



Pre-Training Transformer Decoder for End-to-End ASR Model with Unpaired Speech Data

Junyi Ao^{1,†}, Ziqiang Zhang^{2,†}, Long Zhou³, Shujie Liu³, Haizhou Li¹,
Tom Ko, Lirong Dai², Jinyu Li³, Yao Qian³, Furu Wei³

¹School of Data Science, The Chinese University of Hong Kong (Shenzhen)

²NEL-SLIP, University of Science and Technology of China

³Microsoft

Abstract

This paper studies a novel pre-training technique with unpaired speech data, Speech2C, for encoder-decoder based automatic speech recognition (ASR). Within a multi-task learning framework, we introduce two pre-training tasks for the encoder-decoder network using acoustic units, i.e., pseudo codes, derived from an offline clustering model. One is to predict the pseudo codes via masked language modeling in encoder output, like HuBERT model, while the other lets the decoder learn to reconstruct pseudo codes autoregressively instead of generating textual scripts. In this way, the decoder learns to reconstruct original speech information with codes before learning to generate correct text. Comprehensive experiments on the LibriSpeech corpus show that the proposed Speech2C can relatively reduce the word error rate (WER) by 19.2% over the method without decoder pre-training, and also outperforms significantly the state-of-the-art wav2vec 2.0 and HuBERT on fine-tuning subsets of 10h and 100h. We release our code and model at <https://github.com/microsoft/SpeechT5/tree/main/Speech2C>.

Index Terms: self-supervised speech pre-training, automatic speech recognition, encoder-decoder pre-training.

1. Introduction

Self-supervised learning has been shown effective in natural language processing (NLP), e.g., BERT [1] and BART [2], where it makes use of a large amount of unlabeled data for pre-training to improve the performance of downstream tasks, such as automatic speech recognition (ASR) [3, 4, 5, 6, 7, 8].

By their training objectives, self-supervised methods can be categorized into contrastive learning [9, 10, 11] and reconstructive learning [6, 12, 13]. In contrastive learning, CPC [9] uses a probabilistic contrastive loss, which induces the latent space to capture information and uses an autoregressive model to classify future frames from negative examples. Wav2vec [10] pre-trains a simple multi-layer convolutional neural network optimized via a noise contrastive binary classification task. In reconstructive learning, APC [6] reconstructs the future frame with a unidirectional encoder via learning meaningful, non-specific, and transferable speech representations. HuBERT [12] firstly generates pseudo labels from an offline clustering step, and they are used as the target to calculate a BERT-like loss for model training, which can be boosted by an iterative process.

However, many previous studies only pre-train the speech encoder for various spoken downstream tasks, while the de-

coder is not pre-trained for the end-to-end encoder-decoder based ASR models, which usually rely heavily on a large amount of transcribed audio data. Although there are some attempts to pre-train a Transformer decoder for end-to-end ASR model, they have to use additional unpaired text data [4, 14, 15]. *Can we pre-train the ASR decoder with speech-only data?* There are two main challenges, (1) speech signals are continuous and textual representations are discrete, and (2) the decoder is responsible for generating text which is very different from conventional speech representation (e.g., waveform, log-mel fbank feature, or hidden states).

To address this problem, we propose a speech to code pre-trained model (**Speech2C**), trying to pre-train the encoder-decoder model with speech-only data. Under the multi-task learning framework, we employ two pre-training tasks for encoder-decoder pre-training using speech-only data with the acoustic units learned from an offline clustering model, aka pseudo codes. The first task predicts the pseudo codes via masked language modeling (MLM) in encoder output, like HuBERT model. For the second one, the decoder of Speech2C learns to reconstruct reduced pseudo codes autoregressively, instead of generating real text transcription. Both codes and text are discrete representations and contain semantic information of the speech signal. According to our observation, a certain amount of codes are highly correlated with the transcriptions, which implies that the decoder can learn text prediction in our pre-training framework.

We conduct massive experiments on the Librispeech dataset to validate the proposed Speech2C. To the best of our knowledge, this is the first work to pre-train an encoder-decoder model for ASR with only speech-only data, and our proposed Speech2C achieves a new state-of-the-art performance on the test set of Librispeech.

2. Related Work

We consider our work most related to HuBERT [12], which benefits from an offline clustering step to provide pseudo labels for a BERT-like pre-training. The backbone of HuBERT includes a convolutional feature encoder and a Transformer context encoder. During pre-training, HuBERT first uses k-means to learn the initial quantizer that maps speech signals to discrete labels, and performs BERT-style pre-training where the inputs are masked speech signals and prediction targets are discrete labels. Moreover, HuBERT allows refinement on the pseudo label by further using the pre-trained model as the new quantizer to train a new iteration of the model. However, HuBERT model only pre-trains a speech encoder, leaving the decoder not pre-

[†]Equal contribution. Work done during internship at Microsoft Research Asia.

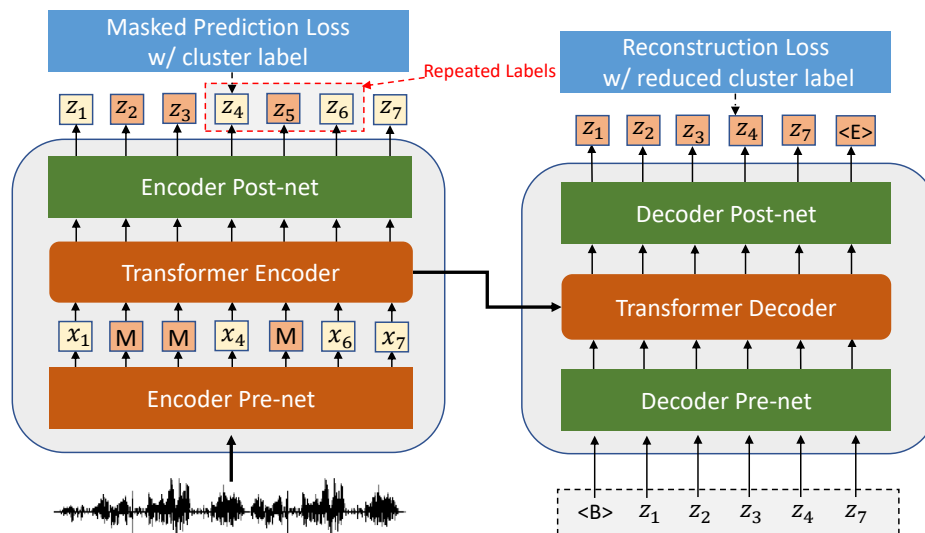


Figure 1: The framework of our proposed Speech2C model, which is pre-trained with masked prediction loss and reconstruction loss on unpaired speech data for end-to-end ASR. In this example, z_4 , z_5 and z_6 are from the same class and can be reduced to a single label for the decoder. In the fine-tuning stage, we initialize the ASR model with pre-trained Speech2C by removing the encoder post-net and the decoder pre-net/post-net, since they are trained to process pseudo codes.

trained for the encoder-decoder based tasks, such as end-to-end ASR [16]. Based on HuBERT encoder, our proposed Speech2C model can also pre-train a Transformer decoder with pseudo label from the clustering model. Besides, some research works [17, 18] attempt to leverage pseudo codes to reconstruct audios, but they are mainly designed for speech resynthesis and speech to speech translation tasks.

In addition, an idea was explored to pre-train a decoder for end-to-end ASR [4, 14, 15]. The authors in [4] employ a single speaker text to speech (TTS) system to generate synthesized speech from a large number of transcripts, and use the generated speech-text pairs to pre-train the decoder. In [14], unpaired text data are used to pre-train the transformer decoder, which is pre-trained as a conditional language model by constructing empty or artificial states to replace the real encoder hidden states. Leveraging large-scale unpaired speech and text data, SpeechT5 [15] pre-trains a shared encoder-decoder model for various spoken language tasks. However, all previous work still utilize text data to pre-train a decoder for end-to-end ASR. In contrast, Speech2C is the first work to pre-train a Transformer decoder for ASR without text data.

3. Methods

In this section, we first illustrate our model architecture, based on which, we present two pre-training tasks in our proposed Speech2C, the masked prediction loss for the encoder and the pseudo-code reconstruction loss for the decoder.

3.1. Model Architecture

The model architecture of the proposed Speech2C for pre-training is composed of an encoder network that extracts latent speech representations from raw acoustic inputs and learns contextualized speech representations, and a decoder network for autoregressively reconstructing the pseudo codes of corresponding source speech. Both encoder network and decoder network are boosted with relative positional encoding [19]. Figure 1 illustrates the framework of the proposed Speech2C.

The encoder network follows the HuBERT BASE architec-

ture, with an encoder pre-net, the Transformer encoder, an encoder post-net, as shown on the left panel of Figure 1. More specifically, the encoder pre-net is a convolutional network for pre-processing waveform, which is of seven 512-channel layers with strides [5,2,2,2,2,2,2] and kernel widths [10,3,3,3,3,2,2]. The Transformer encoder contains 12 layers with model dimension 768, inner dimension 3072 and 12 attention heads. Moreover, the encoder-post network contains a projection layer and a code embedding layer, which are utilized to convert hidden states into pseudo codes.

The decoder network also contains a decoder pre-net, the Transformer decoder, and a decoder post-net. The pre-net transforms a code index into an embedding vector. The Transformer decoder has a similar architecture to the Transformer encoder except for the cross-attention and the masked self attention. Finally, the post-net transforms the hidden state into the probability distribution of codes, normalized by the softmax function.

3.2. Self-Supervised Speech Pre-Training

The proposed pre-training learning method for Speech2C has access to speech-only data. We introduce two pre-training tasks to pre-train the encoder-decoder model, including masked prediction loss for the encoder and reconstruction loss for the decoder.

3.2.1. Masked prediction loss

During pre-training, the encoder pre-net first generates a feature sequence \mathbf{x} from waveform by down-sampling, then the Transformer encoder consumes masked acoustic features $\tilde{\mathbf{x}}$ and output hidden states \mathbf{h}^L . Furthermore, the network including encoder post-net is optimized to predict the discrete target sequence \mathbf{z} , where each $z_t \in [C]$ is a C -class categorical variable. The distribution over codewords is parameterized with:

$$p_f(c|\tilde{\mathbf{x}}, t) = \frac{\exp(\text{sim}(\mathbf{h}_t^L \mathbf{W}, \mathbf{e}_c)/\tau)}{\sum_{c'=1}^C \exp(\text{sim}(\mathbf{h}_t^L \mathbf{W}, \mathbf{e}_{c'})/\tau)} \quad (1)$$

where \mathbf{W} is a projection matrix, \mathbf{h}_t^L is the output hidden state for step t and layer L , \mathbf{e}_c is the embedding for codeword c ,

$\text{sim}(a, b)$ computes the cosine similarity between two vectors and $\tau = 0.1$ is used to scale the logit.

Specifically, $\tilde{\mathbf{x}}$ comes from \mathbf{x} by span mask strategies, where 8% of timesteps are randomly selected as start indices, and spans of 10 steps are masked. Based on the above distribution, we denote the cross-entropy loss computed over masked timesteps as

$$\mathcal{L}_{mlm} = \sum_{t \in \mathcal{M}} \log p_f(\mathbf{z}_t | \tilde{\mathbf{x}}, t), \quad (2)$$

where, \mathcal{M} denotes the set of masked timesteps, and \mathbf{z}_t denotes the frame-level target at timestep t from \mathbf{Z} .

3.2.2. Pseudo-code reconstruction loss

In addition to masked prediction loss in the encoder, we also design a reconstruction loss to pre-train the Transformer decoder. Following the denoising autoencoder in BART [2], the decoder network is optimized to generate the reduced pseudo codes with the maximum likelihood estimation as

$$\mathcal{L}_{mle} = \sum_{n=1}^N \log p(\mathbf{z}_n | \mathbf{z}_{<n}, \tilde{\mathbf{x}}), \quad (3)$$

where N is the length of pseudo codes.

Pseudo codes are similar to real texts because (1) they are discrete representations and have fixed vocabulary; (2) they all have rich semantic information that can be aligned to speech fragments. Hence, we believe this pre-training method on pseudo codes can help the decoder learn how to generate text sequences. The pseudo codes of adjacent speech frames have some repeated codes, which may represent similar semantic information, but repeated words are rarely used consecutively in textual languages. To reduce the gap between pseudo code and real text, we remove the repeating code of adjacent speech frame, which will be studied in the ablation study.

4. Experiments

4.1. Training Details

All models are implemented in Fairseq¹ [20]. For speech pre-training, we use the full 960 hours of LibriSpeech [21] audio without transcription. We optimize the model with Adam [22] by warming up the learning rate for the first 8% of updates to a peak of 2×10^{-4} , which is linearly decayed for the following updates. We pre-train the proposed Speech2C model on 32 V100 GPUs with a batch size of around 87.5s samples per GPU for 400k steps.

For the fine-tuning, we employ 10 hours and 100 hours as the supervised paired corpus, and use the character set as the model units for the text. Because pseudo codes and textual characters have different vocabulary, we initialize the ASR model with pre-trained Speech2C without the encoder post-net and decoder pre-net/post-net, which are trained from scratch in our fine-tuning model. We utilize the CTC and cross-entropy loss to fine-tune the model [23], where the loss weights are 0.5 for both of them. The models are trained on 16 v100 GPUs with a batch size of 100s samples per GPU. The learning rate is warmed up for the first 10% steps, held as a constant for the following 40% steps, and is decayed linearly for the rest steps. For the 10/100 hours subset, we train the model with a learning

¹<https://github.com/pytorch/fairseq>

Table 1: WER on the LibriSpeech test sets when training on the 10 hours and 100 hours subset. † indicates that the results are not reported in [12] and obtained by fine-tuning the public released model.

Model	LM	test-clean	test-other
10 hours subset			
wav2vec2.0 BASE [11]	None	11.1	17.6
HuBERT BASE † [12]	None	10.1	16.8
Our Speech2C	None	7.8	13.1
wav2vec2.0 BASE [11]	4-gram	4.3	9.5
wav2vec2.0 BASE [11]	Transf	3.2	7.8
HuBERT BASE [12]	4-gram	4.3	9.4
Our Speech2C	Transf	3.1	7.0
100 hours subset			
wav2vec2.0 BASE [11]	None	6.1	13.3
wav2vec2.0 LARGE [11]	None	4.7	9.0
HuBERT BASE † [12]	None	6.3	13.2
SpeechT5 [15]	None	4.4	10.4
Baseline	None	5.0	11.9
Our Speech2C	None	4.3	9.0
wav2vec2.0 BASE [11]	4-gram	3.4	8.0
wav2vec2.0 BASE [11]	Transf	2.6	6.3
HuBERT BASE [12]	4-gram	3.4	8.1
SpeechT5 [15]	Transf	2.4	5.8
Baseline	Transf	2.5	6.3
Our Speech2C	Transf	2.4	5.2

rate of $2e-5/4e-5$ for 25k/80k, and fix the encoder part for the first 10k/25k steps.

For ASR inference, we apply the joint CTC/attention decoding [24] and train a transformer language model (LM) by LibriSpeech-LM Corpus with the same architecture as in [15] for the shallow fusion [25] to boost the performance of our baseline and Speech2C. We sweep over the weights of the language model and CTC from 0 to 1 on the dev-other subset and choose the best weights according to the WER.

4.2. Main Results

The results of ASR on the 10 hours and 100 hours set of LibriSpeech are reported in Table 1. The WER is evaluated on the standard Librispeech test-clean/other sets. We compare with several state-of-the-art self-supervised approaches, including encoder-based wav2vec 2.0 [11] and HuBERT [12], and encoder-decoder based SpeechT5 [15], which utilizes the unpaired speech and text corpus to pre-train. We build a strong encoder-decoder based ASR baseline system, which has the same model structure as our Speech2C, while the weights of the encoder are initialized by the HuBERT BASE model.

Without LM fusion, the baseline outperforms wav2vec 2.0 BASE and HuBERT BASE with the help of the joint CTC/attention decoding, which shows the importance of the decoder. Our proposed Speech2C model achieves significant improvements on all settings compared to wav2vec 2.0 BASE, HuBERT BASE, SpeechT5 [15] and our strong baselines, demonstrating the superiority of the proposed pre-training method. More specifically, our Speech2C without LM gets a relative 19.2% WER reduction on the average of all sets compared to the baseline system for 100h subset, which achieves state-of-the-art performance. Furthermore, when decoding with LM shal-

low fusion, our Speech2C still obtains the lower WERs than the strong baseline on all sets.

4.3. Ablation Study

We present a series of ablation studies in the following sections to learn how code reduction, continuing pre-training, and model layer numbers affect performance. The models for ablation studies are pre-trained on 960 hours and fine-tuned on the 100-hour subset using fixed hyperparameters.

4.3.1. Effect of code reduction

Here, we start with probing the effectiveness of the k-means clustering algorithm concerning code reduction. In Table 2, we summarize the results of Speech2C with repeated pseudo codes when calculating reconstruction loss, and the Speech2C with repeated codes performs slightly worse than Speech2C with reduced codes. Moreover, reducing repeated codes has the following advantages: (1) the average length of reduced codes is significantly shorter than that of repeated codes, which will accelerate the training progress of Speech2C; (2) removing repeated codes does not loss semantic information, and brings pseudo codes closer to the text.

Table 2: Comparison of Speech2C with repeated codes or reduced codes in terms of average length and WER.

Model	Length	test-clean	test-other
Speech2C (repeated)	358	4.4	9.4
Speech2C (reduced)	216	4.3	9.0

4.3.2. Continued pre-training from released model

In addition to pre-train Speech2C from scratch, our method can also support continually pre-train from a pre-trained speech encoder model, such as pre-trained HuBERT. Table 3 shows the experimental results of two pre-trained Speech2C. Although two pre-training methods achieve comparable performance, initializing the encoder of Speech2C with HuBERT can speed up the convergence and reduce the training time.

Table 3: WER scores of Speech2C trained from scratch or initialized by HuBERT encoder.

Model	test-clean	test-other
Speech2C (from scratch)	4.3	9.0
Speech2C (from HuBERT)	4.1	9.1

4.3.3. Effect of layer numbers

Compared to wav2vec and HuBERT which adopt CTC inference based on speech encoder, the encoder-decoder based ASR (eg., SpeechT5 and Speech2C) adds an external decoder to generate text. To reduce the influence of model parameters, we reduce the model layers of encoder-decoder model to the similar parameters of encoder model. As shown in table 4, we list the experimental results of Speech2C with different encoder and decoder layers. We change the default setting of 12 encoder layers and 6 decoder layers to the total same layer number of wav2vec2.0 BASE and HuBERT BASE, such as 10 encoder layers and 2 decoder layers. Results show that although reducing the model layers degrade the performance of Speech2C, it still behaves better than encoder based ASR model.

Table 4: WER scores of Speech2C with different layer numbers.

Model	Size	Layer (enc-dec)	test-clean	test-other
wav2vec2.0	95M	12-0	6.1	13.3
HuBERT	95M	12-0	6.3	13.2
SpeechT5	154M	12-6	4.4	10.4
Speech2C	104M	8-4	4.9	10.9
Speech2C	100M	10-2	4.6	9.8
Speech2C	152M	12-6	4.3	9.0

4.4. Analysis

In this section, we want to answer a question: why pre-train the decoder with pseudo code can help text generation in ASR? Frames of the same phonemes are more likely labeled as similar sequences of pseudo codes. We give an example chosen from the LibriSpeech dataset to show the high relevance of pseudo codes to text data and the patterns inherent in them. As shown in Figure 2, the codes corresponding to “wonder” in different samples are also similar and have an obvious pattern, which demonstrates that pseudo codes can be regarded as an intermediary language between waveform and transcription.

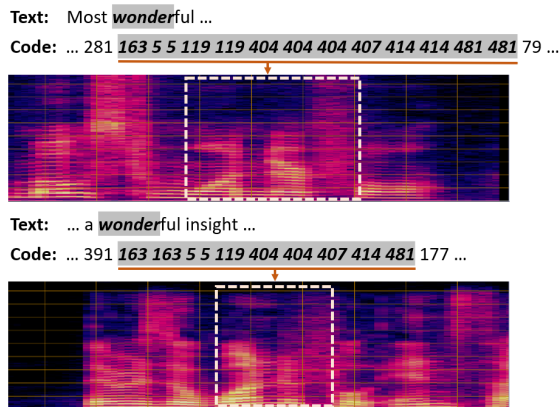


Figure 2: The transcripts, pseudo codes and Mel-Spectrum from two sentences, where the white boxes and the bold pseudo codes corresponding to the subword “wonder”.

5. Conclusion

This paper proposes a novel model, Speech2C, to pre-train an encoder-decoder model with speech-only data. We present two pre-training tasks including masked prediction task and reconstruction task by taking advantage of acoustic units, i.e. pseudo codes derived from an offline clustering model. Massive experiments and analyses on the LibriSpeech dataset show the effectiveness and superiority of our proposed Speech2C. To the best of our knowledge, Speech2C is the first work to pre-train an encoder-decoder based ASR model with speech-only data. For future work, we will pre-train a multilingual Speech2C to address cross-lingual tasks, such as speech translation.

6. Acknowledgements

This research is supported by the internal project of Shenzhen Research Institute of Big Data under the Grant No. T00120220002, the Guangdong Provincial Key Laboratory of Big Data Computing under the Grant No. B10120210117-KP02, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen) and the University Development Fund, CUHK-Shenzhen, under the Grant No. UDF01002333 and UF02002333.

7. References

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.
- [2] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880.
- [3] A. Baevski, M. Auli, and A. Mohamed, "Effectiveness of self-supervised pre-training for speech recognition," *arXiv preprint arXiv:1911.03912*, 2019.
- [4] Z. Fan, S. Zhou, and B. Xu, "Unsupervised pre-training for sequence to sequence speech recognition," *arXiv preprint arXiv:1910.12418*, 2019.
- [5] W. Wang, Q. Tang, and K. Livescu, "Unsupervised pre-training of bidirectional speech encoders via masked reconstruction," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6889–6893.
- [6] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3497–3501.
- [7] C. Wang, Y. Wu, Y. Qian, K. Kumatani, S. Liu, F. Wei, M. Zeng, and X. Huang, "Unispeech: Unified speech representation learning with labeled and unlabeled data," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10937–10947.
- [8] D. Jiang, W. Li, R. Zhang, M. Cao, N. Luo, Y. Han, W. Zou, K. Han, and X. Li, "A further study of unsupervised pretraining for transformer based speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6538–6542.
- [9] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.
- [10] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.
- [11] 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. 12449–12460, 2020.
- [12] 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.
- [13] 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," *arXiv preprint arXiv:2110.13900*, 2021.
- [14] 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 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6543–6547.
- [15] 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 preprint arXiv:2110.07205*, 2021.
- [16] J. Li, "Recent advances in end-to-end automatic speech recognition," *arXiv preprint arXiv:2111.01690*, 2021.
- [17] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed, and E. Dupoux, "On generative spoken language modeling from raw audio," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1336–1354, 2021. [Online]. Available: <https://aclanthology.org/2021.tacl-1.79>
- [18] A. Lee, P.-J. Chen, C. Wang, J. Gu, X. Ma, A. Polyak, Y. Adi, Q. He, Y. Tang, J. Pino *et al.*, "Direct speech-to-speech translation with discrete units," *arXiv preprint arXiv:2107.05604*, 2021.
- [19] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Jun. 2018, pp. 464–468.
- [20] 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," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 2019, pp. 48–53.
- [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] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: <https://arxiv.org/pdf/1412.6980.pdf>
- [23] 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.
- [24] T. Hori, S. Watanabe, and J. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jul. 2017, pp. 518–529.
- [25] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.