vocab(l_1) \cup vocab(l_2)|}{|vocab(l_1) \cup vocab(l_2)|}$, where l_1 and l_2 are two languages and $vocab(\cdot)$ 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 faimily (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.

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

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**

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 5 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.