Federated Representation Learning for Automatic Speech Recognition

Guruprasad V Ramesh[1*], *Gopinath Chennupati*[2], *Milind Rao*[2], *Anit Kumar Sahu*[2], *Ariya Rastrow*[2], *Jasha Droppo*[2]

[1]University of Wisconsin-Madison
[2]Amazon Alexa USA

viswanathanr@wisc.edu, chennug@amazon.com, milinrao@amazon.com, anitsah@amazon.com, arastrow@amazon.com, drojasha@amazon.com

Abstract

Federated Learning (FL) offers a privacy-preserving approach to model training, allowing edge devices to learn collaboratively without sharing data. Edge devices like Alexa and Siri are prospective sources of unlabeled audio data that can be tapped to learn robust audio representations. In this work, we bring Self-supervised Learning (SSL) and FL together to learn representations for Automatic Speech Recognition respecting data privacy constraints. We use the speaker and chapter information in the unlabeled speech dataset, Libri-Light, to simulate non-IID speaker-siloed data distributions and pre-train an LSTM encoder with the Contrastive Predictive Coding framework with FedSGD. We show that the pre-trained ASR encoder in FL performs as well as a centrally pre-trained model and produces an improvement of 12-15% (WER) compared to no pre-training. We further adapt the federated pre-trained models to a new language, French, and show a 20% (WER) improvement over no pre-training.

Index Terms: representation learning, automatic speech recognition, federated learning, self-supervised learning

1. Introduction

Federated Learning (FL) [1, 2] offers collaborative training of models on decentralized data distributed across multiple devices. In FL, model updates from the participating devices are shared to a central server without compromising on user data privacy, as personal data remains intact on client devices.

A wealth of speech data is available on client devices such as smartphones and voice assistants like Alexa and Siri. This data can produce robust speech models for ASR and other downstream speech tasks. The audio data is unlabeled in nature due to the lack of reliable transcripts, which is more challenging in FL, to learn from this data in supervised fashion. Alternatively, we can learn robust representations using self-supervised learning (SSL), which are later fine-tuned with limited transcribed data. SSL attempts in speech [3, 4, 5, 6, 7] show the efficacy of such a two-stage training strategy. Here, we exploit FL, which offers the privacy of audio on the devices while producing effective models when combined with SSL.

In this paper, we combine SSL and FL, to learn speech representations for ASR. Our strategy is two-stage: first, we use the 50K hours of unlabeled monolingual (English) speech corpus, Libri-Light [8] to pre-train a Recurrent Neural Network Transducer (RNN-T) [9] encoder with the FL algorithm, FedSGD [1]; second, we fine-tune RNN-T (with the above pre-trained encoder) on a limited amount of transcribed audio data.

*Work done during internship at Amazon

We simulate non-identical and independently distributed (non-IID) data using the speaker and chapter information of the utterances. Hereafter, this data setup is referred to as *speaker-siloed* data, which is used to pretrain the LSTM encoder in a Contrastive Predictive Coding (CPC)[3] framework across a set of clients using FL. We show that the federated pre-trained models are similar in performance to that of the centrally pre-trained models. We further show the efficacy of FL pre-trained models in adapting to a foreign language, French.

2. Related Work

There are a handful of attempts in literature for applying FL in speech-related tasks. Some of these applications are: ASR [10, 11, 12], Keyword Spotting [13, 14], Emotion Recognition [15, 16], and Speaker Verification [17]. Notably, for combining FL with SSL, the only available works include Federated self-supervised learning (FSSL) [18] for acoustic event detection and [19], where the challenges involved in combining FL & SSL due to hardware limitations on the client are surfaced and a wav2vec 2.0 [4] model is trained with FL on Common-Voice Italian data [20] and fine-tuned for ASR.

In this paper, we study the use of FL for producing acoustic representations using Libri-Light 50K+ hours of audio data. Our contributions are:

- To the best of our knowledge, we are the first to integrate FL for ASR at scale that produces competitive downstream ASR performance compared to regular pretraining.
- Adapting monolingual federated pretrained models to a resource-constrained target language (French) resulting in an improvement of 20% relative WER.

3. Methodology

3.1. Generating speaker-siloed data

Generally, speech corpora (especially ones acquired from audiobooks) have an inherent non-IID nature as the data acquired comes from different speakers with different acoustic characteristics and content. The Libri-Light data contains speakerID and chapterID information embedded in the unique identifier of each utterance. The non-IID partitions are created as follows:

1. Given a speech corpus, silo them based on the speakerID.
2. Sort the siloed utterances based on the chapter information, serving as a proxy for temporal distribution.

This *speaker-siloed* data is used in the federated pre-training experiments. Additionally, during pre-training, we ensure that the batches generated during training are from the same speaker. Another non-trivial difference between federated and centralized pre-training is, in FL we process data once as

opposed to the possibility of multiple passes in centralized pre-training. This strict constraint is to simulate the non-availability of data on the clients after a certain time threshold. More details on device-siloed data are in Appendix B.

3.2. Stage 1: Pre-training

We compare traditional centralized pre-training with the proposed federated pretraining. **Central pre-training**: Our baseline is a model trained using the standard centralized CPC framework as outlined in Appendix A.1. Here, data is gathered and shuffled at the cloud eliminating speaker-specific distributions and running multiple epochs to learn the representations. **Federated pre-training**: Here, we pre-train the model with

Algorithm 1 Federated pretraining with CPC

Input: $\mathbf{K}$ number of clients participating in each round, $\mathbf{B}$ client batch size, $\mathbf{E}$ number of local update steps and $\mathbf{C}$ number of batches per client per update-step. $\mathbf{E}$=1,$\mathbf{C}$=1 for FedSGD
Require: Data Generator $\mathbf{D_k}$ for each client $\mathbf{k}$, groups utterances into batches based on speaker.
Server: randomly initializes weights for model, $\mathbf{w}^0$.
for round $t = 0, 1, 2, ...$ **do**
  Server sends $\mathbf{w}^t$ to the $\mathbf{K}$ clients selected randomly.
  for clients $k \in [1, \mathbf{K}]$ **do**
    $\mathbf{D_k}$ generates $\mathbf{C}$ batches of utterances of max-size $\mathbf{B}$
    Update a local CPC model for $\mathbf{E}$ steps on the $\mathbf{C}$ batches
    Send the updated weights $\mathbf{w}_k^t$ to the server
  end for
  Server aggregates weights and updates its model:
$$\mathbf{w}^{t+1} \leftarrow \sum_{k=1}^{\mathbf{K}} \frac{n_k}{n} \mathbf{w_k^t}$$
end for

speaker-siloed data using the FL algorithm FedSGD. Note that the hyperparameters (non-FL specific) and the network architecture are the same as central pre-training. Each of the clients involved in a round of federated pre-training updates the model based on local data and sends back weight updates to a central server. The central server accumulates and aggregates the weights and broadcasts the updated model to the participating clients of the next round (Algorithm 1). Unlabelled client data is used only once in the training process as indefinite retention of data is infeasible, especially on resource-constraint devices.

3.3. Stage 2: Fine-tuning

To assess the performance of the pre-trained models, both centralized and federated, we initialize the RNN-T encoder with the pre-trained model. The RNN-T encoder in our setup follows the same architecture as the CPC model. The prediction network and joint network are randomly initialized. All layers are centrally fine-tuned with transcribed audio data.

4. Experiments

4.1. Datasets

We use Libri-Light (LL) dataset [8] for our pre-training experiments. LL contains three parts[1]: small, medium, and large, and for our purposes, we consider the 'large' portion, which consists of a substantial $52,000$ hours of unlabeled speech data generated from 6845 speakers. In the centralized pre-training experiments, we use all the 52K hours of the data, for FL, the same data is arranged in non-IID format, *speaker-siloed*, for pre-training. We further study the impact of the amount of data used in FL pre-training, for that, we use, $5K$ hours ($LL-5K$), a randomly selected subset from LL. In fine-tuning the pre-trained models, we use the 960-hour train partition of the Librispeech [21]. The fine-tuned models are evaluated on the dev and test sets of Librispeech. In addition to these, we use the French data from the Multilingual Librispeech (MLS) dataset [22] in our resource-constrained language adaptation experiments. Table 1 summarizes the datasets.

[1]Libri-Light data preparation and download, `https://github.com/facebookresearch/libri-light/tree/main/data_preparation`

Table 1: *Datasets used. For Librispeech, the clean and other splits are mentioned for dev and test*

Dataset	Duration(in hours)		
	Train	Dev (clean, other)	Test (clean, other)
Libri-Light Large	51934	NA	NA
Librispeech	960.9	5.4, 5.4	5.3, 5.1
Multilingual Librispeech French	1076.6	10	10

4.2. Training Details

The CPC (RNN-T encoder) models in centralized and federated pre-training both consist of a 3-layer 512-unit feed-forward feature encoder with ReLU activation and a 6-layer 1024-unit unidirectional LSTM context encoder. The RNN-T model also includes a 1024-unit 2-layer prediction network and a single dense layer joint network. The inputs for both pre-training and fine-tuning models are constructed from 256-dimensional STFT features obtained through a 25ms window and 10ms frame shift, combined into a final 768-dimensional feature by concatenating three consecutive frames. All pre-training experiments are done with random initialization of the weights. The pre-trained model weights are used to initialize the RNN-T encoder in the ASR fine-tuning stage, and the prediction and joint networks are randomly initialized. The ASR model is trained with a 2500 sentence-piece [23] vocabulary. We used 48 V100 GPUs for training the models. During pre-training, in centralized training, we ran for $130K$ steps with a bucket batch size of $[64, 32, 1]$. In FL, we vary the number of physical GPUs for the federated experiments based on the number of clients per training round involved, roughly keeping two clients per GPU. The FedSGD models are trained for 22k rounds or one pass of the entire data. On each client, SGD with a unit learning rate is used, and Adam is applied at the server with a learning rate of $1e^{-5}$. Fine-tuning on Librispeech is run for $100K$ steps.

5. Results

5.1. Federated versus Central Pre-training

We show the efficacy of pre-trained (FL/central) audio representations. The RNN-T model is fine-tuned (with the pre-trained encoder) on the Librispeech 960 hour train data. Table 2 shows the WER of the fine-tuned models on the *dev* and *test* partitions.

We compare the results of the fine-tuned models to an RNN-T model trained without any pre-training, *from-scratch*, where the encoder is randomly initialized. We also study the impact of the amount of data on pre-training with LL-5k and LL-52k datasets. We find that all the pre-trained models

Table 2: *Central pre-trained and Federated pre-trained models with RNN-T fine-tuning on train-*960 *hour Librispeech. WER results are on the dev and test sets of Librispeech. No pre-training model (trained*from-scratch *on train-960 hour) is the baseline. C refers to central pre-training, and FL refer to the Federated pre-trained models. The number next to FL indicates the number of clients used in each training round.*

Pre-training Setting	Fine-tuning Word Error Rate (WER)	
	Dev-Clean/Other	Test-Clean/Other
No pretraining	6.84/17.63	7.24/18.42
C, LL-5k	6.81/17.72	7.3/18.2
FL48,LL-5k	6.59/**17.02**	**6.80/17.42**
FL70,LL-5k	**6.43**/17.17	6.83/17.88
C, LL-52k	5.79/**16.29**	6.07/**16.21**
FL48,LL-52k	5.83/16.9	6.16/16.34
FL70,LL-52k	**5.77**/16.4	**6.05/16.21**

(FL/central) performed better than the *from-scratch* model. On average, the pre-trained models show a relative WER (WERR) improvement of 11.3% and 14.22% on *dev* and *test* sets, respectively. We observe pre-training on the larger dataset (LL-52k, see Table 2) is better in both the cases of FL and central pre-training, similar to [4, 5, 6] where centrally pre-trained models show better ASR performance with more data. We can reassure the above observation for FL pre-training to produce audio representations.

The key observations for FL vs. central pre-training include: i) the performance of the FL pre-trained models is similar to that of the central; ii) in fact, when pre-trained on a small amount of data (LL-5k), the FL pre-trained models perform better than central, average WERR on *dev* and *test* sets is 4.76% and 5.56%; iii) the observed superior/equal performance of the FL models is despite the single pass through the data in FL settings as opposed to multiple passes through the same data in central pre-training. We attribute the performance of the FL models to the hybrid approach of combining FL and SSL, which produces robust privacy-preserving speech representations that are useful for downstream ASR tasks. Finally, we experimented with 48 and 70 clients in FL during pre-training. There is an insignificant difference in the performance of the models between the two settings. However, the impact of number of clients on the performance of the speech models can be explored in future.

5.2. Adaptation of SSL representations

Self-supervised representation learning is beneficial for tasks with a shortage of labeled data. The adaptation of both multilingual and monolingual pre-trained models [24, 25, 26] demonstrated remarkable adaptability to other languages in speech. We explore the effectiveness of the speech representations learned from our approach (FL+SSL) in adapting to a new language, French. We use French data from Multilingual Librispeech [22], consists of 1070 hours of train data. The data is randomly split into two sets: 215 hours (*train-215*) and 855 hours (*train-855*).We conduct two sets of experiments: Direct fine-tuning and continued pre-training followed by fine-tuning. In direct fine-tuning, the pre-trained models from Libri-Light are directly fine-tuned using the *train-215* data. In the other setting, we continue pre-training the LL pre-trained models using the *train-815* unlabeled set and only then fine-tune them using the *train-215* data. Table 3 shows the results on the MLS French dev and test sets.

Table 3: *Results of the language adaptation experiments on train-215 hour MLS French data. We compare no pre-training, direct fine-tuning, and continued pre-raining followed by fine-tuning. Boldface stands for the best pre-trained models.*

Experiment	Source Pre-train Setting	Dev WER (WERR)	Test WER (WERR)
No pre-training	-	46.31	42.73
Direct fine-tuning	C, LL-52k	37.8(18.37)	34.6(19.02)
	FL48,LL-52k	37.82(18.33)	34.25(19.84)
	FL70,LL-5k	43.27(6.56)	39.69(7.11)
Continued pre-training + fine-tuning	C, LL-52k	**35.92(22.43)**	**32.6(23.70)**
	FL48,LL-52k	**37.63(18.74)**	**34.12(20.14)**
	FL70,LL-5k	41.48(10.42)	37.1(13.17)

Overall the pre-trained models perform much better than from-scratch randomly initialized models. The continued pre-training results are better than the direct fine-tuning experiments as models first adapt to the language and then to the ASR task. Another observation is that the amount of pre-training data plays a crucial role in the target language ASR performance in both the fine-tuning experiments (direct and continuous), see the significant difference in the performance of the LL-5k and LL-52k pre-trained models. FL pre-trained models are competitive with the central pre-trained models in adapting to another language when trained on unlabeled data.

6. Conclusion

We empirically demonstrated that FL models pre-trained in SSL style perform similarly to the centralized pre-training for the downstream ASR tasks. We employed the Contrastive Predictive Coding (CPC) framework with FedSGD at scale on a large unlabeled monolingual speech corpus, Libri-Light. The FedSGD pre-trained models also adapt to a new language, where continued pre-training on domain-specific language improves performance. In conclusion, we suggest the inflection of traditional central pre-training of audio representations to FL based pre-training to be as effective as the central case.

In the future, we plan to extend the work to broader speech corpora, such as multilingual audio datasets and closer to real-life federated speech corpora, explore more recent SSL frameworks based on self-attention, and characterize the impact of various FL settings, such as the number of clients participating in a training round and the impact of various FL algorithms.

7. References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

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

[4] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[5] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[6] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[7] A. Mohamed, H.-y. Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaløe *et al.*, "Self-supervised speech representation learning: A review," *arXiv preprint arXiv:2205.10643*, 2022.

[8] J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen *et al.*, "Libri-light: A benchmark for asr with limited or no supervision," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7669–7673.

[9] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[10] D. Dimitriadis, R. G. Ken'ichi Kumatani, R. Gmyr, Y. Gaur, and S. E. Eskimez, "A federated approach in training acoustic models." in *Interspeech*, 2020, pp. 981–985.

[11] D. Guliani, F. Beaufays, and G. Motta, "Training speech recognition models with federated learning: A quality/cost framework," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3080–3084.

[12] Y. Gao, T. Parcollet, S. Zaiem, J. Fernandez-Marques, P. P. de Gusmao, D. J. Beutel, and N. D. Lane, "End-to-end speech recognition from federated acoustic models," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7227–7231.

[13] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2019, pp. 6341–6345.

[14] T. Zhang, T. Feng, S. Alam, S. Lee, M. Zhang, S. S. Narayanan, and S. Avestimehr, "Fedaudio: A federated learning benchmark for audio tasks," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[15] S. Latif, S. Khalifa, R. Rana, and R. Jurdak, "Federated learning for speech emotion recognition applications," in *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2020, pp. 341–342.

[16] T. Feng and S. Narayanan, "Semi-fedser: Semi-supervised learning for speech emotion recognition on federated learning using multiview pseudo-labeling," *arXiv preprint arXiv:2203.08810*, 2022.

[17] F. Granqvist, M. Seigel, R. Van Dalen, A. Cahill, S. Shum, and M. Paulik, "Improving on-device speaker verification using federated learning with privacy," *arXiv preprint arXiv:2008.02651*, 2020.

[18] M. Feng, C.-C. Kao, Q. Tang, M. Sun, V. Rozgic, S. Matsoukas, and C. Wang, "Federated self-supervised learning for acoustic event classification," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 481–485.

[19] Y. Gao, J. Fernandez-Marques, T. Parcollet, A. Mehrotra, and N. D. Lane, "Federated self-supervised speech representations: Are we there yet?" *arXiv preprint arXiv:2204.02804*, 2022.

[20] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[21] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

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

[23] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.

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

[25] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino *et al.*, "Xls-r: Self-supervised cross-lingual speech representation learning at scale," *arXiv preprint arXiv:2111.09296*, 2021.

[26] S. Khurana, A. Laurent, and J. Glass, "Magic dust for cross-lingual adaptation of monolingual wav2vec-2.0," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6647–6651.

[27] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.

[28] W. Yu, J. Freiwald, S. Tewes, F. Huennemeyer, and D. Kolossa, "Federated learning in asr: Not as easy as you think," in *Speech Communication; 14th ITG Conference*. VDE, 2021, pp. 1–5.

A. Background

A.1. Contrastive Predictive Coding (CPC)

CPC [3] architecture (see Figure 1) consists of two components, the feature encoder, and the context encoder. The feature encoder f_{enc} maps the speech sequence vectors $X = [x_1, x_2, ..., x_T]$ into latent representations $Z = [z_1, z_2, ..., z_T]$. The context encoder f_{ar} is autoregressive, creating a contextualized representation c_t for each time-step t based on the current and previous latent representations, $z_t, z_{t-1}, z_{t-2}, ...$ The pretext task of CPC is to predict a set of K future latent steps for every $c_t \ \forall t \in [1, T-K]$ with an objective to maximize the mutual information between the context c_t and the future latent steps $z_{t+k} \forall k \in [1, K]$. The entire model is optimized using a contrastive loss function L_C based on Noise Contrastive Estimation [27]. This loss, referred to as InfoNCE, is expressed as follows:

$$L_C = \sum_{k=1}^{K} \frac{-1}{(T-k)} \sum_{t=1}^{T-k} \log \frac{e^{z_{t+k}^T (W_k c_t + b_k)/\kappa}}{\sum_{\hat{z} \in \mathbf{Z}} e^{\hat{z}^T (W_k c_t + b_k)/\kappa}},$$

where $\mathbf{Z}$ is a collection of the true latent z_{t+k} and N-1 negatives sampled from other time steps in the same audio, and κ is the temperature term. The future latent prediction layer consists of weights W_k and biases $b_k \ \forall k \in [1, K]$, used to predict the future latent steps. This latent prediction layer is only used in the pretraining stage. The ease of implementation and light compute footprint, compared to more recent pretraining frameworks, make CPC a great candidate for integrating it into an FL setup.

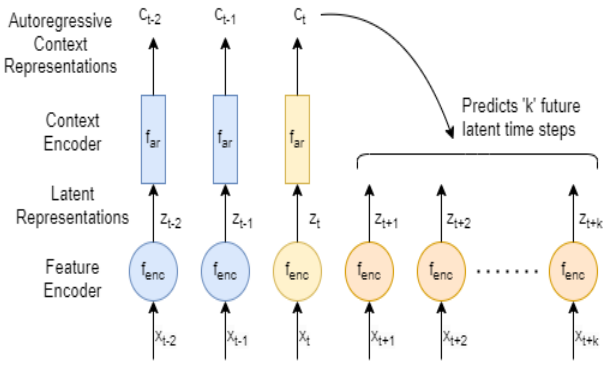

Figure 1: *Pretraining Architecture with CPC algorithm*

A.2. Recurrent Neural Network - Transducer (RNN-T)

A standard RNN-T [9] architecture (refer to figure 2) consists of an encoder, a prediction network, and a joint network. The encoder processes the input speech signal $X = [x_1, x_2, ..., x_T]$ and generates a frame-wise hidden representation h_t . The prediction network takes the previous non-blank tokens y_{u-1} in the output sequence predicted thus far and maps it to a hidden representation r_u. The joint network combines the hidden representations of the encoder and prediction network into a softmax normalized conditional distribution over the target vocabulary augmented with the blank token.

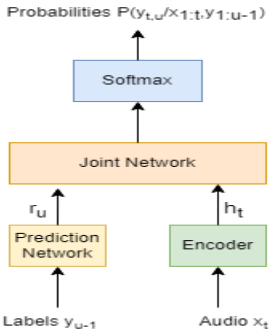

Figure 2: *Standard RNN-T architecture*

B. Generating speaker-siloed data for Libri-Light

One of the challenges in training models in a federated setup is the diverse data distribution among the participants. In the context of speech assistants, the data in a client device might predominantly belong to one speaker and have a specific linguistic and acoustic profile. Mimicking a similar setup using an open-source dataset like Libri-Light (refer to Figure 3) posed a challenge. Previous research attempts to integrate open-source audio data in a federated setup involved splitting the dataset based on the speaker information [19, 11, 28] but does not include a temporal notion for the data. Here we additionally include a temporal notion by treating chapterID as a unit of time. Thus, after splitting the data based on the speaker information, we also use the chapter number and book name together to sort the speaker-grouped utterances. We fix a maximum batch size of 8, and for each client, ensure only the same speaker's utterances are grouped together during federated training (pretraining). To understand this better, let there be three clients, **C1**, **C2**, **C3**, during a round of federated pertaining. Each of them contains batches that have only one speaker's utterances. For example, **C1** will contain utterances from Speaker **S1**-Chapter1, **C2** will contain utterances from Speaker **S2**-Chapter1, and so on. As Libri-Light's speaker-to-utterance ratio is not uniform, the addition of this temporal notion using the chapter information ensures that different clients in a round contain data from different speakers and more closely represent a real-life scenario of federated training of speech models.

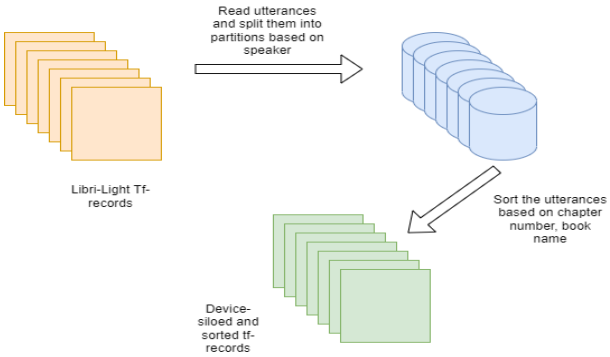

Figure 3: *Steps in obtaining speaker-siloed Libri-Light data for Federated pretraining*