A Complementary Joint Training Approach Using Unpaired Speech and Text for Low-Resource Automatic Speech Recognition

Ye-Qian Du¹, Jie Zhang¹, Qiu-Shi Zhu¹, Li-Rong Dai¹, Ming-Hui Wu², Xin Fang², Zhou-Wang Yang¹

¹University of Science and Technology of China
²iFlytek Research

{duyeqian,qszhu}@mail.ustc.edu.cn, {jzhang6,lrdai,yangzw}@ustc.edu.cn

Abstract
Unpaired data has shown to be beneficial for low-resource automatic speech recognition (ASR), which can be involved in the design of hybrid models with multi-task training or language model dependent pre-training. In this work, we leverage unpaired data to train a general sequence-to-sequence model. Unpaired speech and text are used in the form of data pairs by generating the corresponding missing parts in prior to model training. Inspired by the complementarity of speech-PseudoLabel pair and SynthesizedAudio-text pair in both acoustic features and linguistic features, we propose a complementary joint training (CJT) method that trains a model alternatively with two data pairs. Furthermore, label masking for pseudo-labels and gradient restriction for synthesized audio are proposed to further cope with the deviations from real data, termed as CJT++. Experimental results show that compared to speech-only training, the proposed basic CJT achieves great performance improvements on clean/other test sets, and the CJT++ re-training yields further performance enhancements. It is also apparent that the proposed method outperforms the wav2vec2.0 model with the same model size and beam size, particularly in extreme low-resource cases.

Index Terms: automatic speech recognition, low-resource, semi-supervised learning, speech synthesis, pseudo-label

1. Introduction

The end-to-end (E2E) architecture remains the dominant paradigm for automatic speech recognition (ASR). This single network structure allows for a simpler training process and joint optimization compared to conventional models, and it achieves impressive performances [1, 2, 3, 4]. Nevertheless, it requires a large amount of labeled data for training, which is rather expensive and time-consuming in terms of data collection, resulting in obstruction in the development of low-resource tasks. In contrast, speech-only and text-only data are broadly available. Thus, the focus of this work is on how to make use of unpaired data for low-resource ASR.

There has been extensive research on the utilization of unpaired data. For speech-only data, the common approach is unsupervised training that serves as a feature extractor for downstream ASR tasks [5, 6, 7, 8], or self-training with pseudo-labels following a typical teacher-student training scheme [9, 10]. For text-only data, text is mainly used to train an external language model (LM) for joint decoding [11, 12, 13, 14, 15]. In order to make use of both unpaired speech and text, many methods have recently been proposed, e.g., integration of a pre-trained acoustic model and LM [16, 17, 18, 19], cycle-consistency based dual-training [20, 21, 22, 23], and shared representation learning [24, 25, 26, 27], which rely on hybrid models with multi-task training and some of which become less effective in cases with a very limited amount of labeled data. The current mainstream methods that achieve state-of-the-art (SOTA) results in low-resource ASR use unpaired speech and text for pre-training and training a LM for joint decoding, respectively [28, 29], and adopt an additional iterative self-training [29]. However, these methods require a large beam search space to fully exploit the capability of the LM, leading to a heavy computational cost for decoding or self-training.

In order to leverage unpaired speech and text to train a general sequence-to-sequence (Seq2Seq) model, partial pre-training [29, 30] can be applied, while it was shown that the inconsistency between pre-training and fine-tuning might limit the model performance. To avoid such problem, in this work, we instead train the ASR model using sample pairs, which are generated by pseudo-labeling and Text-to-Speech (TTS) synthesis prior to model training. As in low-resource scenarios, the generated data often largely differs from real data, so a single utilization of speech-PseudoLabel (speech-PsL) pairs or SynthesizedAudio-text (SynA-text) pairs could seriously mislead the model training. Due to the fact that these two kinds of data pairs are complementary in terms of both input acoustic features and output linguistic features, we propose to alternatively train the model on both data pairs, and we refer to this method as complementary joint training (CJT). This CJT method is employed as the first round training and is thus called basic CJT. Based on the analysis of basic CJT, two strategies are proposed for further performance enhancement. Specifically, for pseudo-labels we mask the tokens with a low first-round confidence, and for synthesized audio we proportionally block the gradient back propagation to lower layers to better fit real audio. These two strategies are involved in the second round training, referred to as CJT++.

The proposed CJT method is validated via experiments on the LibriSpeech dataset [31] and a Transformer [32] network with limited computational resources. Experimental results on

Figure 1: The diagram of the proposed CJT method, where \((x, y)\) is the speech-PseL pair that is generated by the ASR model fine-tuned on labeled data, and \((x^*, y)\) is the SynA-text pair generated by a TTS model.
the 10min labeled data show that the basic CJT reduces the word error rate (WER) by around 35%/21% on clean/other sets compared to speech-only training, and the CJT++ re-training further reduces the WER by around 28%/13%, i.e., an overall 53%/31% reduction. It is also shown that on three low-resource data splits, the proposed method decreases the WER by 55%/41%/28% on average on 10min/1h/10h labeled data compared to the wav2vec 2.0 model under the same modest resource data splits, the proposed method decreases the WER by around 35%/21% on clean/other sets further reduces the WER by around 28%/13% , i.e., an over-

2. Complementary Joint Training

The training process of the proposed CJT method is shown in Figure 1. After data preparation, the model is first jointly trained (basic CJT), as described in Section 2.1, and then re-trained with label masking and gradient restriction (CJT++), as proposed in Section 2.2 after an empirical analysis.

2.1. Basic complementary joint training

For the basic CJT, the abundant unpaired speech and text are used for training, and the small amount of paired data is only used for data preparation. Let the paired speech-text be denoted as $D_p = \{ (x^{(i)}, y^{(i)}) \}_{i=1}^{N}$, the unpaired speech and unpaired text as $D_{u}^{s} = \{ (x^{(i)}) \}_{i=1}^{N}$ and $D_{u}^{t} = \{ (y^{(i)}) \}_{i=1}^{N}$, respectively. For an unpaired speech sample $x \in D_{u}^{s}$, we generate the corresponding pseudo-label $y^{*} = \text{ASR}(y^{(i)})(x)$ by using the ASR model fine-tuned on the paired data $D_p$. The set of speech-PseL pairs is denoted as $D_{u}^{s} = \{ (x^{(i)}, y^{*}) \}_{i=1}^{N}$. For an unpaired text sample $y \in D_{u}^{t}$, we synthesize the corresponding audio $x^{*} = \text{TTS}(y)$ with a TTS model. The set of SynA-text pairs is denoted as $D_{u}^{t} = \{ (x^{*}, y^{(i)}) \}_{i=1}^{N}$.

The CJT model is alternatively updated on $D_{u}^{s}$ and $D_{u}^{t}$, where the joint training target is given by

$$ L = L_s + L_t, $$

where $\lambda$ is a balancing parameter, and $L_s$ and $L_t$ are losses of speech-PseL pairs and SynA-text pairs, respectively, given by

$$ L_s = -\mathbb{E}_{(x,y^{*})\in D_{u}^{s}} \log P(y^{*}|x), $$

$$ L_t = -\mathbb{E}_{(x^{*},y)\in D_{u}^{t}} \log P(y|x^{*}). $$

For an unpaired speech sample $x \in D_{u}^{s}$, we generate the corresponding pseudo-label $y^{*} = \text{ASR}(y^{(i)})(x)$ by using the ASR model fine-tuned on the paired data $D_p$. The set of speech-PseL pairs is denoted as $D_{u}^{s} = \{ (x^{(i)}, y^{*}) \}_{i=1}^{N}$. For an unpaired text sample $y \in D_{u}^{t}$, we synthesize the corresponding audio $x^{*} = \text{TTS}(y)$ with a TTS model. The set of SynA-text pairs is denoted as $D_{u}^{t} = \{ (x^{*}, y^{(i)}) \}_{i=1}^{N}$.

The potential benefit of it is twofold: preventing overfitting to incorrect labels and an enhancing context modeling due to an absence of historical information.

For a pseudo-label sequence $y^{*} = (y^{*}_1, y^{*}_2, \cdots, y^{*}_T)$, where $T$ is the length of the target sequence, we generate a binary mask sequence and accordingly replace some of the tokens with $<\text{PAD}>$. Let the masked target sequence be denoted as $\tilde{y}^{*}$ and the set of mask indexes as $\mathcal{M}$, respectively. The loss of speech-PseL pairs in the second-round training becomes

$$ L'_s = -\sum_{(x,y^{*})\in D_{u}^{s}} \mathcal{M} \sum_{i=1}^{T} \log P(y^{*}_i|x, \tilde{y}^{*}_{t-1}), $$

where $\mathcal{M}$ represents the set of mask indexes. The loss of speech-PseL pairs in the second-round training becomes

$$ L'_s = -\sum_{(x,y^{*})\in D_{u}^{s}} \mathcal{M} \sum_{i=1}^{T} \log P(y^{*}_i|x, \tilde{y}^{*}_{t-1}), $$

(4)

Given the predicted probability $p$ of a token from the first-round prediction, three masking approaches are considered: 1)
confidence-driven masking \( (\text{conf}) \), where the token is randomly masked with the probability of \( k + (1 - p) \) with \( k \) denoting a multiplier; 2) threshold-based masking \( (\text{thres}) \), where the token is masked if \( p < P_{\text{thres}} \) with \( P_{\text{thres}} \) denoting the threshold determined by the percentile of probabilities; 3) random masking \( (\text{rand}) \), where the token is randomly masked with a fixed probability. The overall masking probability is empirically set as a multiple of the pseudo-label error rate.

2.2.2. Gradient restriction for synthesized audio

Due to the fact that the TTS synthesized audio exhibits smaller variations than real audio, the utilization of synthesized data might degrade the ASR performance on real speech; however, after joint training with real audio, the model can largely recover the ability of acoustic feature extraction (e.g., see Figure 3). Although synthesized audio causes only small deviations, we propose a gradient restriction training strategy to further reduce feature mismatches.

In this strategy, we randomly block gradient propagation to the shallow layers when training on synthesized audio at a certain probability so that the model can better fit real audio when extracting acoustic features. In this work, the first four layers are regarded as the shallow layers.

3. Performance Evaluation

3.1. Experimental setup

All experiments are performed on the LibriSpeech corpus \([31]\). The unpaired speech originates from the LibriSpeech training data, which contains 960 hours of speech with transcriptions removed. The unpaired text comes from the standard pre-processed LibriSpeech LM corpus without over-lapping transcripts, which contains about 80 times the amount of text in the audio transcriptions. Three Libri-light \([34]\) limited resource training subsets are used for the paired data, including train-10h (10 hours), train-1h (1 hour), and train-10min (10 minutes). Results are evaluated on dev-clean/other and test-clean/other sets.

For the ASR modeling, we use 80-dimensional log-Mel filterbank features. The modeling units in our experiments are 5000 word pieces. We choose the convolutional Transformer architecture \([32]\) as the backbone. This model (71M) is composed of an encoder that contains 2 2-D convolutional blocks \([32]\) followed by 12 Transformer blocks \([35]\), and a decoder that contains 4 1-D convolutional blocks \([32]\) followed by 6 Transformer blocks \([35]\). For each Transformer block, the attention dimension is 512 with 8 attention heads, and the inner dimension between layers is 2048. Besides, the self-attention is linearly warmed up, kept constant, and then exponentially decayed for 10%, 40% and 50% of the updates, respectively. Models on 100h/960h speech data are trained for 80K/250K updates. In case the model is jointly trained with text data, it trains for double updates. For the second-round training, we train for half the time of the first-round training. For regularization, we use a single dropout rate of 0.15 \([39]\) across all Transformer blocks and 0.1 label smoothing \([40]\). We also apply SpecAugment \([38]\) for data augmentation. The final model used for evaluation is calculated by averaging the last 10 checkpoints.

For language modeling, a Transformer-based LM consisting of 6 decoder blocks is trained on the LibriSpeech LM corpus. It trains for 800K updates under almost the same training conditions as in the ASR model. It shares the same 5000 word pieces as output tokens for shallow fusion \([11]\). Finally, the LM has word-piece-level perplexity of 32 on the dev-clean set. For decoding, the LM weight for shallow fusion is set to 0.4, with a beam size of 20.

For the preparation of pseudo-labels, we follow the well-designed pre-training model wav2vec 2.0 \([7]\) to train a teacher model for pseudo-labeling. The 960h pre-trained wav2vec 2.0 BASE model \([7]\) is loaded and fine-tuned on three labeled data splits with CTC loss \([41]\). Then, we pseudo-label the 960h unpaired speech by the fine-tuned models combined with a loaded word-level Transformer LM \([7]\) using a medium beam size of 20. The WERs of pseudo-labels of 10h, 1h, 10min labeled data are 4.96%, 7.96% and 17.10%, respectively.

For the preparation of synthesized audio, we use a ready-made TTS engine with 3 speakers \([31]\) for convenience, as the paired data that are less than 10 hours or even 10 minutes are hard to train a robust TTS model. Only 1/10 of the LM corpus is randomly selected for audio synthesis.

Notice that all experiments are implemented in the fairseq framework \([42]\). Training and decoding hyper-parameters are barely tuned for better possible performance.

3.2. Results

First of all, in order to validate the effectiveness of the basic CJT method with different amounts of data and updating ratios, we show the ASR performance in terms of WER for the proposed CJT, the models trained on a single type of data and the oracle model trained with ground-truth transcriptions in Table 1. We see that the WERs of 100h speech-PseL + 100h SynA-text are comparable to that of 200h speech-PseL (although slightly higher on noisy(other) set). If the amount of SynA-text is increased to 860h, the performance is further improved, e.g., even very close to the WER of 960h speech-PseL on the clean set, but obviously worse on the noisy(other) set. Among the three

### Table 1: Performance of the joint training with different amounts of unlabeled data and various updating ratios \((1: \lambda)\) using 10min of labeled data, where 200h of unpaired speech consists of the train-clean-100 set and 100h of the rest clean data.

| Unpaired Data | Dev WER | Test WER |
|---------------|---------|---------|
|               | clean   | other   | clean   | other   |
| speech-PseL   |         |         |         |         |
| SynA-text     |         |         |         |         |
| 100h          | -       | -       | 25.18   | 40.20   |
| 200h          | -       | -       | 18.88   | 31.19   |
| 960h          | -       | -       | 15.47   | 23.37   |
| -             | 100h    | -       | 92.99   | 95.47   |
| -             | 860h    | -       | 92.27   | 96.01   |
| 100h          | 100h    | 1:1     | 18.68   | 33.66   |
| 100h          | 860h    | 1:1     | 17.29   | 32.14   |
| 100h          | 860h    | 1:3     | 16.31   | 32.16   |
| 100h          | 860h    | 1:5     | 15.96   | 32.24   |
| 100h (speech-text) | -       | 13.90   | 30.46   |

https://github.com/pytorch/fairseq/tree/main/examples/wav2vec
https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_small.pt
https://dl.fbaipublicfiles.com/wav2letter/sota/2019/lm/lm_librispeech_word_transformer.pt
Refer to https://ttsvoice.iflysec.com/
Table 2: Performance of the proposed second-round training strategies with various settings, following the training in Table 1. All results are obtained by ASR-only greedy decoding.

| Method          | Dev WER | Test WER |
|-----------------|---------|----------|
|                 | clean   | other    | clean   | other    |
| basic CJT       | 16.31   | 32.16    | 16.72   | 32.64    |
| + PseLM-rand(p=0.4) | 15.57   | 31.15    | 16.02   | 32.22    |
| + PseLM-conf(p=0.4) | 12.93   | 28.92    | 13.17   | 29.73    |
| + PseLM-thres(p=0.16) | 12.85   | 28.84    | 13.44   | 29.57    |
| + PseLM-thres(p=0.4) | 11.84   | 28.29    | 12.05   | 29.52    |
| + PseLM-thres(p=0.8) | 11.67   | 30.08    | 12.08   | 31.49    |
| + SynGR-all      | 16.46   | 30.71    | 16.76   | 31.54    |
| + SynGR-shallow  | 16.27   | 30.24    | 16.57   | 31.23    |
| + PseLM-thres(p=0.4) & SynGR-shallow | 11.74   | 27.74    | 12.01   | 28.83    |

Table 3: A comparison of WERs on Librispeech with the wav2vec 2.0 BASE model with the same beam size of 20, where "*" stands for self-implementation. The models are trained on 3 Libri-light low-resource data splits, using the 960h untranscribed LibriSpeech data and the LM corpus as unpaired data.

| Method          | LM      | Dev WER | Test WER |
|-----------------|---------|---------|----------|
|                 | clean   | other   | clean   | other   |
| 10min paired    |         |         |         |         |
| wav2vec 2.0    | -       | 46.1    | 51.5    | 46.9    | 50.9    |
| wav2vec 2.0 *  | -       | 47.43   | 53.56   | 48.31   | 53.21   |
| wav2vec 2.0 * Transf. | 17.62   | 26.00   | 17.80   | 25.46   |
| basic CJT      | 13.22   | 19.54   | 13.20   | 19.75   |
| + CJT++        | Transf. | 10.02   | 16.32   | 10.14   | 16.61   |
| + LM           | 6.90    | 12.51   | 7.36    | 12.96   |
| 1h paired      |         |         |         |         |
| wav2vec 2.0    | -       | 24.1    | 29.6    | 24.5    | 29.7    |
| wav2vec 2.0 *  | -       | 18.21   | 25.67   | 18.86   | 26.29   |
| wav2vec 2.0 * Transf. | 7.67    | 14.42   | 7.77    | 14.80   |
| basic CJT      | 6.44    | 12.36   | 6.45    | 12.78   |
| + CJT++        | Transf. | 5.68    | 11.58   | 5.66    | 12.19   |
| + LM           | 4.25    | 8.74    | 4.25    | 9.44    |
| 10h paired     |         |         |         |         |
| wav2vec 2.0    | -       | 10.9    | 17.4    | 11.1    | 17.6    |
| wav2vec 2.0 *  | -       | 9.51    | 17.00   | 9.76    | 17.30   |
| wav2vec 2.0 * Transf. | 4.64    | 10.59   | 4.62    | 10.68   |
| basic CJT      | 4.40    | 10.20   | 4.44    | 10.64   |
| + CJT++        | Transf. | 4.13    | 9.77    | 4.22    | 10.34   |
| + LM           | 3.03    | 7.66    | 3.42    | 8.30    |

5. Acknowledgements

We would like to thank IFLYTEK CO. LTD. for providing computational resources and a TTS engine. This work is supported by Anhui Center for Applied Mathematics, the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDC 08010100), the NSFC of China (No. 11871447), and the National Natural Science Foundation of China (No. 62101523).
6. References

[1] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in ICASSP, 2016.

[2] L. Dong, S. Xu, and B. Xu, “Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition,” in ICASSP, 2018.

[3] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen et al., “State-of-the-art speech recognition with sequence-to-sequence models,” in ICASSP, 2018.

[4] Y. Zhang, W. Chan, and N. Jaitly, “Very deep convolutional networks for end-to-end speech recognition,” in ICASSP, 2017.

[5] A. Van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” arXiv, 2018.

[6] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised pre-training for speech recognition,” in Interspeech, 2019.

[7] 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, 2020.

[8] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhota, 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, 2021.

[9] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, “Iterative pseudo-labeling for speech recognition,” in Interspeech, 2020.

[10] T. Likhomanenko, Q. Xu, J. Kahn, G. Synnaeve, and R. Collobert, “SLIMIPL: Language-model-free iterative pseudo-labeling,” arXiv, 2020.

[11] J. Chorowski and N. Jaitly, “Towards better decoding and language model integration in sequence to sequence models,” in Interspeech, 2017.

[12] A. Sriram, H. Jun, S. Satheesh, and A. Coates, “Cold fusion: Training seq2seq models together with language models,” in Interspeech, 2018.

[13] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, “An analysis of incorporating an external language model into a sequence-to-sequence model,” in ICASSP, 2018.

[14] J. Shin, Y. Lee, and K. Jung, “Effective sentence scoring method using BERT for speech recognition,” in Proceedings of The Eleventh Asian Conference on Machine Learning, 2019.

[15] J. Salazar, D. Liang, T. Q. Nguyen, and K. Kirchhoff, “Masked language model scoring,” in Annual Meeting of the Association for Computational Linguistics, 2020.

[16] C. Yi, S. Zhou, and B. Xu, “Efficiently fusing pretrained acoustic and linguistic encoders for low-resource speech recognition,” IEEE Signal Processing Letters, 2021.

[17] G. Zheng, Y. Xiao, K. Gong, P. Zhou, X. Liang, and L. Lin, “WavBERT: Cooperative acoustic and linguistic representation learning for low-resource speech recognition,” in Findings of the Association for Computational Linguistics: EMNLP, 2021.

[18] K. Deng, S. Cao, Y. Zhang, and L. Ma, “Improving hybrid ctc/attention end-to-end speech recognition with pretrained acoustic and language model,” arXiv, 2021.

[19] Y.-A. Chung, C. Zhu, and M. Zeng, “SPLAT: Speech-language joint pre-training for spoken language understanding,” in NAACL-HLT, 2021.

[20] A. Tjandra, S. Sakki, and S. Nakamura, “Listening while speaking: Speech chain by deep learning,” in IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2017.

[21] Y. Ren, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “Almost unsupervised text to speech and automatic speech recognition,” in International Conference on Machine Learning, 2019.

[22] S. Karita, S. Watanabe, T. Iwata, M. Delcroix, A. Ogawa, and T. Nakatani, “Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders,” in ICASSP, 2019.

[23] T. Horl, R. Astudillo, T. Hayashi, Y. Zhang, S. Watanabe, and J. Le Roux, “Cycle-consistency training for end-to-end speech recognition,” in ICASSP, 2019.

[24] A. Renduchintala, S. Ding, M. Wiesner, and S. Watanabe, “Multi-modal data augmentation for end-to-end ASR,” in Interspeech, 2018.

[25] J. Drexler and J. Glass, “Combining end-to-end and adversarial training for low-resource speech recognition,” in IEEE Spoken Language Technology Workshop (SLT), 2018.

[26] S. Karita, S. Watanabe, T. Iwata, A. Ogawa, and M. Delcroix, “Semi-supervised end-to-end speech recognition,” in Interspeech, 2018.

[27] J. Ao, R. Wang, L. Zhou, S. Liu, S. Ren, Y. Wu, T. Ko, Q. Li, Y. Zhang, Z. Wei et al., “SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing,” arXiv, 2021.

[28] Q. Xu, A. Baevski, T. Likhomanenko, P. Tomasello, A. Conneau et al., “Self-training and pre-training are complementary for speech recognition,” in ICASSP, 2021.

[29] Z. Fan, S. Zhou, and B. Xu, “Unsupervised pre-training for sequence to sequence speech recognition,” arXiv, 2019.

[30] C. Gao, G. Cheng, R. Yang, H. Zhu, P. Zhang, and Y. Yan, “Pre-training transformer decoder for end-to-end ASR model with unpaired text data,” in ICASSP, 2021.

[31] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in ICASSP, 2015.

[32] A. Mohamed, D. Okhonko, and L. Zettlemoyer, “Transformers with convolutional context for ASR,” arXiv, 2019.

[33] A. Morcos, M. Raghu, and S. Bengio, “Insights on representational similarity in neural networks with canonical correlation,” Advances in Neural Information Processing Systems, 2018.

[34] J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu et al., “Libri-light: A benchmark for ASR with limited or no supervision,” in ICASSP, 2020.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones et al., “Attention is all you need,” Advances in Neural Information Processing Systems, 2017.

[36] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in NAACL-HLT, 2018.

[37] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv, 2014.

[38] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” Interspeech, 2019.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” The journal of machine learning research, 2014.

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Re-thinking the inception architecture for computer vision,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[41] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in international conference on Machine learning, 2006.

[42] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, “Fairseq: A fast, extensible toolkit for sequence modeling,” NAACL, 2019.