LIBRI-LIGHT: A BENCHMARK FOR ASR WITH LIMITED OR NO SUPERVISION

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[†], *E. Dupoux*[†‡]

[†]Facebook AI, [‡]EHESS, ENS, PSL-University, CNRS, INRIA

ABSTRACT

We introduce a new collection of spoken English audio suitable for training speech recognition systems under limited or no supervision. It is derived from open-source audio books from the LibriVox project. It contains over 60K hours of audio, which is, to our knowledge, the largest freely-available corpus of speech. The audio has been segmented using voice activity detection and is tagged with SNR, speaker ID and genre descriptions. Additionally, we provide baseline systems and evaluation metrics working under three settings: (1) the zero resource/unsupervised setting (ABX), (2) the semi-supervised setting (PER, CER) and (3) the distant supervision setting (WER). Settings (2) and (3) use limited textual resources (10 minutes to 10 hours) aligned with the speech. Setting (3) uses large amounts of unaligned text. They are evaluated on the standard LibriSpeech dev and test sets for comparison with the supervised state-of-the-art.

***Index Terms*—** unsupervised and semi-supervised learning, distant supervision, dataset, zero- and low resource ASR.

1. INTRODUCTION

Automatic Speech Recognition (ASR) has made striking progress in the recent years with the deployment of increasingly large deep neural networks trained on increasingly large sets of annotated speech (from thousands to tens of thousands of hours). This approach is hit by diminishing returns as the costs of annotating even larger datasets become prohibitive. It is also difficult to scale beyond a handful of high-resource languages and address the needs of a long tail of low-resource languages, dialectal and idiolectal variants, accents, and registers. As such, there has been a recent surge of interest in weakly supervised solutions that use datasets with fewer human annotations. In the semi-supervised setting, only a fraction of the dataset is labelled and the rest is unlabelled [1, 2], while in a distant supervision setting, the dataset is mostly or entirely unlabelled, but large quantities of unaligned text provide a language model corpus [3, 4]. Other approaches have addressed pretraining with labels from other languages [5, 6] or pretraining using unsupervised objectives [7, 8]. At the extreme of this continuum, zero resource ASR discovers its own units from raw speech [9, 10, 11]. Despite many interesting results, the field lacks a common benchmark (datasets, evaluations, or baselines) for comparing ideas and results across these settings. Here, we introduce Libri-light, a large open-source corpus (60K hours) of unlabelled speech and a common set of metrics to evaluate three settings: (1) the zero-resource/unsupervised setting (ABX), (2) the semi-supervised setting (PER and CER), and (3) the distant supervision setting (WER). The last two settings use a limited-resource training set (10 min, 1h, 10h), and the last one large in-domain and out-of-domain text to train language models. The test sets are identical to LibriSpeech [12] so as to facilitate comparison of weakly supervised results with the state-of-the art in supervised learning. We also provide a baseline system on these three settings. All datasets, metrics and baseline systems are open source[1].

2. RELATED WORK

The release of open source software and datasets has facilitated rapid progress in machine learning and in particular large vocabulary ASR. LibriSpeech is one of the first large-scale open-source datasets and contains over 1000 hours of audio books, together with textual annotations aligned at the sentence level. Mozilla's CommonVoice project has facilitated data collection across several languages and currently contains 2900 hours of read speech in 37 languages[2]. A. Black at CMU has compiled the Wilderness dataset which consists of the text of the Bible read in 750 languages [13]. Other open-source resources are available from OpenSLR[3].

The Zero Resource Challenge has released a series of datasets and metrics for the unsupervised setting [9, 10][4], but the datasets are generally small (between 2.5 and 50 h). In this work, we substantially expand dataset size and use the same evaluation metrics (ABX [14]) for comparability. The IARPA Babel program [15] has also initiated a push towards limited supervision for less studied languages. In

---

[1]https://github.com/facebookresearch/libri-light
[2]https://voice.mozilla.org
[3]http://openslr.org/
[4]https://zerospeech.com

its most difficult track, the dataset contains only 10 hours of transcribed speech in conjunction with with larger amounts of untranscribed audio. Here, we retain 10 hours as a *upper* bound, and add lower-resource sets containing 1 hours and 10 minutes of labeled audio. While distant supervision has been the focus of two JHU-JSALT workshops (2016 [16], 2019 [17]) but no benchmark has yet emerged.

3. DATASET AND METRICS

3.1. Dataset

(a)

Genre's categories (in hours)

Literature : 42760
Science, Craft & Essay : 8978
Undefined : 5510
Religion : 5373
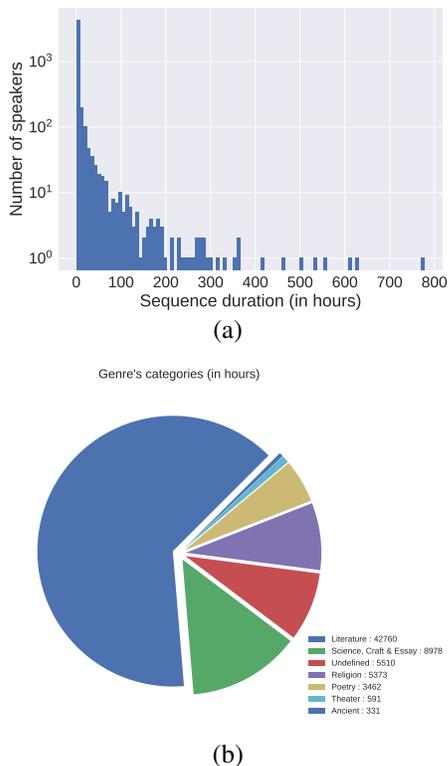Poetry : 3462
Theater : 591
Ancient : 331

(b)

Fig. 1: **Corpus statistics.** (a) Durations in hours per speakers (b) Durations for the 25 most frequent genres.

The dataset is composed four parts: a train set with unlabelled speech, a train set with limited labels, dev/test sets, and a train set containing unaligned text; see Table 1.

Unlabelled Speech Training Set. This dataset was obtained by extracting audio files for English speech from the LibriVox repository[5] containing open source audio books. Files were downloaded and converted to 16kHz FLAC. We then removed corrupted files, files with unknown or multiple speakers, and speakers appearing in LibriSpeech dev and test sets. The potentially duplicated versions of books based on titles were set aside (and distributed as a `duplicate` subset, totalling 4500 hours). We then ran a Voice Activity Detection (VAD) model using the wav2letter++ framework [18] on the recordings to tag onsets and offsets of speech segments.

subset	hours	books	files	per-spk hours	total spkrs
Unlabelled Speech Training Set					
unlab-60k	57706.4	9860	219041	7.84	7439
unlab-6k	5770.7	1106	21327	3.31	1742
unlab-600	577.2	202	2588	1.18	489

subset	hours	per-spk minutes	female sprks	male spkrs	total spkrs
Limited Resource Training Set					
train-10h	10	25	12	12	24
train-1h	1	2.5	12	12	24
train-10m*	10min	2.5	2	2	4
Dev & Test Sets (from LibriSpeech)					
dev-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33
test-clean	5.4	8	20	20	40
test-other	5.1	10	17	16	33

subset				tokens	vocab
Unaligned Text Training Set					
librispeech-LM (in-domain)				800M	200K

Table 1: **Datasets stats in Libri-light**. *Six different versions of the 10 min datasets have been constructed, the union of these small datasets make up the 1h dataset.

The VAD segments were used to derive an average SNR for each file. For each file, we constructed JSON metadata including title, unique speaker ID, SNR, genre, and list of valid VAD_block (block of more than 500ms of speech indicated by onsets and offsets). We created three dataset splits based on different sizes: (`unlab-60k`), (`unlab-6k`) and (`unlab-600`), matched in genre distribution (the smaller cuts are included in the larger ones, see the stats in Table 1). The distributions by speaker and genres are in Figure 1. The total amount of speech in the dataset is 62.2K hours, including the `duplicate` files.

Limited-resource Training Set. For training with limited supervision, we selected three subsets of the LibriSpeech training set: a 10 hour set, a 1 hour set, and six 10-minute sets (the six 10-minute sets together make up the 1h set, and the 1h set is included in the 10h set). In each set, half of the utterances are from the clean and other training sets, respectively. We additionally provide orthographic transcriptions from LibriSpeech and phonetic transcriptions generated from phonemizer[6].

Dev and Test Set. The dev and test sets are the same as that of LibriSpeech (5.4 hours for dev-clean, 5.3 hours for dev-other, 5.4 hours for test-clean, and 5.1 hours for test-other) and are intended for testing and tuning. All dev or test set audio has been removed from training sets. The ground-

System	ABX within speaker				ABX across speaker			
	dev-clean	dev-other	test-clean	test-other	dev-clean	dev-other	test-clean	test-other
MFCC Baseline	10.95	13.55	10.58	13.60	20.94	29.41	20.45	28.5
CPC unlab-600	7.36	9.39	6.90	9.59	9.58	14.67	9.00	15.1
CPC unlab-6k	6.51	8.42	6.22	8.55	8.48	13.39	8.05	13.81
CPC unlab-60k	**6.11**	**8.17**	**5.83**	**8.14**	**8.05**	**12.83**	**7.56**	**13.42**

Table 2: **ABX errors on unsupervised CPC trained features.** Within- and across-speaker phoneme discriminability scores (lower is better) on the LibriSpeech dev and test sets as a function of varying quantities of unlabelled speech.

truth phonetic sequences for the dev and test sets were generated as above; in addition, for ABX evaluation, forced alignment was obtained using a model trained on LibriSpeech.

Unaligned Text Training Set. For training a language model in the distant supervision setting, we consider the LM corpus provided in LibriSpeech[7] which contains 800M tokens and a vocabulary size of 200k from 14.5k public books from Project Gutenberg. This corpus only partially overlaps with the content of our unlabelled training set and can thus be considered in-domain. Several options exist for publicly available out-of-domain corpora (wikitext103, 1Billion word, etc).

3.2. Metrics

We provide 3 sets of metrics for the unsupervised, distantly-supervised, and semi-supervised settings.

For the unsupervised setting, the aim is to extract speech representations (discrete or continuous) which encode the phonetic content of the language while ignoring irrelevant information (channel, speaker, etc). The representation is evaluated using ABX error, a distance-based metric used in previous zero resource challenges [9, 10, 11]. For a given pair of sounds (for instance, "bit" versus "bet"), we compute the probability that the distance between a random token of "bit" (X) is closer to another token of "bit" (A) than to a token of "bet" (B). The ABX error rate is obtained by averaging across all such minimal pairs of phone trigrams in the corpus. For the "within-speaker" score, A, B and X are from the same speakers; in the "across-speaker" score, A and B are from the same speaker, but X is from a different speaker (see [19]).

For the semi-supervised setting, we evaluate the quality of learned acoustic representations with little annotated data. Models can be trained either with character or phonetic targets using limited data and measured by either Character Error Rate (CER) or Phoneme Error Rate (PER).

For distant supervision, we evaluate how the learned representations can be used to decode speech at the word level using a pre-trained language model. We use Word Error Rate (WER) for the evaluation. Because the dev and test sets are from LibriSpeech, this allows to compare distant supervision directly with SOTA supervised models. More details on dataset and metrics in Supplementary Section S1.

System	dev-clean	dev-other	test-clean	test-other
no pretraining+train-10h	45.9	55.7	43.7	58.6
CPC unlab-60k+train-10m	40.1	51.5	39.4	53.3
CPC unlab-60k+train-1h	32.2	44.6	31.6	46.8
CPC unlab-60k+train-10h	**28.4**	**41.4**	**27.9**	**43.6**

Table 3: **PER/CER in the semi-supervised setting.** A pretrained CPC system plus a linear classifier trained on limited amounts of labels compared to the same system trained from scratch (PER).

4. BASELINE SYSTEMS

In the unsupervised setting, we use a PyTorch implementation of the Contrastive Predictive Coding (CPC) system [7] trained to predict the hidden states of N future speech frames and containing an encoder, a sequence model, and a predictor. The encoder maps waveforms to hidden states (one 512 dimensional embedding every 10 ms frames) using a stack of 5 convolutional layers. The sequence model encodes the hidden states into a 512-dimensional phonetic embedding with one layer of Gated Recurrent Units (GRUs). The predictor maps the last phonetic embedding onto a future hidden state using a linear projection (one linear projection per time step, varying from 1 to 12). To avoid collapsing to a trivial solution, the model is trained discriminatively; the loss function aims at decreasing the dot product between predicted and actual future frames while increasing it for frames belonging to negative sequences (distant time windows). We used a reimplementation of the original paper, which we modified to increase stability and performance, as we could not reproduce the original paper results with the provided descriptions. These changes are as follows: we replaced batch-norm with channel-wise normalization, we reduced the hidden and phonetic embeddings to 256 dimensions, used a LSTM instead of a GRU, and used a 1-layer transformer network instead of a linear projection. The original paper obtained 65.5% accuracy on phoneme classification with a linear classifier trained on top of the frozen system's phonetic embedding. Our modified system obtained 68.9% accuracy, while using 4 times fewer parameters in the encoder+sequence model part of the system. We trained it on the three cuts (`unlab-600`, `unlab-6k` and `unlab-60k`).

System	dev-clean	dev-other	test-clean	test-other
Supervised systems (LibriSpeech 1000 h)				
Gated Cnv+4gramLM[20]	4.6	13.8	4.8	14.5
Hybrid+seqdisc+4gramLM[21]	3.4	8.3	3.8	8.8
CPC pretrain + CTC fine-tuning + 4gram-LM				
CPC unlab-600+train-10m	97.3	97.6	97.1	97.7
CPC unlab-600+train-1h	72.2	84.5	70.1	86.3
CPC unlab-600+train-10h	52.5	71.6	49.3	74.1
CPC unlab-6k+train-10m	93.6	95.2	93.2	94.9
CPC unlab-6k+train-1h	67.5	81.3	65.4	82.0
CPC unlab-6k+train-10h	46.4	**66.7**	44.7	69.3
CPC unlab-60k+train-10m	92.5	94.2	92.5	94.4
CPC unlab-60k+train-1h	66.6	80.0	64.7	81.6
CPC unlab-60k+train-10h	**46.1**	66.7	**43.9**	**69.5**
MFSC + TDS + CTC + Grapheme + 4gram-LM				
train-1h	79.4	88.1	78.4	88.0
+ 60k pseudo-label	78.6	86.5	77.2	86.3
train-10h	34.0	60.9	33.5	62.1
+ 60k pseudo-label	30.5	55.8	30.1	57.2
MFSC + TDS + CTC + Phoneme + 4gram-LM				
train-1h	81.1	88.5	80.2	88.7
+ 60k pseudo-label	84.3	90.0	84.0	90.5
train-10h	44.1	64.2	43.8	65.1
+ 60k pseudo-label	**30.0**	**55.8**	**29.3**	**56.6**

Table 4: **WER in the distant supervision setting.** Top: State-of-the-art supervised systems using our 4-gram-LMs. Middle: A CPC system trained with unlabelled speech, fine-tuned with limited data and integrated with a 4-gram word language model (Librispeech-LM). Bottom: A small MFSC TDS system trained on limited labeled data (graphemes or phonemes). The pseudo-labels for the 60k corpus segmented into 36-second chunks are generated and are used to retrain a larger TDS system.

In the semi-supervised setting, we use our baseline pretrained CPC system supplemented with a linear classifier trained with CTC loss on the limited-resource set's phone labels (only the linear layer is fine-tuned). We also provide a from-scratch control with the same architecture trained end-to-end.

For the distant supervision setting, we run two experiments: (1) we use our pretrained CPC system with an improved CTC layer (LSTM) which we fine-tune with orthographic labels in the limited-resource set. We decode with a python wrapped version of the wav2letter++ decoder [18], using a 4-gram KenLM [22] language model trained on the unaligned text set. (2) We use CTC to train small Mel-filterbanks-based TDS systems[23], (7 TDS blocks, 20M parameters, total stride 2) on phonemes/graphemes respec-

tively. We create pseudo-labels by beam-search decoding the 60k-hours unlabelled data with a 4-gram KenLM decoder trained on LibriSpeech-LM. These labels are used to train larger TDS systems (11 TDS blocks, 37M parameters) from scratch which generate WERs when decoding with the same LM. More details on baselines in Supplementary Section S2.

5. RESULTS

The results for the unsupervised setting are shown in Table 2. CPC constructs embeddings with good ABX scores compared to an MFCC baseline, and are in the same range as the SOTA in the Zero Resource Speech Challenge 2017 for English (6.2% within and 8.7% across [24]). The results in the semi-supervised setting (Table 3) show gains in PER in using unsupervised pretraining for several different amounts of fine tuning. The results on the distant supervision (Table 4), while far from supervised state-of-the-art, show that increasing the amount of unsupervised pretraining helps. Pseudo-labels are beneficial but only if the generating and fine-tuned models are initially good (Table 3 and 4).

6. CONCLUSION

We have introduced a new large dataset for benchmarking ASR systems trained with limited or no supervision. We found that unsupervised training with increasingly larger dataset yield better features and can significantly boost the performance of systems trained with limited amounts of labels (from 10 min to 10 hours) both for a phoneme recognition task in a semi-supervised setting and for a word recognition in a distant-supervision setting. The baselines were not particularly optimized for the tasks and are provided only as a proof-of-concept; there is a significant margin with fully-supervised systems. Obvious improvements include using larger models, speaker-adversarial losses, fine tuning the entire system (not just the top layers), and pseudo-labels retraining in all settings. Active learning [25] could further select useful parts of the dataset (we have provided SNR data to facilitate this effort). Yet another approach might apply language modeling techniques directly on unlabelled audio to improve the representations before fine-tuning them [26, 27].

7. REFERENCES

[1] G. Tur, D. Hakkani-Tür, and R.E. Schapire, "Combining active and semi-supervised learning for spoken language understanding," *Speech Communication*, vol. 45, no. 2, pp. 171–186, 2005.

[2] J. Kahn, A. Lee, and A. Hannun, "Self-training for end-to-end speech recognition," *arXiv preprint arXiv:1909.09116*, 2019.

[3] Y.-C. Chen, C.-H. Shen, S.-F. Huang, H.-y. Lee, and L.-s. Lee, "Almost-unsupervised speech recognition with close-to-zero resource based on phonetic structures

learned from very small unpaired speech and text data," *arXiv preprint:1810.12566*, 2018.

[4] Y.-A. Chung, W.-H. Weng, S. Tong, and J. Glass, "Unsupervised cross-modal alignment of speech and text embedding spaces," in *Advances in Neural Information Processing Systems*, 2018, pp. 7354–7364.

[5] 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*. IEEE, 2013, pp. 7304–7308.

[6] K. Veselỳ, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 336–341.

[7] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

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

[9] M. Versteegh, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015: Proposed approaches and results," in *SLTU- Procedia Computer Science*, 2016, vol. 81, pp. 67–72.

[10] E. Dunbar, X.-N. Cao, J. Benjumea, J. Karadayi, M. Bernard, L. Besacier, X. Anguera, and E. Dupoux, "The zero resource speech challenge 2017," in *ASRU*, 2017.

[11] E. Dunbar, R. Algayres, J. Karadayi, M. Bernard, J. Benjumea, X.-N. Cao, L. Miskic, C. Dugrain, L. Ondel, A. Black, L. Besacier, S. Sakriani, and E. Dupoux, "The zero resource speech challenge 2019: TTS without T," in *INTERSPEECH*, 2019.

[12] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[13] A.W. Black, "CMU wilderness multilingual speech dataset," in *ICASSP*. IEEE, 2019, pp. 5971–5975.

[14] 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*, 2013.

[15] M. Harper, "Iarpa babel program," 2014.

[16] C. Liu, J. Yang, M. Sun, S. Kesiraju, A. Rott, L. Ondel, P. Ghahremani, N. Dehak, L. Burget, and S. Khudanpur, "An empirical evaluation of zero resource acoustic unit discovery," in *ICASSP*. IEEE, 2017, pp. 5305–5309.

[17] J. Chorowski, "Distant supervision for representation learning in speech and handwriting," 2014.

[18] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert, "Wav2letter++: A fast open-source speech recognition system," in *ICASSP*. IEEE, 2019, pp. 6460–6464.

[19] T. Schatz, *ABX-discriminability measures and applications*, Ph.D. thesis, 2016.

[20] V. Liptchinsky, G. Synnaeve, and R. Collobert, "Letter based speech recognition with gated convnets," *arXiv preprint arXiv:1712.09444*, 2017.

[21] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "Rwth asr systems for librispeech: Hybrid vs attention-w/o data augmentation," *arXiv preprint arXiv:1905.03072*, 2019.

[22] Kenneth Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the sixth workshop on statistical machine translation*. Association for Computational Linguistics, 2011, pp. 187–197.

[23] A. Hannun, A. Lee, Q. Xu, and R. Collobert, "Sequence-to-sequence speech recognition with time-depth separable convolutions," *CoRR*, vol. abs/1904.02619, 2019.

[24] M. Heck, S. Sakti, and S. Nakamura, "Feature optimized DPGMM clustering for unsupervised subword modeling: A contribution to zerospeech 2017," in *Automatic Speech Recognition and Understanding Workshop*. IEEE, 2017, pp. 740–746.

[25] D. Hakkani-Tür, G. Riccardi, and A. Gorin, "Active learning for automatic speech recognition," in *ICASSP*. IEEE, 2002, vol. 4, pp. IV–3904.

[26] Y.-A. Chung and J. Glass, "Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech," *arXiv preprint arXiv:1803.08976*, 2018.

[27] Anonymous, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *Submitted to International Conference on Learning Representations*, 2020, under review.

[28] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

S1. SUPPLEMENTARY DATASETS AND METRICS

S1.1. Datasets and meta-data

The dataset is constructed according to the following pipeline: data download, exclusion of bad files, conversion to flac, extraction of VAD, SNR, and Perplexity, the construction of JSON files and the split in three cuts of varying sizes.

S1.1.1. VAD

Voice Activity Detection is accomplished using a TDS acoustic model [23] trained using CTC loss [28] on the LibriSpeech dataset using the orthographic transcription. The trained model was used to perform inference (greedy frame-by-frame decoding) with the wav2letter++[18] audio analysis pipeline on the unlabeled audio by mapping all of the letters to SPEECH and the silence symbol to NONSPEECH. The posterior SPEECH probability is added as meta data on the JSON of each file.

The TDS model used for VAD has 100 million parameters and consists of clusters of 2, 3, 4, and 5 TDS blocks separated by 2D convolutions. The duration of audio contained in each label is dependent on the stride of the underlying acoustic model; the model used has a stride of 8. The model is trained with word-pieces using the recipe outlined in [23].

S1.1.2. SNR

The Signal-to-noise (SNR) ratio is calculated using the VAD labels predicted above. For each 80ms frame the VAD will return a posterior of whether the frame is speech or not. We decided to take < 0.8 as the speech threshold, and > 0.995 as noise threshold. If a speech frame is detected, we also automatically include 2 subsequent frames, to compensate for the spiky predictions from the VAD model. Any other frames that does not belong to either bucket are ignored since we are not confident whether they are speech or noise. Finally we compute the SNR ratio by using its definition $\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10}\left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}\right)$
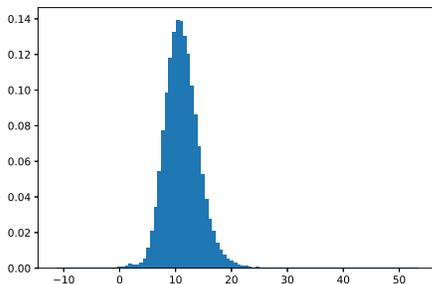

Fig. S1: **Librivox SNR histogram**

S1.1.3. Perplexity

Perplexity was obtained by performing beam search decoding of the trained TSD model defined above supplemented by a 4gram word Language Model trained on LibriSpeech LM. It was computed as the mean of the log probability of the posterior on each file.

S1.1.4. JSON files and splits

To construct the JSON, we duplicated the metadata of the original book JSON file from LibriVox into each of the files associated for a given book (including unique book ID and speaker ID, and we added tags for SNR, perplexity and our own macro-genre tags, by folding the existing ones into 7 categories: "Literature", "Science, Craft & Essay", "Ancient", "Religion", "Poetry", "Theater", and "Undefined". We also added VAD information as a list of onsets and offsets of voice activity.

The files were splitted into cuts of different sizes by trying to maintain the same distribution of macro-genres in the three cuts.

S1.2. The ABX metric

Given a frame-wise distance metric, we want to check that features coding for the same phonemes have a more similar representation than features coding for different phonemes. To quantify this property, we use a minimal pair ABX task as defined in [14]; given a set of sounds $S(\boldsymbol{x})$ from a category $\boldsymbol{x}$ and a set of sounds $S(\boldsymbol{y})$ from a category $\boldsymbol{y}$ we compute:

$$\boldsymbol{\theta}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{nm(m-1)} \sum_{a \in S(\boldsymbol{x})} \sum_{b \in S(\boldsymbol{y})} \sum_{c \in S(\boldsymbol{x}) \setminus \{a\}} \hat{\theta}(a, b, c)$$

With:

$$\hat{\theta}(a, b, c) = \mathbb{1}_{d(a,c) < d(a,b)} + \frac{1}{2} \mathbb{1}_{d(a,c) = d(a,b)}$$

$$m, n = |S(\boldsymbol{x})|, |S(\boldsymbol{y})|$$

Here, $\boldsymbol{\theta}(\boldsymbol{x}, \boldsymbol{y})$ is the probability of a sample from category $\boldsymbol{x}$ to be closer to another element from x than to one from $\boldsymbol{y}$. To compute ABX we average the error $1 - \boldsymbol{\theta}$ over all categories.

S2. SUPPLEMENTARY BASELINE MODELS

S2.1. CPC model and training

To train a feature model in an unsupervised fashion, we used the method implemented by Riviere and al. in [?], inspired by the Contrastive Predictive Coding algorithm (CPC) presented in [7]. We will briefly introduce the algorithm in this section, though we refer the reader to the original papers for more details.

S2.1.1. Contrastive Predictive Coding

CPC relies on forward modeling: given an input sequence of features, we try to predict the k future representations of the sequence. The network must discriminate each future ground truth feature from negative examples randomly sampled in the batch.

More precisely, the model goes like this:

1. The raw waveform w goes through a convolutional network g_c, resulting in a feature sequence $(\mathbf{x}_t)_{t\in 1...T}$.

2. Then we form the current phoneme representation $\mathbf{z}_t$ by applying a recurrent network g_{ar} to $\mathbf{x}_t$.

3. Finally, we predict $(\mathbf{x}_{t_0+1}, ..., \mathbf{x}_{t_0+k})$ from $(\mathbf{z}_t)_{t\leq t_0}$ using a prediction network g_p.

When using theses feature for another task we always consider $\mathbf{z}_t$, the output of the recurrent layer.

S2.1.2. Architecture Details

For g_c, we use five convolutional layers with strides [5, 4, 2, 2, 2], filter-sizes [10, 8, 4, 4, 4] and 256 hidden units with ReLU activations. Besides, the features are normalized channelwise between each convolution. In the end, this network has a downsampling factor of 160, meaning that on a 16kH audio input each feature will encode 10ms of data. Furthermore, g_{ar} is a one-layer LSTM with also a 256 dimensional hidden state. Finally, the predictor g_p is a one-layer transformer.

S2.1.3. Training Details

We considered input sequences of 1280ms gathered in batches of 32 sequences per GPU with a total of 128 GPUs. Our training took approximatly two days on NVIDIA Tesla V100-SXM2-16GB. Besides, in a given batch, all sequences were sampled within the same speaker.

S2.2. TDS model and training

Given the limited amount of supervised training data, we select to use a smaller TDS model [23] with 20 million parameters. The model has a stride 2 in the first convolution, and three groups of TDS blocks with channels (10, 14, 18) and (2, 2, 3) blocks in each group. While on the whole 60k hours training data, we use the original architecture introduced in [23] with 37 million parameters. The only difference is that we reduce the overall stride from 8 to 2. We use dropout to prevent over-fitting, and its value is set to 0.4 and 0.1 in the 20M and 37M models.

In terms of model optimization, we use plain SGD with momentum. The initial learning rate and momentum are set to 0.1 and 0.5 respectively. In the supervised setting, the models are trained for 1500 epochs on 8 GPUs in total with learning rate halved after each 200 epochs and total batch size 64

and 16 for character- and phone- based system. In the semi-supervised scenario, the models are trained for 150 epochs on 32 GPUs in total with learning rate halved after every 30 epochs and total batch size 256.

The beam-search decoding parameters are tuned on only dev-other for the 20M TDS models to generate pseudo-labels, while they are tuned independently for the final 37M model. The same official LibriSpeech 4-gram LM is used in both decoding procedures. The decoding beam size is 1000 in all the experiments.

S3. SUPPLEMENTARY RESULTS

S3.1. Pseudo-labels experiment

System	dev-clean	dev-other	test-clean	test-other
MFSC TDS + train-1h	44.4	57.7	55.6	65.9
+ 60k pseudo-label	57.6	68.1	59.5	72.3
MFSC TDS + train-10h	22.5	40.2	22.2	41.3
+ 60k pseudo-label	**18.7**	**36.0**	**18.7**	**38.6**
MFSC TDS + train-1h	44.3	53.9	46.9	55.4
+ 60k pseudo-label	43.4	52.6	43.2	53.9
MFSC TDS + train-10h	20.7	36.4	21.8	38.0
+ 60k pseudo-label	**15.0**	**31.1**	**14.7**	**32.2**

Table S1: **PER/CER of acoustic models trained in with pseudo-labels.** *Top*: small phone-based TDS [23] models with limited labels using wav2letter++[18], generating pseudolabels on the 60K dataset with an in-domain LM, retraining a larger TDS acoustic model (PER). *Bottom*: the same trained on characters (CER).

Table S1 provides PER/CER in the distant supervision setting with models trained on pseudo-labels. The TDS models above and generated pseudo-labels are trained and generated with the exactly the same procedure introduced in Section 4. Note that the PER/CER results above are not comparable to the semi-supervised ones in Table 3 as pseudo-labels here are generated with the official LibriSpeech LM, whose training set is a super-set of the transcriptions in the supervised training set.