Unsupervised pretraining transfers well across languages

Morgane Rivière, Armand Joulin, Pierre-Emmanuel Mazaré, Emmanuel Dupoux

UNSUPERVISED PRETRAINING TRANSFERS WELL ACROSS LANGUAGES

Morgane Rivière[†], *Armand Joulin*[†], *Pierre-Emmanuel Mazaré*[†], *Emmanuel Dupoux*[†‡*]

†Facebook AI Research, ‡Ecole des Hautes Etudes en Sciences Sociales

ABSTRACT

Cross-lingual and multi-lingual training of Automatic Speech Recognition (ASR) has been extensively investigated in the supervised setting. This assumes the existence of a parallel corpus of speech and orthographic transcriptions. Recently, contrastive predictive coding (CPC) algorithms have been proposed to pretrain ASR systems with unlabelled data. In this work, we investigate whether unsupervised pretraining transfers well across languages. We show that a slight modification of the CPC pretraining extracts features that transfer well to other languages, being on par or even outperforming supervised pretraining. This shows the potential of unsupervised methods for languages with few linguistic resources.

Index Terms— Unsupervised pretraining, low resources, cross-lingual

1. INTRODUCTION

Learning phoneme representations remains a challenge for a large number of languages with limited supervised resources. A common approach is to pre-train these representations on a large supervised corpus in other languages and transfer them to the low resource languages [1, 2]. For example, Vesely et al. [3] learn a shared representation on a supervised multilingual dataset and finetune it on the target language. This pretraining works even between distant languages, but requires massive supervised corpora in the same domain.

Recently, several works [4, 5] have proposed promising methods to train monolingual audio representations without supervision. In particular, Schneider et al. [6] shows that the unsupervised pre-training method of van den Oord [4] improves the quality of automatic speech recognition (ASR) on several competitive benchmarks. In this paper, we are interested to see if similar unsupervised pre-training methods can be leveraged in a cross-lingual setting to improve the quality of phoneme representations for low resource languages.

We focus on the contrastive predictive coding (CPC) method of van den Oord [4] since Schneider et al. [6] has shown its benefit for pre-training features for ASR. CPC is a form of forward modeling in the feature space [7]: it predicts the near future windows in an audio sequence while contrasting with windows from other sequences or more distant in time. We introduce several modifications to the original approach stabilize the training and lead to better phoneme representations. We use our modified CPC model to pre-train phoneme representations in English, namely on Librispeech, and transfer them to several low-resource languages from the Common Voice database.

In this paper, we obtain several results related to transferring across languages the features pre-trained without supervision. First, pre-training phoneme representation outperforms representations trained from scratch in the target language, even if we do not use any supervision for the pre-training. Suprisingly, we also observe that the gap between unsupervised and supervised pre-training is relatively small if we use the same pre-training corpora. Finally, scaling unsupervised pre-training to larger unlabelled datasets further reduces the gap with the supervised pre-training features, and even surpasses it in some low-resource languages.

2. RELATED WORK

2.1. Multilingual pre-training for speech recognition

A common way to improving speech recognition in low-resource languages is to train multilingual speech recognition with shared components [8, 9, 2]. For example, Stolcke et al. [9] train features for phoneme classification in a different language. Burget et al. [10] shares the parameters of a Gaussian Mixture Model. Closer to our work, several works have shared the parameters of a neural network encoder, using feedforward networks [3, 1, 2] or LSTM [11]. The model is then finetuned on the target low-resource language to fit its specificities [12]. The sampling of the languages during the pre-training can focus on languages related to the targeted language [11]. Another approach is to encourage a language-independent encoder with an adversarial loss [13]. As opposed to our work, this line of research focuses on supervised pre-training which restrict its impact to domains or languages with large resources for supervision.

2.2. Unsupervised learning of features

Many unsupervised learning approaches have been proposed for speech and we focus on those based on contrastive learning [7, 14, 15]. In particular, Time Contrastive Learning [5]

* Code and data available in `https://github.com/facebookresearch/CPC_audio`. This is the extended reprint of: Rivière, M., Joulin, A., Mazaré, P.E. and Dupoux, E. (2020). Unsupervised pretraining transfers well across languages. *in ICASSP-2020.*

learns audio features by discriminating between time windows. Our work closely follows van den Oord et al. [4] where a contrastive loss is used to predict forward representations in an audio sequence. Their Contrastive Predictive Coding (CPC) objective function is similar to the objective of word2vec [16], applied to sequences instead of words. Contrastive approaches are also related to examplar self-supervision [17, 18, 19]. However, CPC has the advantage of making no assumption about the nature or number of the training data samples. Recently, variants of CPC have been applied to monolingual ASR [6] and images [20].

3. APPROACH

In this section, we rapidly introduce the approach of van den Oord et al. [4] and we refer the reader to the original paper for details. We also present several modifications to improve the resulting representations and stabilize the training. We made our code as well as our experiments available to the public[1].

3.1. Contrastive Predictive Coding

Unsupervised training of neural networks relies on building a pretext task that requires discriminative features to be solved. The pretext task used in Contrastive Predictive Coding (CPC) [4] is forward modeling, i.e., predicting the future states of a sequence from its past. The particularity of CPC is to frame forward modeling as a reconstruction of future representations, not future inputs. Past and future representations are built from the same model, and a contrastive loss ensures that temporally nearby representations are pushed closer than temporally distant ones.

More precisely, given an audio sequence splitted in T discrete time steps, or windows, we embed the input signal $\mathbf{x}_t$ at each time step t with a encoder. Then, we form the current phoneme representation $\mathbf{z}_t$ by applying a sequence model to the resulting sequence of t embeddings, i.e.,

$$\mathbf{z}_t = \psi_\rho(\phi_\theta(\mathbf{x}_1), \dots, \phi_\theta(\mathbf{x}_t)),$$

where ϕ_θ is the encoder and ψ_ρ is the sequence model, parametrized by θ and ρ respectively. In CPC, the encoder is a 5-layer convolutional network (kernel sizes: 10,8,4,4,4, stride sizes: 5,4,2,2,2) and the sequence model is a 1-layer Gated Recurrent Units (GRU).The encoder also has a downsampling factor of 160, meaning that for a 16kHz input, each feature encodes 10ms of audio.

Given this phoneme embedding $\mathbf{z}_t$, the pretext task in CPC is to predict the next K future representations, i.e., $\phi_\theta(\mathbf{x}_{t+k})$ for $k \in \{1, \dots, K\}$. CPC also pushes away representations from a random subset $\mathcal{N}_t$ of negative examples, or "distant" windows. Overall, the loss function at a time step t is thus:

$$\mathcal{L}_t = -\frac{1}{K}\sum_{k=1}^{K} \log\left[\frac{\exp\left(\phi_\theta(\mathbf{x}_{t+k})^\top \mathbf{A}_k \mathbf{z}_t\right)}{\sum_{\mathbf{n}\in\mathcal{N}_t}\exp\left(\phi_\theta(\mathbf{n})^\top \mathbf{A}_k \mathbf{z}_t\right)}\right]. \quad (1)$$

[1] https://github.com/facebookresearch/CPC_audio

where $\mathbf{A}_k$ is a linear classifier. There are many ways to sample the "distant" windows and we follow van den Oord et al. [4] by sampling negative *within speaker*. The parameters θ, ρ and $\mathbf{A}_{1,\dots,K}$ are learned with stochastic gradient descent.

3.2. Modifications to Contrastive Predictive Coding

3.2.1. Stabilization of the training

We observe empirically that the training of CPC is unstable, and can converge to poor solutions. The reason is the presence of batch normalization [21] between the layers of the encoder. Indeed, batch normalization parameters are learned by computing statistics over the whole batch. Since the encoder is shared across a sequence, these parameters leak information between past and future windows. This makes minimizing eq. (1) trivial when the batch normalization is activated, resulting in instability. We fix this issue by replacing batch normalization with a channel-wise normalization that plays a similar role of conditioning internal representations. As oppposed to batch normalization, the parameters are not shared across the sequence and do not leak information (see Supplementary Section S1.1 for details).

3.2.2. Improving the model

The prediction of future representations is made by linear classifiers on top of a phoneme embedding, as shown in eq. (1). The motivation is to encourage the phoneme embeddings to encode linearly separable phonemes. However, the future representations are not phoneme representations themselves; they are embeddings of the time window. Comparing the outputs of a sequence model and an encoder with a linear classifier may not result in linearly separable phoneme representations. Several alternatives are possible, such as adding a sequence model on the future representations. In practice, we find that replacing each linear classifier with a 1-layer Transformer network [22] works well (see Supplementary Section S1.2 for details). This layer accesses the entire sequence of $z_1, ..z_t$ to predict a particular $\phi(x_{t+k})$. We also observe that reducing the dimension of convolutionnal layers from 512 to 256 does not impact performance while reducing memory footprint. Finally, using an LSTM instead of a GRU slightly improves the performance.

3.3. Transferring features to phoneme classification

In this work, we evaluate the quality of phoneme representations trained with no supervision when transferred across languages. Standard cross-lingual approaches finetune their pre-trained network on the targeted language. While this improves the quality of the resulting representations, it does not assess the quality of the pre-trained representations. Instead, we freeze the model after the pre-training and simply learn a linear classifier for the targeted language. Specifically, we perform the linear classification of a concatenation of 8 windows to match the average size of a phoneme. We then use the CTC loss between our model predictions and the non-aligned

phoneme transcriptions [23]. This procedure explicitly measures the linear separability of the phoneme representation, once transferred to a target language.

4. EXPERIMENTAL SETTING

4.0.1. Pre-training on the Librispeech dataset

We pre-train models on the English Librispeech dataset (LS). We consider both the 100h and 360h splits of clean data. For the supervised pre-training model, we use the aligned phone labels provided by [4] for Librispeech-100h.

4.0.2. Transferring to the Common Voice database

After the pre-training, we freeze the parameters of our models and transfer the features across languages. We consider the common Voice database[2] as it comes in many languages. We retrieve the non-aligned phoneme transcription of each audio sample by running the open-source tool phonemizer[3] on their corresponding text scripts. We split our dataset between train, validation and test sets along speakers to reduce the influence of speakers on the performance of phoneme predictions. We consider two train sets of either 1 or 5 hours. We will open source our train-test splits along with our code.

4.0.3. Measuring phoneme separability on Zerospeech2017

Zerospeech2017 is a dataset made to measure phoneme separability of unsupervised models in different languages. We consider the English, Mandarin and French benchmarks and we report the ABX score on them [24]. The ABX score measures the discriminability between phonemes by estimating the probability speech segments to be closer to one another if they encode the same phoneme than if they don't (the distance being DTW-realigned average frame-wise cosine).

5. RESULTS

5.1. Within-language results

In this set of experiments, we compare the original CPC with our modified version on two *within-language* tasks: phoneme discriminability on the English Zerospeech2017 dataset, and phoneme linear separability on Librispeech 100h [4]. In Table 1, we compare our ABX score with that of the toplines from the Zerospeech leaderboards. It is interesting to note that CPC does not perform well on this metric but our modified version is on par with the state of the art. Overall, our modified CPC surpasses the original model on phoneme classification and even matches unsupervised approaches dedicated to phoneme separability. In Table 2, we show that our modifications to CPC leads to an improvement of 3.4 points in phoneme classification compared to the original CPC implementation.

[2] https://voice.mozilla.org

[3] https://gitlab.coml.lscp.ens.fr/mbernard/phonemizer

	Across	Within
Trained on ZeroSpeech2017 (45 h)		
Supervised topline [25]	6.9	5.3
Heck et al. [26]	8.7	6.2
Chorowski et al. [27]	8.0	5.5
Trained on Librispeech-360		
CPC [4]	13.0	9.6
Modified CPC	8.5	6.5

Table 1. **Phoneme discriminability within languages.** Within- and across-speakers ABX scores for the English Zerospeech2017 test set. We compare CPC and modified CPC trained on Librispeech-360 to the best performing models.

	Phone accuracy
Supervised topline	76.3
CPC [4]	65.5
Modified CPC	68.9

Table 2. **Phone classification within language.** Accuracy on the English LibriSpeech-100h dataset for a linear classifier trained on top of frozen features obtained with the original and our modified CPC model.

5.2. Cross-lingual transfer of phoneme features

In a first experiment, we consider the problem of phoneme classification across languages on the Common Voice database. In Table 3, we report the phone error rate (PER) for the linear classifiers trained on top of the phoneme features pretrained with and without supervision. We also compare with a model trained from scratch on the target dataset. The training set of each target dataset is only 1 hour long. The model trained from scratch thus performs poorly. On the other hand, pretrained features significantly improve the performance in all languages, even without any finetuning. First, on 100 hours of librispeech, our modified CPC outperforms the original CPC by 5.4 points on average. However, supervised pretraining still performs slightly better (1.3 points) than our unsupervised pre-training on the same corpus. An advantage of unsupervised pre-training is that we can apply it to any larger unannotated dataset. We show the benefits of this by pre-training our modified CPC on 360 hours of unlabelled data from Librispeech and match the performance of the supervised model. This result not only confirms the findings of [6] but it also shows that unsupervised pre-training can match supervised pre-training with enough data (see Supplementary Section S2 with the larger Libri-light dataset [29]).

In a second experiment, we compare the quality of our pre-trained features against other unsupervised methods on the Zerospeech2017. In Table 4, we compare on French and Mandarin, the ABX score of our approach trained on English

Model	Pretraining	Frozen	du	es	fr	it	ky	ru	sv	tr	tt	zh	Avg
From scratch	-	No	84.7	95.9	95.1	95.0	81.5	97.7	86.1	83.1	72.9	84.3	87.6
Bottleneck [28]	Babel-1070h	Yes	47.9	36.6	48.3	39.0	38.7	45.2	52.6	43.4	42.5	54.3	44.9
Supervised	LS-100h	Yes	42.4	36.4	47.0	40.5	41.0	43.6	47.0	48.5	41.5	56.8	**44.5**
CPC [4]	LS-100h	Yes	51.5	44.2	54.5	47.0	44.8	49.0	54.0	54.7	48.9	60.1	50.9
Modified CPC	LS-100h	Yes	44.4	38.7	49.3	42.1	40.7	45.2	48.8	49.7	44.0	55.5	45.8
Modified CPC	LS-360h	Yes	42.5	38.0	47.1	40.5	41.2	43.7	47.5	47.3	42.0	55.0	**44.5**

Table 3. **Transfer of pre-trained phoneme features across languages.** We pre-train the features on 100h and 360h of Librispeech with supervision ("Supervised") or not ("CPC" and "Modified CPC"). We also include multilingual bottleneck features ("Bottleneck") pre-trained on 1070h from the Babel dataset. We train a linear classifier on the frozen features using 1h of speech from the Common Voice database in different languages. We also report a supervised model trained entirely from scratch on the 1h of speech. We report Phone Error Rate. The languages are: Dutch (du), Spanish (es), French (fr), Italian (it), Kyrgyz (ky), Russian (ru), Sweedish (sv), Turkish (tr), Tatar (tt) and Mandarin (zh).

	French		Mandarin	
	A.	W.	A.	W.
Trained within language				
Supervised topline	9.1	6.8	5.7	4.2
Heck et al. [26]	11.7	8.7	7.4	7.9
Chorowski et al. [27]	10.8	7.5	11.2	10.7
Trained on English (Librispeech-360)				
CPC [4]	18.0	12.3	11.5	10.0
Modified CPC	14.6	10.0	9.5	8.9

Table 4. **Phoneme discriminability of unsupervised features across languages.** Across- ("A.") and within-speakers ("W.") ABX scores on French and Mandarin speech for CPC features pre-trained in English. For comparison: the best systems plus supervised topline of the Zerospeech leaderboard trained within-language.

Librispeech with unsupervised methods trained for these languages. Surprisingly, our English features transfered to other languages are competitive with the top lines of the leaderboard. This result further shows that unsupervised pre-trained features generalize well across languages.

5.2.1. Impact of finetuning phoneme features

We also study the impact of fine-tuning the phoneme features instead of freezing them. We use 5 hours of speech in 5 target languages for this experiment. In Table 5, we compare the difference between frozen features and fine-tuning. As for the experiments on 1h of speech, our approach is on par with supervised pre-training when the features are frozen. We also observe a boost around 7 performance points for all the pre-training methods when we fine-tune the features. Our approach is still relatively competitive with supervised pre-training, but slightly worse (-1.5 points) on average.

Model	pretraining	frozen	finetune
From scratch	-	-	38.3
Supervised	LS-100	37.6	**29.2**
CPC [4]	LS-100	43.5	33.3
Mod. CPC	LS-100	38.8	31.0
Mod. CPC	LS-360	**37.2**	30.7

Table 5. **Comparison between frozen and fine-tuned features.** PER averaged over 5 languages (Spanish, French, Italian, Russian and Tatar). The training set for each language contains 5 hours extracted from the Common Voice database.

6. CONCLUSION

Pre-training in a given language, with or without supervision, can produce features usable across other languages and other domains. Moreover, these features can be matched with a set of phonemes even with extremely low resources datasets and unaligned labels. They are usable with a very simple linear model and can be trained at low cost. Finally, though supervised pre-training tends to be better than the unsupervised one, the gap between them is small and can be greatly reduced with the use of a larger amount of unlabelled data. We did not attempt to push numbers in order to achieve good phone error rates in the low resource languages, as we only tested a linear separation layer for phoneme classification. Further work needs to be done to establish how these pretrained features can be best used in the low resource setting (see [30]), and with other ASR tasks [29].

7. REFERENCES

[1] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. A. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *ICASSP*, 2013.

[2] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *ICASSP*, 2013.

[3] K. Veselỳ, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *SLT*, 2012.

[4] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv:1807.03748*, 2018.

[5] A. Hyvarinen and H. Morioka, "Unsupervised feature extraction by time-contrastive learning and nonlinear ica," in *NIPS*, 2016.

[6] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv:1904.05862*, 2019.

[7] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*, 2005.

[8] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," *Speech Communication*, 2001.

[9] A. Stolcke, F. Grezl, M.-Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in *ICASSP*, 2006.

[10] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, and D. Povey, "Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models," in *ICASSP*, 2010.

[11] X. Li, S. Dalmia, A. W. Black, and F. Metze, "Multilingual speech recognition with corpus relatedness sampling," *arXiv:1908.01060*, 2019.

[12] S. Dalmia, R. Sanabria, F. Metze, and A. W. Black, "Sequence-based multi-lingual low resource speech recognition," in *ICASSP*, 2018.

[13] O. Adams, M. Wiesner, S. Watanabe, and D. Yarowsky, "Massively multilingual adversarial speech recognition," *arXiv:1904.02210*, 2019.

[14] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *JMLR*, 2009.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015.

[16] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[17] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *NIPS*, 2014.

[18] P. Bojanowski and A. Joulin, "Unsupervised learning by predicting noise," in *ICML*, 2017.

[19] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," *arXiv:1808.06670*, 2018.

[20] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," *arXiv:1906.05849*, 2019.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

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

[24] T. Schatz, V. Peddinti, F. Bach, A. Jansen, H. Hermansky, and E. Dupoux, "Evaluating speech features with the minimal-pair abx task: Analysis of the classical mfc/plp pipeline," *INTERSPEECH*, 2013.

[25] E. Dunbar, X. Cao, J. Benjumea, J. Karadayi, M. Bernard, L.t Besacier, X. Anguera, and E. Dupoux, "The zero resource speech challenge 2017," *arXiv:1712.04313*.

[26] M. Heck, S. Sakti, and S. Nakamura, "Feature optimized dpgmm clustering for unsupervised subword modeling: A contribution to zerospeech 2017," in *ASRU*, 2017.

[27] J. Chorowski, R. J. Weiss, S. Bengio, and A. Oord, "Unsupervised speech representation learning using wavenet autoencoders," *arXiv:1901.08810*, 2019.

[28] R. Fer, P. Matějka, F. Grézl, O. Plchot, K. Veselỳ, and J. H. Černockỳ, "Multilingually trained bottleneck features in spoken language recognition," *Computer Speech & Language*, vol. 46, pp. 252–267, 2017.

[29] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-light: A benchmark for asr with limited or no supervision," in *INTERSPEECH*, 2020.

[30] K. Kawakami, L. Wang, C. Dyer, P. Blunsom, and A. van den Oord, "Learning robust and multilingual speech representations," 2020.

S1. SUPPLEMENTARY METHODS

We describe here ablation experiments comparing our reimplementation of the original CPC model [4] and improvements we made to this model.

S1.1. Changing the normalization method

In order to make the training more stable, we replaced the batch normalization in the original model with layer normalization. The results are illustrated in Table S1.

	Across	Within
Trained on Librispeech-100		
CPC [4]	13.0	9.6
CPC + Layer norm (LN)	**12.0**	**8.7**

Table S1. **Impact of the normalization method on the phoneme discriminability**. Within- and across-speakers ABX scores for the English Zerospeech2017 test set.

S1.2. Choosing the right predictor design

We compared several alternatives to the linear prediction model initially presented in [4]. We supposed that if the prediction network is too simple, then the auto-regressive network will perform a significant part of the prediction task. Thus we though that more complex architecture would improve the quality of our output features. The results of our experiments are compiled in Table S2.

	Across	Within
Trained on Librispeech-100		
CPC + LN	12.0	8.7
CPC + LN + Conv8	13.4	9.2
CPC + LN + FFD	11.7	8.56
CPC + LN + transformer	9.5	7.3
CPC + LN + transformer + dropout	**9.3**	**6.8**

Table S2. **Phoneme discriminability for various predictors design**. Within- and across-speakers ABX scores for the English Zerospeech2017 test set.

S2. SUPPLEMENTARY RESULTS

Here, we present results on the CPC features trained on the recently released Libri-light 60K dataset[29]. As seen in Table S3, we now beat both the Bottleneck and Supervised features on all languages except one. The comparison between Bottleneck and CPC features is displayed in Figure S1.

Model	Pretraining	Frozen	du	es	fr	it	ky	ru	sv	tr	tt	zh	Avg
Bottleneck [28]	Babel-1070h	Yes	47.9	36.6	48.3	39.0	38.7	45.2	52.6	**43.4**	42.5	54.3	44.9
Supervised	LS-100h	Yes	**42.4**	**36.4**	47.0	40.5	41.0	43.6	47.0	48.5	41.5	56.8	44.5
Modified CPC	LL-60K	Yes	43.1	**36.4**	**44.3**	**37.8**	**37.5**	**42.4**	**46.5**	45.7	**40.6**	**53.2**	**42.7**

Table S3. **Transfer of pre-trained phoneme features across languages.** Phone Error Rate on linear classification of phonemes based on pre-trained features on 60kh of Libri-light, compared to multilingual bootleneck features ("Bottleneck") trained on 1070h from the Babel dataset and a supervised baseline trained on LibriSpeech 100h clean. The linear classifier is trained on the frozen features using 1h of speech from the Common Voice database in different languages. We report Phone Error Rate. The languages are: Dutch (du), Spanish (es), French (fr), Italian (it), Kyrgyz (ky), Russian (ru), Sweedish (sv), Turkish (tr), Tatar (tt) and Mandarin (zh).

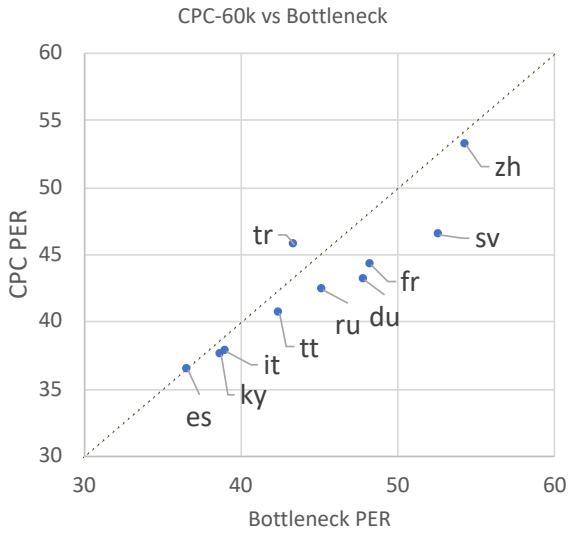

Fig. S1. **CPC versus Bottleneck features.** The CPC features here have been trained on the 60Kh libri-light dataset.