Interspeech 2024
1-5 September 2024, Kos, Greece

Self-Supervised Learning for ASR Pre-Training with Uniquely Determined Target Labels and Controlling Cepstrum Truncation for Speech Augmentation

Akihiro Kato, Hiroyuki Nagano, Kohei Chike, Masaki Nose

RICOH, Japan

{akihiro.kato, hiroyuki.nagano, kohei.chike, masaki.nose}@jp.ricoh.com

Abstract

To utilize a pre-trained large-scale model is an effective choice to develop automatic speech recognition (ASR) at limited data conditions. However, if we try pre-training with supervised manner, it causes high costs, specifically for transcription. To tackle this problem, recent research has presented self-supervised learning and it has successfully performed at ASR tasks. For further improvement, we study a new approach to self-supervised learning for ASR including methods for generating self-supervised labels and data augmentation.

Experimental results on Libri-Light and LibriSpeech corpora without any external language models demonstrate that our proposed method outperforms non pre-trained Conformer at limited data conditions in terms of character error rate (CER). Furthermore, the proposed method also exhibits comparable performance to HuBERT, which is one of the state-of-the-art model for self-supervised representation learning.

Index Terms: speech recognition, self-supervised learning, representation learning, data augmentation, conformer

1. Introduction

In recent years, end-to-end models for automatic speech recognition (ASR) have become a major option for research and they have brought great advances to this field [1, 2, 3]. Specifically, Connectionist Temporal Classification (CTC) [4, 5, 6] and attention-based models [7, 8, 9] are widely adopted to enable end-to-end ASR. In addition, Transformer [10, 11, 12] has brought remarkable achievement in the speech and language field of research. Transformer is multi-layered networks each layer of which comprises multi-head self-attention blocks and feed-forward blocks. It has demonstrated significant advantages in deepening layers and parallelizing computation at training. Convolution-Augmented Transformer (Conformer) [13, 14] has applied convolutional networks (CNNs) between the multi-head self-attention module and the feed-forward module to successfully capture both the global and local dependencies from an acoustic sequence as an ASR encoder.

To give such models higher learning capacity, it is effective to stack more layers [15, 16, 17]. However, the bigger networks, the larger dataset we have to feed to avoid overfitting [18, 19]. SpecAugment [20, 21, 22] has proposed a simple and effective data augmentation policies to give masking blocks and time-warping into the log-mel spectrogram. It has enabled ASR to exploit bigger networks by switching the problem from overfitting to underfitting.

On the other hand, recent research using a self-supervised learning [23, 24] by applying masked language model to pre-train an ASR model has also drown a lot of attention since Bidirectional Encoder Representation from Transformers (BERT) [25] and Generative Pre-trained Transformer (GPT) [26] successfully established themselves as generative language models. For instance, wav2vec 2.0 [27] applies Gumbel-Softmax [28] to carry out contrastive learning between contextualized representations from Transformer outputs and quantized feature vectors. Hidden-Unit BERT (HuBERT) [29, 30] has successfully removed the complicated processes related to the contrastive learning from wav2vec 2.0. To this end, HuBERT has an offline process to train a k-means quantizer that generates the target labels to which the Transformer outputs corresponding to the masked feature vectors are classified during the pre-training. The k-means quantizer, however, detaches the k-means modeling process from the end-to-end model and it could leave us other effort when we try to pre-train our own model. Besides this, the codewords representing the target labels span the subspace within the training set for k-means, therefore, it could not be suitable for additional model updates with different datasets. BERT-based Speech Pre-training with Random-projection Quantizer (BEST-RQ) [31] uses random projections and a random initialized codebook, therefore, the arrange of the codewords are independent of the dataset distribution. However, it does not guarantee appropriate sparseness of codewords, and what the Transformer blocks learn during pre-training is also uncertain.

To sum up our contribution, we aim at developing state-of-the-art pre-trained model that enables us to develop ASR with less transcribed speech. To this end, we propose a *novel method for self-supervised labeling* and we also study an effective approach to *data augmentation*. These techniques bring the following benefit at the model training compared to the existing methods.

- The input acoustic feature vectors can uniquely determine their class labels by a simple deterministic algorithm (i.e. no needs for contrastive learning or k-means clustering).
- The codewords for quantization can uniformly span whole vector space (i.e. the codewords are independent of the distribution of datasets).
- Data augmentation varies the harmonic and noise elements while it retains the formants (i.e. it produces unchanged utterances with various intelligibility without additional noise).

2. Proposed Method

2.1. Overview of model pre-training

As Figure 1 illustrates, we use Conformer as an encoder of our ASR model. The structure and settings of the Conformer encoder follow the original large-sized Conformer, i.e. Conformer (L) [13]. We feed 80-dimension log-mel spectrogram, $\mathbf{X} \in \mathbb{R}^{80\times T}$, into the Conformer encoder. Then the projection

10.21437/Interspeech.2024-1933

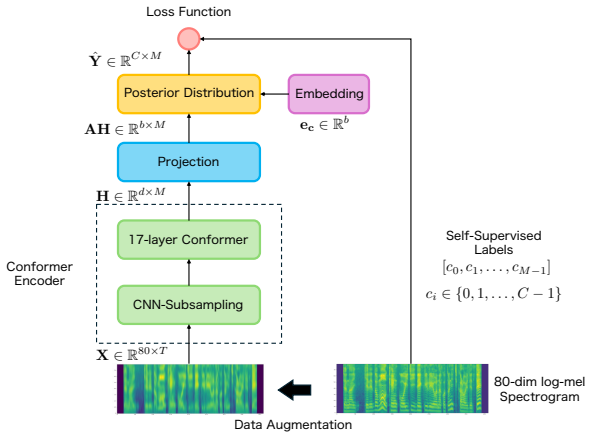

Figure 1: *Our pre-training model consists of the Conformer encoder and a projection layer. The model transforms an 80-dimensional log-mel spectrogram into space of class embeddings to compute the posterior distribution over the classes which is then evaluated with the self-supervised labels.*

layer projects the output sequence of the Conformer encoder, $\mathbf{H} \in \mathbb{R}^{d\times M}$, into the space of class embeddings, $\mathbf{e}_c \in \mathbb{R}^b$, to derive posterior distribution over classes, $p(c \mid \mathbf{X})$, where $c \in \{c_0, c_1, \ldots, c_{C-1}\}$. The posterior distribution is then evaluated by a loss function referring to self-supervised labels, $\mathbf{c}$.

$$\mathbf{X} = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{T-1}], \quad \mathbf{x}_i \in \mathbb{R}^{80}, \tag{1}$$

$$\mathbf{H} = \text{Encoder}(\mathbf{X}) = [\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{M-1}], \quad \mathbf{h}_i \in \mathbb{R}^{d}, \tag{2}$$

$$\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_{M-1}], \quad \hat{\mathbf{y}}_i \in \mathbb{R}^{C}, \tag{3}$$

$$\hat{\mathbf{y}}_i = [p(c_0 \mid \mathbf{x}_i), \ldots, p(c_{C-1} \mid \mathbf{x}_i)]^\top, \tag{4}$$

$$p(c_k \mid \mathbf{x}_i) = \frac{\exp\left(\text{sim}\left(A\mathbf{h}_i, \mathbf{e}_k\right)/\tau\right)}{\sum_{c=0}^{C-1} \exp\left(\text{sim}\left(A\mathbf{h}_i, \mathbf{e}_c\right)/\tau\right)}, \tag{5}$$

where $\mathbf{x}_i$, $\mathbf{h}_i$ and $\hat{\mathbf{y}}_i$ denote the i-th vectors of the sequences at the inputs, Conformer encoder outputs and the posterior distribution respectively. d and C are the dimension of the Conformer encoder and the number of the target classes respectively while T and M denote the number of time steps of $\mathbf{X}$ and time steps after subsampling at the Conformer encoder respectively. $\text{sim}(\cdot,\cdot)$, A and τ represent the cosine similarity, projection matrix and scale factor respectively. $\mathbf{e}_c$ is initialized as uniform distribution and trained simultaneously with the encoder parameters.

2.2. Uniquely determined self-supervised labels

We pre-train our ASR model with speech dataset having *no transcripts*. Thus, we make their target labels to which the model tries to classify each frame of the input sequence instead of the text labels during pre-training. To this end, we propose a simple method to uniquely determine the target labels corresponding to $\mathbf{X}$. We first apply discrete cosine transform (DCT) to $\mathbf{X}$ to obtain sequence of mel frequency cepstral coefficients (MFCCs), $\mathbf{U}$.

$$\mathbf{U} = \text{DCT}(\mathbf{X}) = [\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{T-1}], \quad \mathbf{u}_i \in \mathbb{R}^{80}, \tag{6}$$

where $\mathbf{u}_i$ represents an 80-dimensional MFCCs. We then extract only the first n coefficients from $\mathbf{U}$. In other words, we truncate coefficients after the n-th order and the energy coefficient to extract vector sequence, $\mathbf{V}$, as

$$\mathbf{V} = [\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{T-1}], \quad \mathbf{v}_i \in \mathbb{R}^{n}, \tag{7}$$

where $\mathbf{v}_i$ represents a subspace vector of $\mathbf{u}_i$. After that, we normalize $\mathbf{V}$ to constrain its dynamic range.

$$\tilde{\mathbf{V}} = \frac{\mathbf{V} - \boldsymbol{\mu}}{\mathbf{s}} = [\tilde{\mathbf{v}}_0, \tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_{T-1}] \tag{8}$$

$$\boldsymbol{\mu} = \frac{1}{T}\sum_{i=0}^{T-1} \mathbf{v}_i, \tag{9}$$

$$\mathbf{s} = \sqrt{\frac{1}{T}\sum_{i=0}^{T-1} (\mathbf{v}_i - \boldsymbol{\mu})^2}. \tag{10}$$

Then we quantize $\tilde{\mathbf{V}}$ according to base β number with thresholds, ξ_γ, where $\gamma \in \{1, \ldots, \beta-1\}$.

$$\bar{\mathbf{V}} = [\bar{\mathbf{v}}_0, \bar{\mathbf{v}}_1, \ldots, \bar{\mathbf{v}}_{T-1}] \tag{11}$$

$$\bar{\mathbf{v}}_i = \left[\bar{v}_i^{(0)}, \bar{v}_i^{(1)}, \ldots, \bar{v}_i^{(n-1)}\right]^\top \tag{12}$$

$$\bar{v}_i^{(k)} = \begin{cases} \beta-1 & \text{as } \bar{\mathbf{v}}_i^{(k)} \geq \xi_{\beta-1} \\ \beta-2 & \text{as } \xi_{\beta-1} > \bar{\mathbf{v}}_i^{(k)} > \xi_{\beta-2} \\ \vdots & \\ 0 & \text{as } \bar{\mathbf{v}}_i^{(k)} < \xi_1 \end{cases} \tag{13}$$

In the case of $\beta = 2$ (i.e. binary), we can set, for example, $\xi_1 = 0.0$ while $\xi_1 = -0.6$ and $\xi_2 = 0.6$ in the case of $\beta = 3$ (i.e. ternary).

Since $\bar{\mathbf{V}}$ is quantized as base β digits, we can obtain corresponding self-supervised target labels, $\mathbf{c}$, by converting it to decimal integer as

$$\mathbf{c} = [c_0, c_1, \ldots, c_{T-1}]^\top, \tag{14}$$

$$c_i = \sum_{k=1}^{n} \beta^{k-1} \cdot \bar{v}_i^{(k)} \in \{0, 1, \ldots, \beta^n - 1\}, \tag{15}$$

If we set β and dimension of $\mathbf{v}_i$ to, for example, $\beta = 2$ and $n = 10$ respectively, we can obtain 1,024-class self-supervised labels. We then apply downsampling to $\mathbf{c}$ in order to match the sequence steps of $\hat{\mathbf{Y}}$ and $\mathbf{c}$.

$$\mathbf{c}' = \left[c'_0, c'_1, \ldots, c'_{M-1}\right]^\top, \tag{16}$$

$$c'_i = c_{pi+q} \tag{17}$$

where we determine p and q according to the kernel size and stride of CNN-subsampling layers in the Conformer encoder.

The preceding manner of quantization to derive $\bar{\mathbf{V}}$ is very simple. Figures 2 (a) and (b), however, depict $\bar{\mathbf{V}}$ still retaining the characteristic patterns in $\mathbf{V}$ after quantization. Besides this, Figures 2 (c) and (d) represent log-mel spectrograms retransformed back from $\mathbf{V}$ and $\bar{\mathbf{V}}$ respectively, and they demonstrate that *the quantized vectors still keep most of formant information* in $\mathbf{X}$. Therefore, we expect that our self-supervised learning rationally enables the model to operate an acoustic representation learning. The retransform above is done by inverse DCT after recovering the energy coefficient and zero padding to the truncated coefficients.

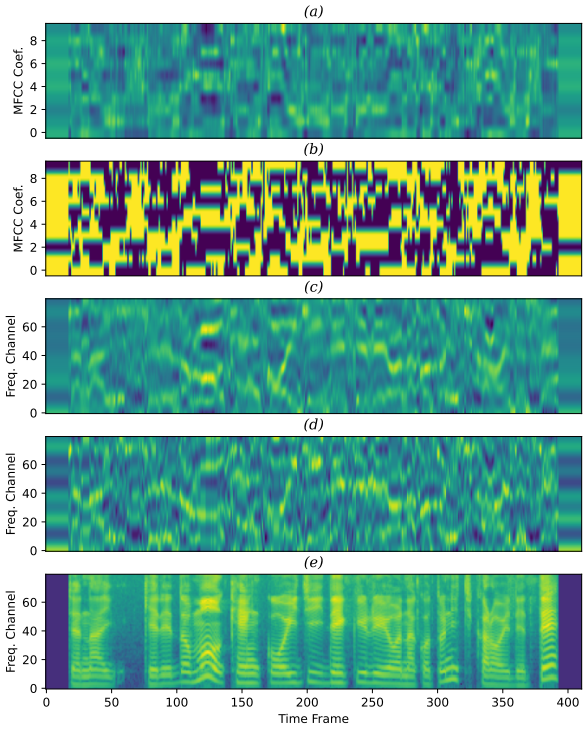

Figure 2: *MFCCs and log-mel spectrograms extracted from female speech. (a) represents 10-dimensional MFCCs as* $\mathbf{V}$ *while (b) shows the quantized version as* $\bar{\mathbf{V}}$*. (c) and (d) are log-mel spectrograms retransformed back from (a) and (b) respectively. (e) illustrates the original log-mel spectrogram as* $\mathbf{X}$*.*

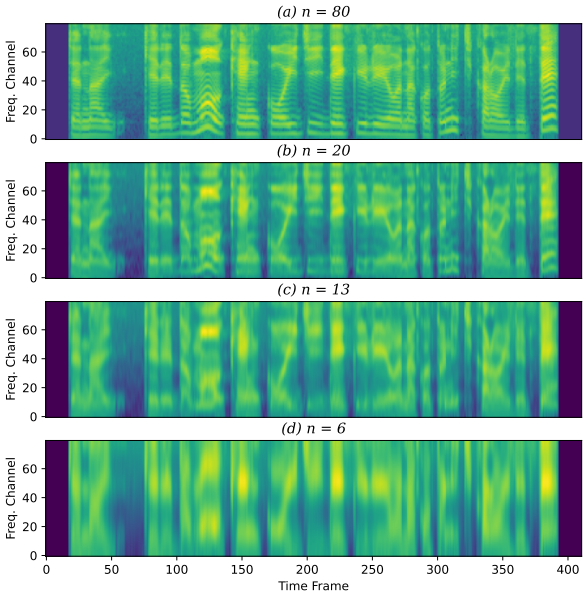

Figure 3: *Augmented log-mel spectrograms of female speech. (a), (b), (c) and (d) are augmented with* n *= 80, 20, 13 and 6 respectively.*

2.3. Data augmentation

To prevent the model from overfitting and, moreover, to make the model more robust to low intelligibility speech, we apply data augmentation to input speech during both pre-training and fine-tuning. To this end, we first extract 80-dimensional log-mel spectrogram, $\mathbf{X}$, from input speech as shown by Equation 1. Secondly, we apply DCT to $\mathbf{X}$ to derive sequence of 80-dimensional MFCCs, $\mathbf{U}$, as Equation 6 shows. Next we randomly select positive integer, n, that is less than or equal to the MFCC dimension and truncate $\mathbf{U}$ to the n-th order coefficient, $\mathbf{V}$, as Equation 7 represents. We then retransform $\mathbf{V}$ to log-mel spectrogram by inverse DCT (IDCT) after zero padding to the truncated coefficients.

Figure 3 (a), (b), (c), and (d) illustrate the log-mel spectrograms augmented with n = 80 (i.e. identical to the original), 20, 13 and 6 respectively. Our proposed method augments speech by discarding some information from the original whereas it tries to retain the elements of formants. The amount of information loss is determined by n that is randomly given. Consequently, it varies intelligibility of speech with random intensity, specifically, it produces a lot of variation in harmonic elements. It potentially makes the model more robust to speech intelligibility and avoid overfitting to specific speakers in training datasets. For easier description, we name this data augmentation method **Con**trolling **Cep**strum **T**runcation for Speech **Augment**ation (Concept-Augment).

3. Experiments

We examine the proposed method by first pre-training the Conformer encoder model with our approach to self-supervised representation learning. We then fine-tune the pre-trained model with supervised learning to evaluate the performance in terms of ASR accuracy. For ASR tasks, we do NOT use any external language model to evaluate our method simply, and we gauge the performance in terms of character error rate (CER).

3.1. Datasets

For self-supervised learning, we use Libri-light [32] that includes 60,000-hour English speech recordings without transcription. The dataset consists of three subsets, namely small (577 hours), medium (5193 hours) and large (51,934 hours), and we combine these three subsets (*LL-60k*) to use full 60,000 hours for training.

On the other hand, we use LibriSpeech [33], which contains 970-hour transcribed English speech recordings, for supervised learning. As the training set, 960 hours of speech (*LS-960*) consists of three subsets, namely train-clean-100 (*LS-100*), train-clean360 (*LS-360*) and train-other500 (*LS-500*) while subset, *dev-other*, is used as the validation set during training. We also use a 10-hour portion of speech from LibriSpeech as the limited supervision training sets (*LS-10*). For the evaluation of ASR accuracy after fine-tuning, we use *test-clean* and *test-other* as the test sets.

Speech of the datasets are framed with Hann window into 25ms frames at 10ms intervals and then transformed to an 80-dimensional log-mel spectrogram. We use a single GPU where the mini-batch size is set to at most 300 seconds.

3.2. Model

The overview of our model is represented by Figure 1. The architecture and hyperparameters of the Conformer encoder fol-

lows the original Conformer (L) [13]. The projection layer is a single linear layer in order to map 512-dimensional outputs of the encoder onto b-dimensional embedding space at which the posterior distribution over the classes are estimated without any language models.

3.3. Concept-Augment

We apply the proposed data augmentation (Concept-Augment) at pre-training and fine-tuning. Coefficient truncation order, n, is randomly chosen as $n \in [6, 80]$ during training. To examine the performance of Concept-Augment, we carry out a preliminary experiment which trains the Conformer encoder and a linear 29-class[1] CTC greedy decoder with supervised learning on LibriSpeech (*LS-960*). We compare the learning curves between three different conditions, namely without augmentation, with SpecAugment [20] and with Concept-Augment. For SpecAugment, we set frequency mask parameter, F, to 27, number of frequency mask, m_F, to 2, maximum-size of the time mask, T, to 0.05 $\times$ utterance length and number of time mask, m_T, to 10. Time warping is not used.

For the training settings, we use Adam optimizer [34] with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-8}$. The learning rate is scheduled with 100,000 linear warmup steps from 10^{-8} to 10^{-4}, 450,000 hold steps and 600,000 exponential decay steps from 10^{-4} to 10^{-6} and then, held at 10^{-6}.

Figure 4 depicts Concept-Augment having the best convergence characteristics of the three different conditions showing the lowest CER of 7.36 % on *dev-other* during training. The

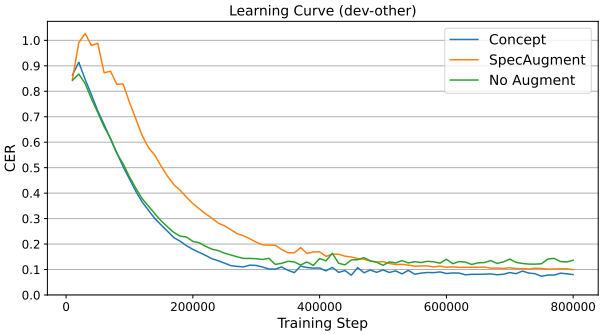

Figure 4: *The learning curves at the conditions of No augmentation, SpecAugment and Concept-Augment.*

CER scores are not state-of-the-art performance because we do NOT use any language models and besides, our decoder is a simple CTC greedy decoder with a single linear layer. Nonetheless, this result confirms that the proposed data augmentation method is sufficiently effective as data augmentation for this size of network models.

3.4. Pre-training

To pre-train the model, we apply the proposed labeling method during self-supervised learning with Concept-Augment. We pre-train the model on Libri-light (*LL-60k*) with the same optimizer and scheduler settings as ones described in Section 3.3. For this pre-training, we set scale factor, τ, in Equation 5 to 0.1, apply 256 to embedding dimension, b, and the number of classes C is 729 since we set coefficient truncation order, n, to 6 with base 3 quantization for labeling. Quantization thresholds, ξ_1 and ξ_2, are set to -0.6 and 0.6 respectively. We apply masks to the output sequence from the CNN-subsampling with the same manner as HuBERT [29] where mask ratio p=0.22 and mask span l=3.

We use the cross-entropy loss between the self-supervised labels and the posterior distribution derived by Equation 5 as the objective function. The loss calculation is applied only to the masked frames.

3.5. Fine-tuning

After pre-training with the proposed method, we fine-tune the model with supervised learning on LibriSpeech. To evaluate the performance of pre-trained model on limited training data, we use each of *LS-10*, *LS-100* and *LS-960* as the training sets. For fine-tuning, we reinitialize the embedding parameters of the pre-trained model as uniform distribution to retrain 29 embedding vectors for CTC decoding. The settings for the optimizer and the scheduler are same as the ones used in Section 3.3. Only the projection layer and the embeddings are updated during the first 10,000 steps of fine-tuning and the CNN parameters are frozen during the fine-tuning.

Table 1 presents the test results on *test-clean* and *test-other* of LibriSpeech with the limited training data settings. It compares CER with the baseline Conformer (L) without pre-training and HuBERT LARGE, which is a state-of-the-art pre-trained model consisting of much larger scale of network parameters than our proposed model. The CER scores are not

Table 1: *Character error rate (CER) after fine-tuning at limited training data conditions.*

	Model Params	Unlabeled Data	Labeled Data	test-clean [%]	test-other [%]
Proposed	118M	LL-60k		4.65	11.3
HuBERT	317M	LL-60k	LS-10	4.42	8.89
Conformer	118M	N/A	(10 hours)	-	-
Proposed	118M	LL-60k		3.22	6.44
HuBERT	317M	LL-60k	LS-100	2.92	5.08
Conformer	118M	N/A	(100 hours)	12.8	35.2
Proposed	118M	LL-60k		1.98	4.95
HuBERT	317M	LL-60k	LS-960	1.84	4.64
Conformer	118M	N/A	(960 hours)	3.56	7.45

state-of-the-art performance because we do NOT use any external language models and besides, we use simple CTC greedy decoding. However, the proposed method shows explicit effectiveness of pre-training at the limited labeled data conditions by comparing with Conformer without pre-training. Besides this, the proposed method also exhibits comparable performance to HuBERT LARGE in spite of the fact that the model size is approximately 38 % of HuBERT LARGE.

4. Conclusion

We studied the novel approach to self-supervised learning to pre-train ASR model with no transcribed speech datasets. It includes the labeling method to uniquely determine the target class corresponding to the input acoustic feature vectors and effective data augmentation algorithm.

Experimental results demonstrated that our proposed method outperformed the original Conformer at the limited training data conditions. In addition, the proposed method also exhibits comparable performance to the state-of-the-art method.

[1]For 26 letters of English alphabet, apostrophe, space and *blank*.

5. References

[1] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-end speech recognition: A survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[2] D. Wang, X. Wang, and S. Lv, "An overview of end-to-end automatic speech recognition," *Symmetry*, vol. 11, no. 8, p. 1018, 2019.

[3] L. Deng and X. Li, "Machine learning paradigms for speech recognition: An overview," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 1060–1089, 2013.

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[5] A. Graves, "Connectionist temporal classification," *Supervised sequence labelling with recurrent neural networks*, pp. 61–93, 2012.

[6] Y. Miao, M. Gowayyed, X. Na, T. Ko, F. Metze, and A. Waibel, "An empirical exploration of ctc acoustic models," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 2623–2627.

[7] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[8] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Advances in neural information processing systems*, vol. 28, 2015.

[9] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019.

[12] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*. IEEE, 2019, pp. 449–456.

[13] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Interspeech 2020*, 2020.

[14] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5874–5878.

[15] N.-Q. Pham, T.-S. Nguyen, J. Niehues, M. Müller, S. Stüker, and A. Waibel, "Very deep self-attention networks for end-to-end speech recognition," *Interspeech 2019*, 2019.

[16] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer, 2016, pp. 646–661.

[17] A. Bapna, M. X. Chen, O. Firat, Y. Cao, and Y. Wu, "Training deeper neural machine translation models with transparent attention," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3028–3033.

[18] X. Ying, "An overview of overfitting and its solutions," in *Journal of physics: Conference series*, vol. 1168. IOP Publishing, 2019, p. 022022.

[19] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[20] 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*, 2019.

[21] D. S. Park, Y. Zhang, C.-C. Chiu, Y. Chen, B. Li, W. Chan, Q. V. Le, and Y. Wu, "Specaugment on large scale datasets," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6879–6883.

[22] N. K. Singh, Y. J. Chanu, and H. Pangsatabam, "A study of various audio augmentation methods and their impact on automatic speech recognition," in *International Conference on Science, Technology and Engineering*. Springer, 2023, pp. 481–491.

[23] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[24] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," *Knowledge and Information systems*, vol. 42, pp. 245–284, 2015.

[25] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[26] OpenAI, "Gpt-4 technical report," *ArXiv*, vol. abs/2303.08774, 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[27] 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.

[28] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations*, 2016.

[29] 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.

[30] W.-N. Hsu, Y.-H. H. Tsai, B. Bolte, R. Salakhutdinov, and A. Mohamed, "Hubert: How much can a bad teacher benefit asr pre-training?" in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6533–6537.

[31] C.-C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *International Conference on Machine Learning*. PMLR, 2022, pp. 3915–3924.

[32] 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.

[33] 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.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980