TOWARDS UNSUPERVISED LEARNING OF SPEECH FEATURES IN THE WILD

Morgane Rivière[1], *Emmanuel Dupoux*[1,2]

[1]Facebook AI Research [2]EHESS, CNRS, INRIA, ENS-PSL University

ABSTRACT

Recent work on unsupervised contrastive learning of speech representation has shown promising results, but so far has mostly been applied to clean, curated speech datasets. Can it also be used with unprepared audio data "in the wild"? Here, we explore three potential problems in this setting: (i) presence of non-speech data, (ii) noisy or low quality speech data, and (iii) imbalance in speaker distribution. We show that on the Libri-light train set, which is itself a relatively clean speech-only dataset, these problems combined can already have a performance cost of up to 30% relative for the ABX score. We show that the first two problems can be alleviated by data filtering, with voice activity detection selecting speech segments, while perplexity of a model trained with clean data helping to discard entire files. We show that the third problem can be alleviated by learning a speaker embedding in the predictive branch of the model. We show that these techniques build more robust speech features that can be transferred to an ASR task in the low resource setting.

Index Terms: speech recognition, unsupervised representation learning, contrastive predictive coding, data filtering, speaker adaptation

1. INTRODUCTION

Unsupervised representation learning has been studied as a topic of its own [1, 2, 3], but recently gained attention as a pretraining method to obtain speech features that can be fine tuned for downstream application with little labelled data [4, 5, 6, 7, 8]. This opens up the prospect of constructing speech technology for low resource languages. However, at present, unsupervised representation learning studies have mostly focused on relatively clean, well curated read speech (e.g., the Libri-light corpus [5], and most of the zero resource speech corpora [2, 3, 9]). Do these technique still work with raw, unfiltered audio? For practical applications, obtaining large quantities of clean data is challenging, especially if this is to be scaled to many languages. Scientifically, this question relates to whether future frame prediction objective functions still work when the sources of audio include not only speech but other kinds of sounds (noise, music), when the speech is of relatively low quality, or when speakers are heavily imbalanced. In [7], it is claimed that Contrastive Predictive Coding (CPC), a recent unsupervised representation learning algorithm does work robustly with diverse and noisy speech data (a mix of audio from YouTube videos plus several speech corpora). However, the authors did not compare their pretraining with that on equivalent amount of clean data, and it is impossible to quantify the degradation imposed by noisy input. In addition, they used labels (>80 hours in English) to fine tune the representation, which may partially compensate for low quality speech features.

Here, we systematically study the effect of non-speech, low quality speech and speaker imbalance, which we evaluate with an unsupervised metric before any fine tuning (Exp. 1). We work with Libri-light [5], a large (60kh), unlabelled, relatively clean open-source speech dataset of volunteer-recorded English books. We show that non-speech can be filtered out with a pretrained Voice Activity Detector (VAD). As for low quality speech, we show that perplexity-based data selection can partly alleviate the problem (Exp. 2). In Exp. 3 we show that speaker normalization techniques can help with speaker imbalance. In Exp. 4, the best models are tested on sections of Libr-light of varying sizes (from 100h to 60kh). Finally, in Exp. 5, we study the impact of the most successful of these manipulations (perplexity-based filtering and speaker normalization) as a pretraining method for a downstream ASR task in a low resource setting. We fine tune our unsupervised features with only 1h or 10h of label and find that our robust CPC pretrained features improve both Phone Error Rate (PER) and Word Error Rate (WER), by a factor of 14% relative. The code used in this paper is publicly available[1].

2. RELATED WORK

CPC. Van den Oord et al. [1] introduced Contrastive Predictive Coding, a method for unsupervised representation learning. Applied to speech, CPC trains a convolutional encoder and a predictor for future embeddings of the encoder. The contrastive loss prevents mode collapsing: an embedding should be close to positive future embeddings and distant from negative ones. CPC was used as pretraining for ASR [4] and speaker identification [10, 11]. Here we reuse the CPC implementation of [5, 6].

Data Filtering. Data filtering has been used in the context of semi-supervised learning. Most papers used confidence-based filtering [12, 13, 14, 15]. We use the perplexity of several types of pretrained decoder to filter out bad quality files.

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

Speaker adaptation. Several methods have been used to make ASR systems more robust to speaker variations. Some models introduce speaker embeddings as auxiliary inputs [16, 17], other apply speaker-aware layer-wise affine transformation [18], or a speaker memory [19] . Another technique consists in applying adversarial losses on a speaker classification auxiliary tasks to render the representation speaker invariant [20]. Finally, label imbalance can be addressed by reweighting the loss [21] or by differently resampling the data [22]. While such techniques have been applied to unsupervised representation learning [23, 24, 25] to our knowledge, they have not been tested with predictive learning.

Evaluation of unsupervised features Unsupervised features can be evaluated with two kinds of methods, depending of the end goal of these features. In the zero resource setting [26, 27], the aim is to build speech representations without any labels. Distance-based methods like ABX [28, 29] or Mean Average Precision [30] evaluate the intrinsic quality of the features without having to retrain the system on any label. They compare the distance of segments of speech that belong to the same phonemes to those of segments of speech belonging to different phonemes. In the low resource setting, features are viewed as pretraining and are evaluated as their ability to transfer to some downstream task like phone or word recognition [4, 7, 31, 32, 33]. Typically, studies belong either to one class or the other, making it difficult to know whether the two kinds of metric correlate. Here, we use both zero resource (ABX) and low resource evaluations (PER and WER), building on Libri-light [5] which provides the three metrics on the same dev and test set, together with a large unlabeled train set and a limited train set of labeled speech.

3. METHODS

3.1. CPC architecture

A CPC architecture is composed of three components. A convolutional *encoder network* produces an embedding z_t of the raw audio signal. The sequence (z_t) is then passed to a *recurrent context network* to build the context representation c_t. At each time step t, a *predictor neural network* $Pred$ produces from c_t several outputs $Pred^k$ each one reconstructing future embeddings z_{t+k} $(0 < k \leq K, K = 12)$. The loss $\mathcal{L}$ is contrastive and tries to maximize the dot product between the predicted and correct future representation while minimizing the dot product with a sample of 128 negative examples $\mathcal{N}_{t,k}$.

$$\mathcal{L} = \frac{1}{K}\sum_{k=1}^{K} \log \frac{\exp(Pred^k(c_t)^T z_{t+k})}{\sum_{n\in\mathcal{N}_{t,k}} \exp(Pred^k(c_t)^T z_n)}$$

In this paper, we use a re-implementation of the CPC model [34], which we call CPC2. The encoder architecture is the same (5 convolutional layers with kernel sizes [10,8,4,4,4], strides [5,4,2,2,2] and hidden dimension 256), for the context network, we used 2-layer LSTM, and for the prediction network, we used a multi-head transformer [35], each of the 12 heads predicting one future time slice.

3.2. Datasets and evaluation metrics

In the experiments reported below, we trained our CPC2 model on subsets from two datasets: the first one is Librispeech [36], which contains well-segmented short sentences in good quality read speech, from a balanced set of speakers; the second one is the unlabeled train set of Libri-light [5], which is a less curated but larger dataset (60kh) of read speech. Both datasets originate from the same source: Librivox, a public dataset of audiobooks,[2] but differ in the amount of preprocessing and data filtering.

We investigate the quality of the learned representations using the Libri-light ABX metric for unsupervised representation learning. This is a distance-based metric which estimates the probability that two speech segments with the same transcriptions are closer to one another than to a speech segment with different transcriptions.

Formally, given the embedding features for the n_A tokens of a speech category A and the n_B tokens of a category B, the *asymmetric ABX discriminability error* between A and B is the proportion of representation triplets a, b, x, with a and x taken from A and b taken from B, such that x is closer to b than to a, i.e.:

$$\hat{e}(A,B) := \frac{1}{S_{AB}} \sum_{\substack{a,x\in A\\ x\neq a}} \sum_{b\in B} \left[1_{d(b,x)<d(a,x)} + \frac{1}{2} 1_{d(b,x)=d(a,x)}\right]$$

Where $S_{AB} = n_A(n_A-1)n_B$ and d is a distance function between a pair of model representations. The *ABX discriminability error* between A and B is then obtained with:

$$\hat{\epsilon}(A,B) = \hat{\epsilon}(B,A) := \frac{1}{2}[\hat{e}(A,B) + \hat{e}(B,A)].$$

In our case, d is the Dynamic Time Warping-realigned average angle (arc-cosine of the normalized dot product) between each frames. The test uses as categories minimal pairs of triphones that only change in the central phoneme ('bet' vs 'bit'). As described, the test is conducted within-speaker (a, b and x are from the same speaker). This score can also be computed across speakers: in this case we compare identical triphones but from different speakers. This metric has been shown to be useful to analyse the linguistic content of speech features without having to train a classifier [29], and has been used in the Zero Resource Challenge series [2, 3, 9]. We also found that this metric correlates very well with the linear separability of the features for phone classification.

Once the best method for pretraining according to the ABX metric is found, we scale the models and test it on low resource phone recognition and word recognition tasks evaluated with PER and WER.

[2] https://librivox.org/

4. EXPERIMENTS

4.1. Exp. 1. How much does noisy data hurt?

In this experiment, we demonstrate that the data selection process impacts the quality of the learned representation. We construct 80h samples from LibriSpeech and Libri-light. For LibriSpeech, we construct two samples, one from the 100h-clean subset and one from the 500h-other (`LS80-clean`, and `LS80-other`, resp.). The "clean" and "other" sections of LibriSpeech were originally selected by using WER from a baseline system as a filter [36]. This gives us a natural split in terms of speech quality.

Algorithm 1: Greedy speaker selection algorithm to build a speaker-balanced dataset

```
Data: Target time $T$; Speakers sizes $(l_0, l_1, ..l_n)$ with
      $l_0 \leq l_1 \leq l_n$;
Result: Out speaker sizes $(o_0, o_1, ..o_n)$
$i_l, t, (o_i)_i := 0$;
while $t < T$ do
    $tar = \frac{T-t}{n-i_l}$;
    if $s_{i_l} \geq tar$ then
        $i_c := i_l$;
    else
        $i_c := \min\{i \in [i_l; n] | l_i \geq tar\}$;
    end
    for $i \in [i_c; n]$ do
        $d = \min(l_i, tar)$;
        $t$ += $d$; $o_i$ += $d$; $l_i$ −= $d$;
    end
    $i_l := i_c$
end
```

Contrary to Librispeech, Librilight's data is unfiltered in terms of speech quality and even contains segments without any voice activity (about 8% of the raw recording). We extract 4 subsets of 80 hours from Libri-light-600, the 'small' cut of the dataset containing approximately 600h of speech. In the first two subsets, the non-speech parts were filtered out using Voice Activity Detection (VAD) computed with pyannote.audio [37]. The `LL80-p` subset samples the files uniformly, and ends up with a power law distribution of speakers. The `LL80-e` subset attempts to mimic the speaker distribution of LibriSpeech, equal amounts of speech per speaker, using a greedy sampling method (Algorithm 1). We also made two other version, where no VAD has been applied (raw versions: `LL80-pR` and `LL80-eR`).

In Table 1, we see that the two samples of Librispeech (`LS80-clean` and `LS80-other`) are quite different in terms of their ability to provide good unsupervised representations. Averaged across the 4 ABX scores, training on the "clean" sample yields a 12% relative advantage compared to the "other" sample. This effect is the same whether the representation is evaluated on the "clean" or on the "other" section of the Librispeech dev set. Interestingly, there is no advantage of being trained and tested on the same domain: the "other" sample is simply worse for pretraining. We can also see that the `LL80-e sample` gives results intermediate between the clean and the other training set, which makes sense because Libri-light has actually been less curated than LibriSpeech, and may correspond to the base distribution from which the clean and other sets were originally sampled. Next, we can see that VAD is helping CPC learning: the LL80 subsets without VAD suffer a decrement of 4.5%-4.8% relative performance. Finally, we can see that speaker imbalance has a large effect, with a drop of 13.8%-14.1% relative.

Training Setup			ABX within		ABX across		
			dev	dev	dev	dev	
Features	VAD	spk.	clean	other	clean	other	*avg*
MFCC			10.95	13.55	20.94	29.41	*18.71*
LS80-clean	˜y	equi	6.02	8.11	7.55	12.9	*8.65*
LS80-other	˜y	equi	7.15	9.23	9.11	13.8	*9.82*
LL80-e	y	equi	6.20	8.31	7.93	12.8	*8.81*
LL80-p	y	power	7.42	9.40	9.08	14.3	*10.05*
LL80-eR	n	equi	6.82	8.20	8.52	13.4	*9.23*
LL80-pR	n	power	7.62	9.76	9.67	15.0	*10.5*

Table 1: **Exp 1. Effect of noisy data on CPC features.** Within- and across-speaker phoneme ABX discriminability scores (lower is better) on the Libri-light clean and other dev sets on CPC features. LS datasets are sampled from LibriSpeech datasets, and LL from Libri-light-600. -p indicates follow a power law speaker distribution, as opposed to a balanced one -e. -R means that we kept the original raw files including non-speech (˜ in LS, non-speech parts are minimal).

4.2. Exp 2. Data selection through perplexity

Here, we test whether the loss of performance due to variations in speech quality can be addressed with appropriate data selection. Panayotov et al. [36] used the word error rate (WER) to split LibriSpeech in "clean" and "other" subsets. This method requires a valid language model and clean text labels roughly aligned with the audio data. Those elements may not be available for low resources languages. This is why we turned to somewhat simpler downstream tasks for data filtering. We tested three of them: phone classification, unaligned phone transcription and CPC classification.

As for *phone classification*, we consider for each sequence S of a given dataset the average perplexity of a classifier trained on aligned phones:

$$\mathcal{P}(S) = \frac{1}{|S|} \sum_{s_i \in S} 2^{-\sum_{p \in P} f(s_i,p) \log(f(s_i,p))} \quad (1)$$

Where $f(s_i, p)$ is the posterior probability of the segment s_i to represent the phoneme p. To build the phone decoder, we simply plug a phone classifier on top of a CPC model trained on LibriSpeech clean-100. As for *unaligned phoneme transcription*, we reasoned that aligned phonemes are not necessarily available in given languages, and computed the perplexity of a decoder trained with unaligned data using the CTC loss [38]. Finally, we considered the fact that for some languages, even a small amount of labelled unaligned phone data can be hard to find. CPC being a classification task (an embedding being classified as being either a positive or a negative future embedding), we used the average perplexity over the K=12 time predictions of the CPC model itself trained on clean data, as a way to single out noisy data.

For each of the three tasks, we found out that data sampled from LibriSpeech-other have on average an higher perplexity score than those sampled from the clean subsets. We then used the perplexity of the models trained on LibriSpeech clean-100 to filter the LibriSpeech other-500 data down to 80-hours; this was done on a file-by-file basis. Similarly, we filtered Libri-light-600 down to the 200h top or bottom perplexity before reducing to 80 hours using Algorithm 1 to obtain a balanced speaker distribution. Table 2 shows the results of these different filtering techniques. In the case of Libri-light, the files correspond not to entire chapters, but to automatically segmented files based on the VAD (less than 1min; segmentation scripts provided in the distribution). The two supervised techniques were the most successful, especially CTC which practically cancelled the detrimental effect of low quality speech with a relative gain of 11% for LibriSpeech (and only 2.5% in Libri-light) compared to no filtering. After filtering, both LibriSpeech-other and Libri-light are within 0.1% absolute of the performance of the clean dataset. The unsupervised method using CPC however was less consistent, with a relative reduction of only 6% for the LibriSpeech dataset, and a decrement of (4.5%) for the Libri-light dataset.

4.3. Exp 3. Mitigating unbalanced speaker distributions

As seen in Table 1, speaker imbalance affects negatively the performance of CPC training (around 14% relative). Our method (Algorithm 1) to get speaker balanced datasets amounts to throwing away more than 80% of the data in an unbalanced dataset like Libri-light. Here (Table 3) we investigate three additional methods to address the effect of speaker imbalance at train time while keeping all the data.

Re-sampling speakers. Here, the idea is that when building the batches for each step of CPC, sequences from a speaker poorly represented are over-sampled, while sequences from a speaker appearing a lot in the dataset are under-sampled. This way, we simulate a balanced distribution without discarding any data. We found that a square root compression of speaker probability worked better than a log compression or a flattening to a uniform distribution (with a very modest, less than 1% relative improvement).

Speaker embedding in prediction. Another possible option to mitigate the effect of speaker imbalance in the dataset is to provide the predictor network with speaker data in order to incite the context embedding to be more speaker invariant. To do so, we train a speaker embedding and insert it to the prediction input. Cui et al. [18] proposed to transform the speaker embedding into an affine transform to apply to the input vector. We found out that, in our case, it was better to just concatenate the embedding with the input features, which

	Within spk.		Across spk.		
	dev	dev	dev	dev	
Train set	clean	other	clean	other	avg
LS80-clean	6.02	8.11	**7.55**	12.9	8.65
LS80-other	7.15	9.23	9.11	13.8	9.82
LS80-other-hiPhone	8.66	10.3	11.0	15.2	11.29
LS80-other-loPhone	6.62	8.74	8.13	13.1	9.14
LS80-other-hiCTC	8.56	10.4	10.5	15.0	11.21
LS80-other-loCTC	6.17	**8.07**	7.94	**12.8**	8.75
LS80-other-hiCPC	8.33	10.4	10.3	15.0	11.0
LS80-other-loCPC	6.49	8.46	8.49	13.4	9.21
LL80-e	6.20	8.31	7.93	12.8	8.81
LL80-e-hiPhone	6.22	8.21	7.97	12.9	8.82
LL80-e-loPhone	**5.99**	8.17	7.70	**12.8**	8.66
LL80-e-hiCTC	6.51	8.44	8.27	13.3	9.13
LL80-e-loCTC	**5.95**	**8.05**	7.64	**12.7**	**8.59**
LL80-e-hiCPC	6.86	8.72	8.50	13.7	9.45
LL80-e-loCPC	6.67	8.68	8.37	13.2	9.23

Table 2: **Exp 2. Effect of perplexity-based data filtering on CPC features.** Within- and across-speaker phoneme discriminability scores (lower is better). LS80-other datasets are sampled from LibriSpeech-other500; LL80-e from Librilight600, with similar speaker representation and VAD filtering. Data selection samples high (`hi`) or low (`lo`) perplexity of a system trained on phoneme classification (`Phone`), phone decoding (`CTC`) or `CPC` on clean data.

	Within spk.		Across spk.		
	dev	dev	dev	dev	
System	clean	other	clean	other	avg
LS80-clean	6.02	8.11	7.55	12.9	8.65
LS80-other	7.15	9.23	9.11	13.8	9.82
LL80e	6.20	8.31	7.93	12.8	8.81
LL80p	7.42	9.40	9.07	14.3	10.05
LL80p+SResamp	6.99	9.21	9.14	14.5	9.96
LL80p+SEmb	6.90	8.89	8.70	13.8	9.57
LL80p+SAdv	9.84	11.5	13.3	18.1	13.18
LL80p+All	10.8	12.8	15.5	20.9	15.00

Table 3: **Exp 3. Effect of compensation for speaker imbalance at train time on CPC features.** Within- and across-speaker phoneme discriminability scores (lower is better) on the Libri-light clean and other dev sets. We tested moderating speaker imbalance through square root resampling (`+SResamp`), a speaker embedding in the prediction network (`+SEmb`), a speaker adversarial loss (`+SpeakAdv`), and a combination of all (`+All`).

gained a 4.7% relative improvement.

Speaker adversarial training. Following the work of Hadad et al. [39] in computer vision, we considered the use of an adversarial classification loss on speaker label, to disentangle the feature representation. More precisely, a classifier is trained to discriminate speakers using CPC features while the model learn to fool the classifier. An adversarial loss $\mathcal{L}_A$ is added to the CPC optimization criterion:

$$\mathcal{L}_A = \sum_{z_i} \sum_{s_j \in \mathcal{S}} p(s_j|z_i) \log p(s_j|z_i) \qquad (2)$$

Where $\mathcal{S}$ is the set of speakers, and $p(s_j|z_i)$ is the score of the classifier for the speaker s_j given the feature z_i. Minimizing $\mathcal{L}_A$ is equivalent to maximizing the entropy of the classifier. Indeed, when working with a classifier containing 3 classes or more, we need to reward the encoder for building features invariant across speaker and thus with a high entropy. It is not enough to force the classifier to point out a wrong class.

Unfortunately, this idea did not work; not only Speaker Adversarial training did not help mitigating speaker imbalance, but it degraded the performance. A possible explanation for this is that the encoder embedding is too low level for speaker invariance to be achievable. Therefore we are asking the encoder to perform an impossible task.

Dataset name	Duration	N_S	H_s/H_{max}	Avg. PPX
LS-80	80h	251	1.00	3.29
LL600	526h	489	0.80	3.23
LL600-e-loCTC	597h	1,875	1.00	3.13
LL6k	4,746h	1,596	0.78	3.21
LL6k-e-loCTC	5,703h	5,597	0.92	3.12

Table 4: **Statistics of unfiltered and filtered Libri-light**. N_S is the number of speakers in the dataset. $H_s/H_{max} = H_s/\log N_S$ is the normalized entropy of the speaker distribution, the closer this ratio is to one, the more balanced the dataset. Speaker balancing was made with Algorithm 1. We consider the CTC phone perplexity (-loCTC).

4.4. Exp. 4. Scaling up to larger datasets

Here, we use the best of the data selection and speaker compensation methods we have found on the 60,000 hours of the Libri-light dataset. We took the top 30,000 hours of data with low phone perplexity and built two speaker-balanced datasets from it: LL600-e-loCTC containing 600 hours of data and LL6k-e-loCTC with 6000 hours.

We then compare their ABX scores with models trained on the unfiltered Libri-light 600 and 6kh datasets (LL600 and LL6k). Statistics on theses datasets are available in Table 4 and the results are shown in Table 5. We found that these two operations substantially improved the performance over the same amount of data without filtering and compensation (relative improvement: 16% and 12% for 600 and 6k hours, respectively). The result (together with the improved architecture CPC2) yields scores that beat the previous state-of-the art by a substantial margin, with 100 to 10 times less training data (although the large dataset was presumably necessary for the filtering to select adequate amounts of good data).

4.5. Exp 5. Application to ASR

To estimate the value of our data selection method, we test it for low resource ASR tasks. The idea is to be able to pretrain CPC features on potentially noisy data, and fine tune the system for ASR on very limited amounts of labels. We will use English as a 'low resource' language by limiting the amount of labeled data to 1h and 10h. Therefore, we will not obtain a state-of-the art system, since such systems are trained on 1000 hours of data or more. Our aim is rather to test whether our methods for robust CPC training yield features that translate into improved low-resource ASR performance.

We consider two datasets for the pretraining: the unfiltered Librilight-6kh dataset (LL6k) and our filtered and speaker balanced 6kh subset (LL6k-e-loCTC). Besides, in order to take advantage of the large amount pretraining data we propose a scaled up version of our model: CPC-big with 512 hidden dimension (instead of 256) and a 4 layers-LSTM (instead of 2). As shown in table 4 LL6k-e-loCTC has a quite balanced speaker distribution. For this reason, we tested CPC-big with and without speaker embedding (SEmb) to see how they would impact the performance of a larger model in this configuration.

We add a linear layer on top of the contextual network and fine-tune the entire models on the 1h and 10h phonetically labeled train subsets of Libri-light using the CTC loss. Besides, in order to take full advantage of the labelled data, we perform some pitch augmentation using the WavAugment library as described in [40]. The PERs of this decoder on LibriSpeech dev and test are shown in Table 6. The combination of the data selection and the speaker embeddings provide an improvement of the PER 9 to 14% for both architectures.

Finally, we need to estimate how our data selection method impacts the performance of speech to text. To do so we consider a rather simple setting: we simply plug our phonetic representations to the KenLM decoder provided by the wav2letter library [41]. As far the language model is concerned, we simply use the 4-gram model provided with librispeech. The WER resulting on LIbriSpeech dev and test is shown in Table 7. As shown in the table, the data filtering also has an impact on the final word inference task, even with a larger architecture : we see a 13% relative improvement for the small architecture and a 25% relative improvement for the big one. However, it is worth noticing that the speaker embeddings tend to decrease the overhall performances when applied to a larger architecture.

Our performances are correct but not as good as what has previously been obtained in similar low resource settings like wav2vec 2.0 [42]. However, our architecture is much smaller than the ones proposed in wav2vec 2.0 : CPC-big has 8 times less parameters than their small model and 30 times less

	ABX within speaker				ABX across speaker			
System	dev-clean	dev-oth.	test-clean	test-oth.	dev-clean	dev-oth.	test-clean	test-oth.
MFCC Baseline	10.95	13.55	10.58	13.60	20.94	29.41	20.45	28.5
CPC LL60k [5]	6.11	8.17	5.83	8.14	8.05	12.83	7.56	13.42
CPC2 LL600	5.79	7.89	5.50	8.00	7.31	12.3	7.17	12.7
CPC2+SEmb LL600-e-loCTC	4.67	6.66	4.49	6.81	5.89	10.6	5.78	11.0
CPC2 LL6k	5.20	7.37	5.01	7.34	6.49	12.0	6.62	12.0
CPC2+SEmb LL6k-e-loCTC	**4.64**	**6.52**	**4.31**	**6.65**	**5.78**	**10.2**	**5.62**	**10.9**

Table 5: **Exp 4. Scaling up data filtering and speaker imbalance mitigation to large datasets.** Within- and across-speaker phoneme discriminability errors (lower is better) on the LibriSpeech dev and test sets for CPC features obtained with the `CPC2` model, or the same model with speaker embeddding (`+SEmb`) trained on raw (`LL600`, `LL6k`) and speaker balanced, filtered with CTC perplexity (`-e-loCTC`) Libri-light datasets. For comparison, the sota on the large Libri-light dataset `LL60k`.

than their big one. The improvements provided by our method remain when increasing the size of the model. Therefore, we hope that data selection can also be successfully used with bigger architectures.

System	dev clean	dev other	test clean	test other
Fine-tuned on 1h of data				
CPC2 LL6k	28.9	42.3	29.2	43.1
CPC2+SEmb LL6k-e-loCTC	26.9	38.0	26.6	40.2
CPC-BIG LL6k	25.0	36.9	24.7	38.3
CPC-BIG+LL6k-e-loCTC	**21.3**	**32.6**	**21.5**	**34.4**
CPC-BIG+SEmb LL6k-e-loCTC	21.4	34.7	**21.5**	35.1
Fine-tuned on 10h of data				
CPC2 LL6k	27.8	39.3	27.1	41.9
CPC2+SEmb LL6k-e-loCTC	24.6	36.1	24.3	38.3
CPC-BIG LL6k	18.8	31.2	18.7	33.4
CPC-BIG+LL6k-e-loCTC	**16.2**	**27.6**	16.3	**29.3**
CPC-BIG+SEmb LL6k-e-loCTC	**16.2**	28.1	**16.0**	30.5

Table 6: **Exp 5. PER on LibriSpeech dev and test sets**. Both architectures are pre-trained with or without Speaker Embedding (`+SEmb`) on Libri-light 6k (`LL6k`) or a sample of Libri-light of the same duration, but balanced by speaker and filtered using the perplexity of a phone classifier (`LL6k-e-loCTC`).

5. CONCLUSION

We have found that all of the three tested factors of noise (presence of non-speech parts, low quality speech, speaker imbalance) degrade CPC pretraining. The cumulative effect of these three factors adds up to a drop of 30% relative ABX score, even though the Libri-light dataset is itself relatively clean (home made audio books). For instance, most of the recording time is devoted to speech, and the non-speech parts are minimal (8.8%); this may be why VAD filtering is having such a small effect. We suspect that nonspeech may have a larger detrimental effect with in-the-wild recordings when it becomes a dominant category and may capture the CPC loss to the detriment of phonetic learning.

We showed that these problems can be partially addressed. Non-speech segments can be excised out using a VAD, low quality recordings can be detected and discarded using perplexity, and speaker imbalance partially compensated by speaker adaptation techniques. We also show that these techniques can be scaled up to large datasets, yielding a new state-of-the-art in the libri-light unsupervised representation learning metric. However, only a half to 60% of the performance gap can be filled by these techniques. More problematic, some of these techniques do require supervised training. The VAD was constructed in a supervised fashion and so is the phone recognizer for perplexity filtering. Even speaker adaptation requires that speaker labels are available for each recording. It is to be hoped, however, that some of this supervision can be made non-language specific and therefore apply to low-resource languages.

System	dev clean	dev other	test clean	test other
Fine-tuned on 1h of data				
CPC2 LL6k	39.4	62.7	37.8	65.6
CPC2+SEmb LL6k-e-loCTC	37.0	60.0	35.8	62.2
CPC-BIG LL6k	25.8	40.2	29.1	55.3
CPC-BIG+LL6k-e-loCTC	**23.3**	**43.1**	**22.9**	**45.8**
CPC-BIG+SEmb LL6k-e-loCTC	23.8	43.2	23.1	46.2
Fine-tuned on 10h of data				
CPC2 LL6k	33.2	56.4	31.6	60.7
CPC2+SEmb LL6k-e-loCTC	31.1	53.8	29.2	57.3
CPC-BIG LL6k	19.5	40.1	20.3	45.3
CPC-BIG+ LL6k-e-loCTC	**16.9**	**34.8**	16.8	**37.7**
CPC-BIG+SEmb LL6k-e-loCTC	17.2	35.0	**16.6**	38.3

Table 7: **Exp 5. WER on LibriSpeech dev and test sets**. A CPC2 or CPC-BIG (with a 4 Layer LSTM) with or without Speaker Embedding (`+SEmb`) is pretrained on Libri-light 6k (`LL6k`) or a sample of Libri-light of the same duration, but balanced by speaker and filtered on the CTC perplexity of a pretrained phone classifier (`LL6k-e-loCTC`).

In the meantime, even our partial response to these problem effectively improves the quality of the unsupervised features, which also translates to downstream phone recognition or ASR tasks in the low resource setting.

6. ACKNOWLEDGMENTS

ED's work in his EHESS role was supported by the Agence Nationale de la Recherche (ANR-17-EURE-0017 Frontcog, ANR-10-IDEX-0001-02 PSL*, ANR-19-P3IA-0001 PRAIRIE 3IA Institute) and CIFAR (Learning in Minds and Brains).

7. REFERENCES

[1] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[2] M. Versteegh, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015: Proposed approaches and results." in *SLTU*, 2016, pp. 67–72.

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

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

[5] 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 *ICASSP*, 2020.

[6] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," *arXiv preprint arXiv:2002.02848*, 2020.

[7] K. Kawakami, L. Wang, C. Dyer, P. Blunsom, and A. v. d. Oord, "Learning robust and multilingual speech representations," *arXiv preprint arXiv:2001.11128*, 2020.

[8] W. Wang, Q. Tang, and K. Livescu, "Unsupervised pre-training of bidirectional speech encoders via masked reconstruction," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6889–6893.

[9] E. Dunbar, R. Algayres, J. Karadayi, M. Bernard, J. Benjumea, X.-N. Cao, L. Miskic, C. Dugrain, L. Ondel, A. W. Black *et al.*, "The zero resource speech challenge 2019: Tts without t," *arXiv preprint arXiv:1904.11469*, 2019.

[10] S. Löwe, P. O'Connor, and B. Veeling, "Putting an end to end-to-end: Gradient-isolated learning of representations," in *NIPS*, 2019, pp. 3033–3045.

[11] C.-I. Lai, "Contrastive predictive coding based feature for automatic speaker verification," *arXiv preprint arXiv:1904.01575*, 2019.

[12] R. Zhang and A. I. Rudnicky, "A new data selection approach for semi-supervised acoustic modeling," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1. IEEE, 2006, pp. I–I.

[13] K. Veselỳ, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 267–272.

[14] D. Imseng, B. Potard, P. Motlicek, A. Nanchen, and H. Bourlard, "Exploiting un-transcribed foreign data for speech recognition in well-resourced languages," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 2322–2326.

[15] S. Li, Y. Akita, and T. Kawahara, "Semi-supervised acoustic model training by discriminative data selection from multiple asr systems' hypotheses," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1524–1534, 2016.

[16] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 55–59.

[17] M. Delcroix, S. Watanabe, A. Ogawa, S. Karita, and T. Nakatani, "Auxiliary feature based adaptation of end-to-end asr systems." in *Interspeech*, 2018, pp. 2444–2448.

[18] X. Cui, V. Goel, and G. Saon, "Embedding-based speaker adaptive training of deep neural networks," *CoRR*, vol. abs/1710.06937, 2017. [Online]. Available: http://arxiv.org/abs/1710.06937

[19] L. Sarı, N. Moritz, T. Hori, and J. Le Roux, "Unsupervised speaker adaptation using attention-based speaker memory for end-to-end asr," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7384–7388.

[20] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gang, and B.-H. Juang, "Speaker-invariant training via adversarial learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5969–5973.

[21] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," *arXiv preprint arXiv:1803.09050*, 2018.

[22] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "Addressing imbalance in multilabel classification: Measures and random resampling algorithms," *Neurocomputing*, vol. 163, pp. 3–16, 2015.

[23] S. Feng, T. Lee, and Z. Peng, "Combining adversarial training and disentangled speech representation for robust zero-resource subword modeling," *arXiv preprint arXiv:1906.07234*, 2019.

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

[25] R. Riad, C. Dancette, J. Karadayi, N. Zeghidour, T. Schatz, and E. Dupoux, "Sampling strategies in siamese networks for unsupervised speech representation learning," *arXiv preprint arXiv:1804.11297*, 2018.

[26] J. Glass, "Towards unsupervised speech processing," in *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*. IEEE, 2012, pp. 1–4.

[27] E. Dunbar, J. Karadayi, M. Bernard, X.-N. Cao, R. Algayres, L. Ondel, L. Besacier, S. Sakriani, and E. Dupoux, "The zero resource speech challenge 2020: Discovering discrete subword and word units," in *INTERSPEECH-2020*, 2020.

[28] 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.

[29] T. Schatz, "Abx-discriminability measures and applications," Ph.D. dissertation, Univ. Pierre et Marie Curie, Paris, 2016.

[30] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[31] A. Baevski, M. Auli, and A. Mohamed, "Effectiveness of self-supervised pre-training for speech recognition," *arXiv preprint arXiv:1911.03912*, 2019.

[32] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, "Multi-task self-supervised learning for robust speech recognition," *arXiv preprint arXiv:2001.09239*, 2020.

[33] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," *arXiv preprint arXiv:1910.12607*, 2019.

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

[35] 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.

[36] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," pp. 5206–5210, 2015.

[37] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," in *ICASSP*, 2020.

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

[39] N. Hadad, L. Wolf, and M. Shahar, "Two-step disentanglement for financial data," *CoRR*, vol. abs/1709.00199, 2017. [Online]. Available: http://arxiv.org/abs/1709.00199

[40] E. Kharitonov, M. Rivière, G. Synnaeve, L. Wolf, P.-E. Mazaré, M. Douze, and E. Dupoux, "Data augmenting contrastive learning of speech representations in the time domain," 2020.

[41] Q. X. J. C. J. K. G. S. V. L. R. C. Vineel Pratap, Awni Hannun, "wav2letter++: The fastest open-source speech recognition system," *CoRR*, vol. abs/1812.07625, 2018. [Online]. Available: https://arxiv.org/abs/1812.07625

[42] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020.