Speech Representation Learning Combining Conformer CPC with Deep Cluster for the ZeroSpeech Challenge 2021

Takashi Maekaku[1], Xuankai Chang[2], Yuya Fujita[1], Li-Wei Chen[2], Shinji Watanabe[2],
Alexander Rudnicky[2]

[1]Yahoo Japan Corporation, Tokyo, Japan
[2]Carnegie Mellon University, PA, USA

{tmaekaku,yuyfujit}@yahoo-corp.jp,
{xuankaic,liweiche,alex.rudnicky,swatanab}@andrew.cmu.edu

Abstract

We present a system for the Zero Resource Speech Challenge 2021, which combines a Contrastive Predictive Coding (CPC) with deep cluster. In deep cluster, we first prepare pseudo-labels obtained by clustering the outputs of a CPC network with k-means. Then, we train an additional autoregressive model to classify the previously obtained pseudo-labels in a supervised manner. Phoneme discriminative representation is achieved by executing the second-round clustering with the outputs of the final layer of the autoregressive model. We show that replacing a Transformer layer with a Conformer layer leads to a further gain in a lexical metric. Experimental results show that a relative improvement of 35% in a phonetic metric, 1.5% in the lexical metric, and 2.3% in a syntactic metric are achieved compared to a baseline method of CPC-small which is trained on LibriSpeech 460h data. We achieve top results in this challenge with the syntactic metric.

Index Terms: contrastive predictive coding, deep cluster, conformer

1. Introduction

Many studies have shown that textual information is essential for building speech recognition systems and language models (LM). Recently, several important studies on representation learning [1, 2, 3, 4, 5] and semi-supervised training [6, 7, 8] explored using a large amount of speech data without corresponding text annotations and demonstrated significant improvements in speech recognition performance. This suggests that such systems may learn to train their own LM from raw audio only. Therefore, it is hoped that eventually spoken language modeling tasks can be done without any text annotations.

The Zero Resource Speech (ZeroSpeech) Challenge 2021 [9] is designed to tackle such unsupervised LM training using only raw speech data as input. The evaluation is done using a suite of 4 black-box, zero-shot metrics, which probe for the quality of the training models at 4 linguistic levels: phonetics, lexicon, syntax and semantics. The baseline system consists of three components: an acoustic model, a clustering module (k-means), and an LM. The acoustic model is built upon Contrastive Predictive Coding (CPC) [10], where the representation of the audio is learned by predicting the future frames using an autoregressive model. After training the CPC model, the baseline system trains a k-means clustering module on the outputs of the final layer of the autoregressive model to obtain sequences of discretized audio files. Finally, the LM is trained with the discretized units as pseudo-labels.

In this challenge, the final goal is to solve a couple of discrimination tasks. However, the representation obtained by the CPC model does not have sufficient linguistically discriminative characteristics since the CPC model itself is trained for the prediction task. To address this issue, we propose a method that combines the CPC model with a deep cluster method [11, 12, 13, 14]. We train an autoregressive model for phoneme classification using pseudo-labels obtained by clustering the outputs of a CPC network using k-means. The phoneme discriminative representation is obtained by doing a second-round clustering on the outputs of the final layer of the autoregressive model. Note that we call it phoneme classification in the sense of classifying pseudo-labels, which are likely to capture the phonetic meaning.

Furthermore, we examine replacing the Transformer [15] layer of the CPC model with a Conformer [16] layer. Conformer incorporates a convolutional neural network (CNN) [17] inside the Transformer to handle not only global but also local contexts, and its usefulness has been recognized in speech recognition tasks [16, 18, 19, 20]. Likewise, it is expected that more precise phonetic and lexical representation is achieved by capturing both contexts using the Conformer network. We apply the above two methods separately and confirm that both methods outperform the baseline method using the phonetic metric. In addition, we observe that the proposed method combining the Conformer CPC model with the deep cluster method outperforms the baseline method using the lexical metric. This reveals that the two methods have a complementary effect on both tasks.

2. Challenge Overview

In this section, we briefly introduce the baseline system and the task of the ZeroSpeech Challenge 2021 [9].

2.1. Baseline System

The baseline system consists of a speech representation learning model, a clustering model, and a language model. Fig 1 illustrates the architecture of the baseline system.

2.1.1. Contrastive Predictive Coding

The speech representation model is based on CPC, a self-supervised representation learning method proposed in [10]. Instead of using a conditional generative model to predict the future input signal, the CPC model learns the representation via maximizing the mutual information between the current context and future embeddings. The CPC model consists of two modules. First, given an input speech signal $\mathbf{x}$, a non-linear encoder g_{enc} maps it to a T-length sequence of embeddings with a lower time resolution: $\mathbf{z} = g_{enc}(\mathbf{x})$, where $\mathbf{z} = (z_1, \ldots, z_T)$. Then,
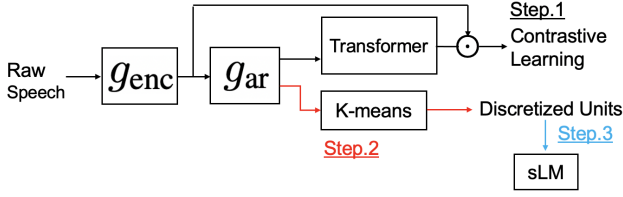
Figure 1: *Illustration of the baseline system. First, we train a Contrastive Predictive Coding (CPC) model which consists of g_{enc} and g_{ar} optimized by Eq. (1) (Step.1). Then, k-means clustering is performed to generate discretized units of audio data (Step.2). Finally, we train a spoken language model (sLM) using the discretized units as pseudo-labels (Step.3).*

an autoregressive encoder g_{ar} aggregates the information from $\mathbf{z}$, producing a context latent representation $c_t = g_{ar}(\mathbf{z}_{\leq t}), t \in \{1, \dots, T\}$. The CPC model is optimzied by minimizing the noise-contrastive estimation-based (NCE) loss [21]. At each time t, given the context representaion c_t and its K future embeddings $\{z_{t+k}\}_{1 \leq k \leq K}$, the loss is defined as:

$$\mathcal{L}_t = -\frac{1}{K} \sum_{k=1}^{K} \log \left[\frac{\exp(z_{t+k}^T h_k(c_t))}{\sum_{\tilde{z} \in \mathcal{N}_t} \exp(\tilde{z}^T h_k(c_t))} \right], \quad (1)$$

where $\mathcal{N}_t$ is a set of negative embedding samples and $h_k(\cdot)$ is a transformation for each step k. In this challenge, we use two different versions of CPC model: CPC-small and CPC-big, the differences of which are elaborated in Table 1.

2.1.2. Clustering and Language Models

To train a spoken language model (sLM) on pseudo-labels, the raw speech signal needs to be mapped to a sequence of discrete symbols. The pre-trained CPC model first generates a sequence of representations given the raw speech signal as input. Then, these representations are used to train a clustering model, which is k-means, with 50 clusters used in this work.

After training, the clustering model is applied to the speech representation of the training data to produce class labels. The class label can be regarded as a pseudo linguistic subword unit. Using these label sequences as pseudo-text data, we can train an sLM. In this work, we trained a BERT [15] language models. This model consists of multiple Transformer layers. Note that the BERT model is only trained with the masked language model objective, following [22]. Finally, the score of the language model on the pseudo-label sequence is regarded as a pseudo-probability (PP).

2.2. Dataset

The training data is comprised of the audio from the LibriSpeech 960h dataset [23] and the Libri-light dataset [24]. The CPC-small model is trained on the 100 hours of clean audio subset (train-clean-100) from the LibriSpeech data, while the CPC-big model is trained on a 6K-hour subset of Libri-light data. The k-means clustering is performed on the train-clean-100h subset to obtain the centroid coordinates. Then the k-means estimates the pseudo-label sequences on LibriSpeech 960h data, which becomes the training set for the language model.

Each of the four metrics is evaluated on its *dev* and *test* sets, which are specially designed for the corresponding task. Please refer to the challenge description [9] for more details of how the evaluation data are generated.

2.3. Evaluation Metrics

The performance of the spoken language model is evaluated using four different metrics, each corresponding to a task at a specific linguistic level: phonetics, lexicon, syntax and semantics.

Phonetics. The ABX metric [25] discriminates the speech sound between phonetic minimal pairs (e.g. "aba" and "apa"). Given the speech sounds a, x and b, where a and b are from two categories A and B ($A \neq B$), and x belongs to category A respectively, it computes the probability that the two sounds from the same category are closer than the two sounds from different categories:

$$\hat{e}(A, B) := \frac{1}{n_A(n_A - 1)n_B} \sum_{\substack{a, x \in A \\ x \neq a}} \sum_{b \in B} \left[\mathbb{1}_{d(b,x) < d(a,x)} + \right.$$
$$\left. \frac{1}{2} \mathbb{1}_{d(b,x) = d(a,x)} \right]. \quad (2)$$

n_A and n_B represent the cardinalities of category A and B.

Lexicon. The sWUGGY "Spot-the-word" [26] is used to discriminate an existing word from a lexically similar non-word using the sLM (e.g. "brick" and "blick"). The metric measures the accuracy that the PP of the real word is higher than that of the non-word: $\mathbb{1}_{PP(word) > PP(non-word)}$.

Syntax. sBLIMP acceptability, adapted from BLIMP [27], discriminates a grammatical sentence from an ungrammatical sentence (e.g. "dogs eat meat" and "dogs eats meat"). The metric accepts it if the PP of a grammatical sentence is greater than the ungrammatical one: $\mathbb{1}_{PP(Sentence_{gram}) > PP(Sentence_{ungram})}$.

Semantic. sSIMI similarity measures the similarity between the representations of pairs of words and compares the results with human judgment. The metric is computed as the Spearman's rank correlation coefficient ρ between the semantic similarity scores given by the model and the human scores in the dataset.

3. Proposed System

The two proposed methods are described below. As each of these methods modifies a separate component in the baseline system, they can be used in combination.

3.1. CPC with deep cluster

All four evaluation metrics in this challenge are discriminative tasks. However, as we mentioned, the baseline system does not have sufficiently linguistically discriminative characteristics. To solve this problem, our system combines the CPC model with the deep cluster method [11, 12, 13]. Deep cluster is a clustering method initially designed for image processing. It iterates between doing k-means clustering on the features produced by a neural network and updating its weights by classifying the cluster assignments of each feature. HUBERT [14] is similar to our method in that it uses the deep cluster method to perform self-learning. Fig 2 illustrates the architecture of our method. First, we follow the same procedure as the baseline system until the k-means clustering step. After that, we obtain the discretized pseudo-labels for each feature frame. Then, we randomly initialize[1] a new model with the same architecture as the original one. However, this time the objective is to classify pseudo-labels of feature steps with the cross-entropy (CE) criterion, which is more straightforward than the NCE loss.

---

[1]In a preliminary experiment, we compared the case where the network is initialized with the first round of the CPC network weights and the case where the network is reinitialized randomly. As a result, better performance was obtained with the latter.

Table 1: *Characteristics of the baseline acoustic CPC models. We took the last LSTM layer of CPC-small and the second LSTM hidden layer of CPC-big as inputs to the clustering.*

Model	CPC model configuration		Training data	Input to k-means
	autoregressive	hidden units		
CPC-small	2-layer LSTM	256	LibriSpeech clean-100h	2nd layer of LSTM
CPC-big	4-layer LSTM	512	Libri-light clean-6kh	2nd layer of LSTM

Finally, we execute the second-round k-means clustering with the outputs of the final layer of the autoregressive model. A phoneme discriminative representation is achieved by imposing a phoneme classification task with the pseudo-labels on the autoregressive model.
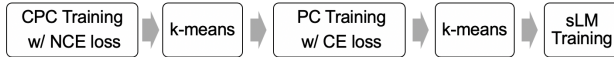

Figure 2: *Illustration of our proposed system (CPC with deep cluster). First, we train a CPC model which consists of g_{enc} and g_{ar} optimized by Eq. (1). Then, k-means clustering is performed to generate discretized units of audio data. Next, another CPC network is trained for phoneme classification (PC) using the discretized units as pseudo-labels. After that, we obtain more linguistically discriminative representation by second-round clustering. Finally, we train a sLM based on pseudo-labels.*

3.2. Conformer CPC

We propose Conformer CPC which replaces the Transformer classifier $h_k(\cdot)$ in Eq. (1) with a Conformer block. It contains two Feed Forward modules sandwiching the Multi-Headed Self-Attention [15] module and the Convolution module. For input c in Eq. (1) to a Conformer block, the output y of the block is:

$$\tilde{c} = c + \frac{1}{2}\text{FFN}(c), \tag{3}$$

$$c' = \tilde{c} + \text{MHSA}(\tilde{c}), \tag{4}$$

$$c'' = c' + \text{Conv}(c'), \tag{5}$$

$$y = \text{Layernorm}(c'' + \frac{1}{2}\text{FFN}(c'')) \tag{6}$$

where FFN refers to the Feed Forward module, MHSA refers to the Multi-Head Self-Attention module, and Conv refers to the Convolution module as described in [16]. This network can capture not only long-term contexts via the self-attention block but also local contexts through a Convolution module. Therefore, it is expected that more precise phonetic and lexical representations are achieved.

4. Experiments

4.1. Experimental Setup

Following the baseline system [9], the encoder g_{enc} consists of five 1d-convolutional layers with kernel sizes of (10, 8, 4, 4, 4) and stride sizes of (5, 4, 2, 2, 2). The downsampling factor of g_{enc} is 160 and the embedding **z** has a sampling rate of 100Hz. Then, the multi-layer long short-term memory (LSTM) [28] network is used as an autoregressive encoder g_{ar}. The CPC model can be divided into two categories: CPC-small and CPC-big, the differences of which are elaborated in Table 1

The transformation $h_k(\cdot)$ in Eq. (1) is a 1-layer Transformer or Conformer network, the parameters of which are as follows: The number of attention heads is 8 and the hidden unit size is 512. The number of hidden units for the feed-forward layers is 2048. As for Conformer, the kernel size of the convolution module is 30. During the training of CPC models, we applied dropout [29] with a rate of 0.1 for the Transformer and Conformer block in the same way as existing studies [15, 16] to achieve a better generalization. We also applied dropout with a rate of 0.5 for the outputs of the CPC prediction network before taking the product with z. K in (1) was set to 12.

The number of iterations for k-means clustering was set to 150. This is the same for the first-round clustering and the second-round one. The language model was based on BERT [22]. We reduced the number of parameters by considering the training time. The model consists of 8 Transformer layers, each of which has 8 attention heads with hidden dimensionality of 512. The dimensionality of feed-forward layers is 2048. The sLM can be trained within 60 hours on a single GPU using the pseudo-text of LibriSpeech 960h.

All models were implemented with PyTorch, including CPC_audio[2] and fairseq[3]. The former was a modified version of the CPC that stabilizes the CPC training by replacing batch normalization [30] with a channel-wise normalization. The latter was only used for the sLM training.

We identified three baseline systems: CPC-small trained on LibriSpeech 100h and 460h, respectively, and CPC-big trained on Libri-light 6kh. Seven proposed systems that combine different methods and training data sizes were included. The proposed methods do not necessarily require the same configuration for the initial autoregressive network and the network for phoneme classification. For this reason, we also compared a system in which the size of the hidden units in the network for phoneme classification was increased from 512 to 1024.

4.2. Results and Discussion
4.2.1. ABX metric

In Table 2, we present the results of the ABX metric for the baseline system and our two proposed systems before clustering. It is clear that all proposed systems of CPC-small outperform the original CPC-small baseline. The combination of CPC with deep cluster and Conformer CPC improves the performance up to 35% relative to the baseline, although not as much as the performance of CPC with deep cluster alone. This shows that the two proposed systems yield linguistically discriminative characteristics for the CPC network. Comparing the CPC-big models, we see that our systems outperform the baseline system only under the condition of "within speaker". One possible reason for this is that the training data for the phoneme classification task in the 2nd stage was LibriSpeech 960h and was not sufficient compared with the baseline CPC-big training

---

[2]https://github.com/facebookresearch/CPC_audio
[3]https://github.com/pytorch/fairseq

Table 2: *Within (all stimuli a, b and x in Eq. (2) are uttered by the same speaker) and Across (a and b are from the same speaker, and x from a different speaker) Speaker ABX metric (lower is better) on Libri-light dev-clean and dev-other. All embeddings are extracted from the final layer of the autoregressive network before clustering. "DC" stands for the deep cluster. "1st" of Training Data means a data set for contrastive learning and "2nd" of that means a data set for phoneme classification. Each model is trained on LibriSpeech (LS) or Libri-light (LL).*

Embedding	Training Data		within ($\downarrow$)		across ($\downarrow$)	
	1st	2nd	dev-clean	dev-other	dev-clean	dev-other
Baseline : CPC-small	LS-100h	/	6.24	8.48	8.17	13.55
Baseline : CPC-small	LS-460h	/	6.19	7.34	8.71	13.02
Proposed: Conformer CPC-small	LS-100h	/	5.78	7.83	8.23	13.59
Proposed: Conformer CPC-small	LS-460h	/	5.40	7.17	7.55	12.19
Proposed: CPC-small+DC	LS-100h	LS-100h	4.78	6.78	7.01	12.34
Proposed: CPC-small+DC	LS-460h	LS-460h	**3.93**	**5.18**	**5.99**	**10.00**
Proposed: Conformer CPC-small+DC	LS-460h	LS-460h	4.05	5.38	6.12	10.60
Baseline : CPC-big	LL-6kh	/	3.41	4.18	**4.85**	**7.64**
Proposed: CPC-big+DC	LL-6kh	LS-960h	3.28	4.14	4.96	8.28
Proposed: CPC-big+DC (1024units)	LL-6kh	LS-960h	**3.11**	**3.98**	4.96	7.92

Table 3: *Overall performance (higher is better) of the baseline and the proposed models on dev sets on three zero-shot metrics. For all models, the k-means clustering (k=50) was performed on LibriSpeech clean-100h, and the BERT-small models were trained on discretized units of LibriSpeech 960h.*

System	Training Data		sWUGGY ($\uparrow$)	sBLIMP ($\uparrow$)	sSIMI ($\uparrow$)	
	1st	2nd			synth.	libri.
Baseline : CPC-small	LS-100h	/	65.79	52.88	-0.09	**9.23**
Baseline : CPC-small	LS-460h	/	66.21	52.79	-0.67	4.92
Proposed: Conformer CPC-small	LS-100h	/	62.22	52.96	**0.90**	7.22
Proposed: Conformer CPC-small	LS-460h	/	66.10	**53.39**	-1.84	5.17
Proposed: CPC-small+DC	LS-100h	LS-100h	65.42	52.86	-1.10	8.14
Proposed: CPC-small+DC	LS-460h	LS-460h	64.89	52.75	-2.11	8.89
Proposed: Conformer CPC-small+DC	LS-460h	LS-460h	**67.21**	53.38	-0.17	7.07
Baseline : CPC-big	LL-6kh	/	65.81	52.91	**3.88**	**5.56**
Proposed: CPC-big+DC	LL-6kh	LS-960h	**66.01**	**54.15**	-0.81	5.45
Proposed: CPC-big+DC (1024units)	LL-6kh	LS-960h	62.64	54.06	-1.65	4.81

with Libri-light 6kh.

4.2.2. sWUGGY metric

Table 3 compares sWUGGY, sBLIMP, and sSIMI metrics with the baseline, and the proposed methods. The two proposed systems, when applied independently, failed to outperform the baseline results compared to the CPC-small baseline systems. Therefore, better performance in the ABX metric does not necessarily guarantee better performance in the sWUGGY metric. However, the best performance is achieved when the two proposed systems are applied simultaneously, i.e., Conformer CPC-small+DC. Compared to the CPC-small trained on LibriSpeech 460h data, Conformer CPC-small+DC achieves a relative improvement of 1.5%. This result suggests that the two methods have a complementary effect on the lexical metric.

4.2.3. sBLIMP metric

The proposed system of CPC-big with deep cluster achieves the highest score among all methods in Table 3. This is also the top result in this challenge[4]. Besides, we can see that all Conformer CPC systems outperform all baseline systems regardless of the amount of training data. It indicates that the Conformer block works to help learn higher-level linguistic features.

4.2.4. sSIMI metric

For all methods including the proposed systems and the baseline systems, there are no systems that are significantly better for both synthetic (synth.) and LibriSpeech (libri.) sets. We can see that the amount of training data does not directly contribute to the performance improvement even if comparing within baseline methods. The proposed systems generally achieve a performance that is almost competitive with the baseline systems.

5. Conclusions

In this paper, we have proposed a system which combines CPC with deep cluster. In deep cluster, we first prepare pseudo-labels obtained by clustering the outputs of a CPC network with k-means. Then, we train an additional autoregressive classifier to predict the previously obtained pseudo labels in a supervised manner. Phoneme discriminative representation is achieved by executing the second-round clustering with the outputs of the final layer of the autoregressive model. In addition, we show that replacing the Transformer layer with a Conformer layer leads to a further gain in a lexical metric. Experimental results show that a relative improvement of 35% in a phonetic metric, 1.5% in the lexical metric, and 2.3% in a syntactic metric are achieved compared to a baseline method of CPC-small which is trained on LibriSpeech 460h data. This result suggests that both methods have a complementary effect on the lexical metric.

---

[4]The leader-board can be viewed at https://zerospeech.com/2021/results.html.

6. References

[1] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *International Conference on Learning Representations*, 2019.

[2] 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, 2020.

[3] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3497–3501.

[4] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," *Proc. Interspeech*, pp. 161–165, 2019.

[5] A. T. Liu, S.-w. Yang, P.-H. Chi, P.-c. Hsu, and H.-y. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6419–6423.

[6] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, "Iterative pseudo-labeling for speech recognition," *Proc. Interspeech*, pp. 1006–1010, 2020.

[7] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," *Proc. Interspeech*, pp. 2817–2821, 2020.

[8] R. Masumura, M. Ihori, A. Takashima, T. Moriya, A. Ando, and Y. Shinohara, "Sequence-level consistency training for semi-supervised end-to-end automatic speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7054–7058.

[9] T. A. Nguyen, M. de Seyssel, P. Rozé, M. Rivière, E. Kharitonov, A. Baevski, E. Dunbar, and E. Dupoux, "The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling," in *NeurIPS Workshop on Self-Supervised Learning for Speech and Audio Processing*, 2020.

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

[11] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 132–149.

[12] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487.

[13] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *International conference on neural information processing*, 2017, pp. 373–382.

[14] 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 *NeurIPS Workshop on Self-Supervised Learning for Speech and Audio Processing*, 2020.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.

[16] 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," *Proc. Interspeech 2020*, pp. 5036–5040, 2020.

[17] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[18] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, "Pushing the limits of semi-supervised learning for automatic speech recognition," in *NeurIPS Workshop on Self-Supervised Learning for Speech and Audio Processing*, 2020.

[19] M. Huang, J. Zhang, M. Cai, Y. Zhang, J. Yao, Y. You, Y. He, and Z. Ma, "Improving rnn transducer with normalized jointer network," *arXiv preprint arXiv:2011.01576*, 2020.

[20] 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," *arXiv preprint arXiv:2010.13956*, 2020.

[21] 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*, 2010, pp. 297–304.

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[23] 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)*, 2015, pp. 5206–5210.

[24] J. Kahn, M. Rivière, 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7669–7673.

[25] 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," *Proc. Interspeech*, pp. 1006–1010, 2013.

[26] G. Le Godais, T. Linzen, and E. Dupoux, "Comparing character-level neural language models using a lexical decision task," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, pp. 125–130.

[27] A. Warstadt, A. Parrish, H. Liu, A. Mohananey, W. Peng, S.-F. Wang, and S. Bowman, "Blimp: The benchmark of linguistic minimal pairs for english," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 377–392, 2020.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] 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*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.