Efficiently Fusing Pretrained Acoustic and Linguistic Encoders for Low-Resource Speech Recognition

Cheng Yi, Shiyu Zhou, and Bo Xu, *Member, IEEE*

***Abstract*—End-to-end models have achieved impressive results on the task of automatic speech recognition (ASR). For low-resource ASR tasks, however, labeled data can hardly satisfy the demand of end-to-end models. Self-supervised acoustic pre-training has already shown its impressive ASR performance, while the transcription is still inadequate for language modeling in end-to-end models. In this work, we fuse a pre-trained acoustic encoder (wav2vec2.0) and a pre-trained linguistic encoder (BERT) into an end-to-end ASR model. The fused model only needs to learn the transfer from speech to language during fine-tuning on limited labeled data. The length of the two modalities is matched by a monotonic attention mechanism without additional parameters. Besides, a fully connected layer is introduced for the hidden mapping between modalities. We further propose a scheduled fine-tuning strategy to preserve and utilize the text context modeling ability of the pre-trained linguistic encoder. Experiments show our effective utilizing of pre-trained modules. Our model achieves better recognition performance on CALLHOME corpus (15 hours) than other end-to-end models.**

***Index Terms*—BERT, end-to-end modeling, low-resource ASR, pre-training, wav2vec.**

I. Introduction

PIPELINE methods decompose the task of automatic speech recognition (ASR) into three components to model: acoustics, pronunciation, and language [1]. It can dramatically decrease the difficulty of ASR tasks, requiring much less labeled data to converge. With self-supervised pre-trained acoustic model, the pipeline method can achieve impressive recognition accuracy with as few as 10 hours of transcribed speech [2]–[5]. However, it is criticized that the three components are combined by two fixed weights (pronunciation and language), which is inflexible [6].

On the contrary, end-to-end models integrate the three components into one and directly transform the input speech features to the output text. Among end-to-end models, the sequence-to-sequence (S2S) model is composed of an encoder and a decoder, which is the dominant structure [7]–[10]. The end-to-end modeling achieves better results than pipeline methods on most of public datasets [9], [11]. Nevertheless, it requires at least hundreds of hours of transcribed speech for training due to the enormous parameter space.

Pre-training can help the end-to-end model work well in the target ASR task on the low-resource condition [12]. Supervised pre-training, also known as supervised transfer learning [13], uses the knowledge learned from other tasks and applies it to the target one [8]. However, this solution requires sufficient and domain-similar labeled data, which is hard to satisfy. Another solution is to partly pretrain the end-to-end model with unlabeled data. For example, [14] pre-trains the acoustic encoder of Transformer by masked predictive coding (MPC). Unlike the encoder, the decoder of the S2S model cannot be separately pre-trained since it is conditioned on an acoustic representation. In other words, it is difficult to guarantee the consistence between pre-training and fine-tuning for the decoder.

Instead of realizing linguistic pre-training for the S2S model, we fuse a pre-trained acoustic encoder (wav2vec2.0) and a pre-trained linguistic encoder (BERT) into a single end-to-end ASR model. The fused model has been exposed separately to adequate speech and text data, so that it only needs to learn the transfer from speech to language during fine-tuning with limited labeled data. To bridge the length gap between speech and language modalities, a monotonic attention mechanism without additional parameters is applied. Besides, a fully connected layer is introduced for the mapping between hidden states of the two modalities. Our model works in the way of non-autoregressive (NAR) way [15] due to the absent of a well-defined decoder structure. Different from self-training, an acoustic representation is fed to the linguistic encoder during fine-tuning. The inconsistency can severely influence the representation ability of the linguistic encoder. We help this module get along with the acoustic encoder by a scheduled fine-tuning strategy.

II. Related Work

A lot of work propose methods to leverage text data for the end-to-end model. Deep fusion [16] and cold fusion [17], [18] integrate a pre-trained auto-regressive language model (LM) into a S2S model. In these settings, the S2S model is randomly initialized and still needs a lot of labeled data for training.

Tran *et al.* [10] builds a S2S model with a pre-trained acoustic encoder and a multilingual linguistic decoder. The decoder is part of a S2S model (mBART [19]) pre-trained on text data. Although this model achieves great results in the task of speech-to-text translation, it is not verified in ASR tasks. Besides,

Manuscript received January 22, 2021; revised February 24, 2021; accepted March 30, 2021. Date of publication April 7, 2021; date of current version April 29, 2021. This work was supported by the National Key Research and Development Program of China under Grant 2017YFB1002102. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nancy F. Chen. *(Corresponding author: Cheng Yi.)*

Cheng Yi is with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: yicheng2016@ia.ac.cn).

Shiyu Zhou and Bo Xu are with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: shiyuzhou@ia.ac.cn; xubo@ia.ac.cn).

Digital Object Identifier 10.1109/LSP.2021.3071668

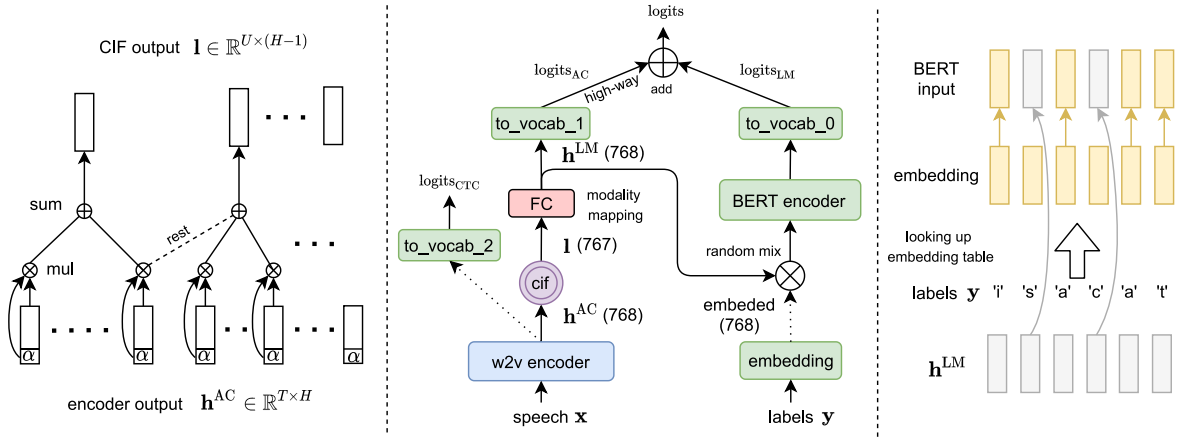

Fig. 1. The structure of w2v-cif-bert. On the left part, a variant CIF mechanism converts $\boldsymbol{h}^{\mathrm{AC}}$ to $\boldsymbol{l}$ without any additional module; on the right part, $\boldsymbol{h}^{\mathrm{LM}}$ is not directly fed into BERT but mixed with embedded labels in advance during training. Modules connected by dot lines are ignored during inference. Numbers in "()" indicate the size of hidden in the model.

this work does not deal with the inconsistency we mentioned above.

Some works also focus on knowledge distillation [20] or semi-supervised learning [21]–[23]. These methods usually require a seed end-to-end ASR model, and they are not as convenient as the pretrain-and-finetune paradigm.

Our scheduled fine-tuning strategy is similar to the training of NAR models [24], [25], where decoders are fed with the hybrid of masked and unmasked tokens. However, these NAR decoders cannot be separately pre-trained since they rely on the acoustic representation.

III. Methodology

We propose an innovative end-to-end model called **w2v-cif-bert**, which consists of wav2vec2.0 (pre-trained on a speech corpus), BERT (pre-trained on a text corpus), and CIF (Continuous Integrate-and-Fire) mechanism to connect the above two modules. The detailed realization of our model is demonstrated in Fig. 1. It is worth noting that only the fully connection in the middle (colored red) does not participate in any pre-training.

A. Acoustic Encoder

We choose wav2vec2.0 as the acoustic encoder since it has been well verified [4], [10], [26]. Wav2vec2.0 is a pre-trained encoder that converts raw speech signals into the acoustic representation [4]. During pre-training, it masks the speech input in the latent space and solves a contrastive task. Wav2vec2.0 can outperform previous semi-supervised methods simply through fine-tuning on transcribed speech with the CTC criterion [27].

Wav2vec2.0 is composed of a feature encoder and a context network. The feature encoder produces outputs with a stride of about 20 ms between each frame, and a receptive field of 25 ms of audio. The context network is a stack of self-attention blocks for context modeling [28]. In this work, wav2vec2.0 is used to convert speech $\boldsymbol{x}$ to the acoustic representation $\boldsymbol{h}^{\mathrm{AC}}$ (w2v encoder) in our model, which is colored blue in Fig. 1.

B. Modality Adaptation

Most commonly, a global attention is applied to connect the acoustic and language representation [28]. However, this mechanism is to blame for the poor generalization of text length [11], which is worse under the sample scarcity. Instead, we use the CIF mechanism [7] to bridge the discrepant sequence lengths. CIF constrains a monotonic alignment between the acoustic and linguistic representation, and the reasonable assumption drastically decreases the difficulty of learning the alignment.

In the original work [7], CIF uses a local convolution module to assign the attention value to each input frame. To avoid introducing additional parameters, the last dimension of the $\boldsymbol{h}_t^{\mathrm{AC}}$ is regarded as the raw scalar attention value (before sigmoid operation), as demonstrated in the left part of Fig. 1. It separates the vector $\boldsymbol{h}_t^{\mathrm{AC}}$ at time t into a scalar $\boldsymbol{h}_{t,d}^{\mathrm{AC}}$ and a remnant vector $\boldsymbol{h}_{t,1:d-1}^{\mathrm{AC}}$, where d is size of the vector. The normalized attention value α_t are accumulated along the time dimension T and a linguistic representation $\boldsymbol{l}_u$ outputs whenever the accumulated α_t surpasses 1.0 [7]. During training, the sum of attention values for one input sample is resized to the number of output tokens $\boldsymbol{y}$ ($n^* = |\boldsymbol{y}|$) [7]. The formalized operations in CIF are:

$$\alpha_t = \mathrm{sigmoid}(\boldsymbol{h}_{t,d}^{\mathrm{AC}}) \tag{1}$$

$$\hat{n} = \sum_{t}^{T} \alpha_t \tag{2}$$

$$\alpha_t' = \frac{n^*}{\hat{n}} \alpha_t \tag{3}$$

$$\boldsymbol{l}_u = \sum_{t}^{\mathrm{S}_u} \alpha_t' \boldsymbol{h}_{t,1:d-1}^{\mathrm{AC}}, \tag{4}$$

where $\boldsymbol{h}^{\mathrm{AC}}$ represents acoustic vectors with length of T, $\boldsymbol{l}$ represents accumulated acoustic vectors with length of U, $\hat{n}$ represents the predicted decoding length, and S_u represents the u-th segment of $\boldsymbol{h}_t$ where the sum of α_t exceeds 1.0.

CIF introduces a quantity loss to supervise the encoder generating the correct number of final modeling units:

$$L_{\text{qua}} = ||n^* - \hat{n}||_2, \tag{5}$$

where $||\cdot||_2$ is the L2 norm. During inference, an extra rounding operation is applied on $\hat{n}$ to simulate n^*.

Based on the matched sequence length, the accumulated acoustic vector $\boldsymbol{l}$ is mapped into the linguistic vector $\boldsymbol{h}^{\text{LM}}$ by a randomly initialized fully connected layer (FC), realizing the modality adaptation.

C. Linguistic Encoder

We choose BERT as the linguistic encoder in our model. BERT is a masked LM, applying the *mask-predict* criterion for self-training and utilizes both left and right context on a huge amount of text data [29]. BERT has empirically shown impressive performance on various NLP tasks [29]–[31].

BERT is composed of three modules: an embedding table (embedding) to convert tokens to hidden vectors, a final fully connection layer to convert hidden vectors to an output softmax over the vocabulary, and a Transformer encoder (BERT encoder) for bidirectional context modeling. These modules are colored green in Fig. 1. Pre-trained BERT can compensate for the lack of text data on low-resource ASR tasks.

D. Additional Connections

We add two additional connections after stacking the three modules. Firstly, a high-way directly connects the acoustic encoder to the final output. Secondly, an auxiliary CTC supervision is attached to the acoustic encoder (L_{ctc}). Both of the two connections can make the encoder affected by the target supervision more effectively [7], [32]. We use BERT's final fully connected layer ("to_vocab_0" in Fig. 1) to initialize the new ones ("to_vocab_1" and "to_vocab_2" in Fig. 1). The final output of our model is:

$$\text{logits} = \lambda_{\text{AC}}\text{logits}_{\text{AC}} + \lambda_{\text{LM}}\text{logits}_{\text{LM}}, \tag{6}$$

where λ_{AC} and λ_{LM} are the weights for the output from the acoustic encoder $\text{logits}_{\text{AC}}$ and the linguistic encoder $\text{logits}_{\text{LM}}$. The final output is supervised by the cross-entropy criterion (L_{ce}) [28].

The final loss during fine-tuning over labeled speech is:

$$L = L_{\text{ce}} + \mu_1 L_{\text{qua}} + \mu_2 L_{\text{ctc}}, \tag{7}$$

where μ_1 and μ_2 are the weights for these losses respectively.

E. Scheduled Modality Fusion

We notice a distinct mismatch between BERT as a text feature extractor during pre-training and as a linguistic encoder in the ASR model during fine-tuning. BERT cannot process $\boldsymbol{h}^{\text{LM}}$ according to its pre-trained knowledge of text processing. It needs to greatly adjust the parameters on the bottom, which is significantly different from fine-tuning on NLP tasks. Worst of all, BERT's parameters cannot transfer from the bottom up with a minimum cost. On the contrary, BERT will undergo a massive top-down change following the broadcast of gradients.

In response to the above mismatch, we randomly replace $\boldsymbol{h}^{\text{LM}}$ with embedded target tokens $\boldsymbol{y}$ along linguistic length U with a scheduled *gold rate* $p \in [0, 1]$, as demonstrated in the right part of Fig. 1. p will decrease during fine-tuning for BERT to get rid of the dependency on $\boldsymbol{y}$. At the beginning of fine-tuning, BERT cannot understand the frames from $\boldsymbol{h}^{\text{LM}}$ and views them as *masked* input. BERT mainly utilizes the gold context (label embedding vectors) to predict. Due to the consistency with pre-training, BERT can fastly converge to high predicting accuracy. Along with the fine-tuning and the decrease of p, BERT gradually grasps the meaning of $\boldsymbol{h}^{\text{LM}}$ and predicts more accurately. During inference, p is set to 0 and BERT needs to predict with pure $\boldsymbol{h}^{\text{LM}}$.

IV. Experiments

A. Datasets and Experimental Settings

We focus on low-resource ASR and mainly experiment on CALLHOME corpus [1]. CALLHOME is a multilingual corpus with less than 20 hours of transcribed speech for each language. In this work, we use CALLHOME Mandarin (MA, LDC96S15) and English (EN, LDC97S20). MA has 23 915 transcribed utterances (15.6 h) for training and 3021 for testing. EN has 21 194 transcribed utterances (14.9 h) for training and 2840 for testing. To compare with more work, we also test our model on a relatively large and popular corpus: HKUST (178 h) [33]. Both of them are telephone conversational speech corpora, which is much more realistic and harder than Librispeech [4].

We use the open source wav2vec2.0[1] as the encoder, bert-base-uncase[2] as the linguistic encoder for English and bert-base-chinese[3] as the linguistic encoder for Chinese. We are free from pre-processing the transcripts since the built-in tokenizers of BERTs can automatically generate the modeling units. All the code and experiments are implemented using *fairseq* [34]. We keep most of the training settings in wav2vec2.0 fine-tuning demonstration.[4] We optimize with Adam, warming up the learning rate for 8000 steps to a peak of 4×10^{-5}, holding 42 000 steps and then exponential decay it. we only use a single GPU (TITAN Xp) for each experiment. Considering the NAR property, our model simply uses the greedy search to generate final results.

B. Overall Results

In this section, we compare our model with other end-to-end methods. Transformer [8] conducts experiments on low-resource tasks (MA and EN) by supervised transfer learning. W2v-ctc [35] also applies pre-trained wav2vec2.0 as encoder and adds a randomly initialized linear projection on top of the encoder. W2v-ctc is optimized by minimizing a CTC loss, and it is one of the most concise end-to-end models [36].

[1]https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_small.pt

[2]https://storage.googleapis.com/bert_models/2020_02_20/uncased_L-12_H-768_A-12.zip

[3]https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip

[4]https://github.com/pytorch/fairseq/tree/master/examples/wav2vec

TABLE I
COMPARISON ON CALLHOME AND HKUST. PERFORMANCE IS CER (%) FOR MA AND HKUST; WER (%) FOR EN

Model	MA(15h)	EN(15h)	HKUST(150h)
CIF [7]	-	-	23.09
Transformer + MPC [14]	-	-	**21.70**
Transformer [8], [37]	37.62	33.77	26.60
w2v-ctc [35]	36.06	24.93	23.80
w2v-seq2seq			
decoder with 1 blocks	39.81	26.18	24.06
+ cold fusion	37.90	-	24.02
decoder with 4 blocks	54.82	47.66	25.73
+ cold fusion	-	-	25.46
w2v-nar			
decoder with 1 blocks	51.34	54.09	50.35
decoder with 4 blocks	46.42	46.18	27.08
w2v-cif-bert	**32.93**	**23.79**	22.92

In this work, we reproduce w2v-seq2seq [35], which is composed of pre-trained wav2vec2.0 and Transformer decoder (with 1 or 4 blocks, randomly initialized) along with cross-attention [28]. We also train and infer w2v-seq2seq models in the NAR way [25], which is marked as w2v-nar. These models apply the same modeling units (character for Chinese and subword for English) as [35]. We further implement *cold fusion* for w2v-seq2seq with pre-trained Transformer LM (6 blocks) on a private Chinese text corpus (200 M samples). W2v-seq2seq models decode through the beam search with a size of 50. All of these models are trained under the same setups as w2v-cif-bert.

As demonstrated in Table I, our model achieves best results on low-resource tasks, showing the promising direction to fuse pre-trained acoustic and linguistic modules. On the relatively abundant labeled ASR task, our model still achieves comparable performance, where the MPC training [14] utilizes a private large speech corpus (10 000 h) that is similar to the target task.

W2v-seq2seq models are inferior to w2v-ctc, even with cold-fusion. So do w2v-nar models. We think it is the random-initialized decoder that causes the low performance under the low-resource condition. Compared with cold-fusion, our method can make better use of pre-trained LM.

C. Ablation Study

We explore different structures and hyper-parameters of w2v-cif-bert by ablations to find a reasonable setting. Some connections in Fig. 1 are shut off by setting corresponding weights to 0. Ablations are conducted on CALLHOME-MA.

Results are listed in Table II. We get the following conclusions according to the corresponding ablations:

1) The quantity loss is inevitable for the CIF mechanism since a proper alignment between the acoustic and linguistic representation is hard to learn through other supervisions;
2) Adding the auxiliary CTC criterion greatly matters. It can help the encoder to learn the alignment;
3) BERT as linguistic encoder makes an impressive contribution to the performance, showing our effective utilization of pre-trained masked LM;
4) The acoustic high-way connection is inevitable for the performance convergence of our model.

TABLE II
ABLATIONS ON STRUCTURE OF W2V-CIF-BERT OVER CALLHOME MA. PERFORMANCE IS CER (%) ON TEST SET

Description	Settings	CER
w2v-cif-bert	$\mu_1 = 0.2$ $\mu_2 = 1.0$ $\lambda_{\mathrm{LM}} = 0.2$ $\lambda_{\mathrm{AC}} = 1.0$ no sharing to_vocab $0.9 \rightarrow 0.2/4000$ $TH = 0.8$	**32.93**
(1) quantity loss	$\mu_1 = 0$	154.7
	$\mu_1 = 0.5$	33.05
(2) CTC loss	$\mu_2 = 0$	35.77
	$\mu_2 = 2.0$	33.01
(3) LM weight	$\lambda_{\mathrm{LM}} = 0.0$	36.43
	$\lambda_{\mathrm{LM}} = 0.4$	33.22
(4) acoustic weight	$\lambda_{\mathrm{AC}} = 0.0$	98.79
(5) share to_vocab	share 0,1	33.00
	share 0,2	78.76
	share 1,2	35.10
(6) gold rate schedule	$0.9 \rightarrow 0.2/8000$	34.13
	$0.9 \rightarrow 0.2/2000$	33.37
	$0.9 \rightarrow 0.0/4000$	34.92
	$0.2 \rightarrow 0.2/\infty$	33.26
	$0.0 \rightarrow 0.0/\infty$	35.95
(7) confidence threshold	$TH = 1.0$	33.37
	$TH = 0.6$	35.51

5) Using three separate-updating "to_vocab" modules is the best option.

During fine-tuning, we apply a trick of mixing $\boldsymbol{h}^{\mathrm{LM}}$ with embedded labels at a scheduled sampling rate p. we explore different schedules of p in (6). "$\rightarrow$" indicates the range of p, and the number after "/" is the decreasing step. Directly fine-tuning without any mixing ($0.0 \rightarrow 0.0/\infty$) achieves a rather poor performance, demonstrating the necessity of our proposed schedule fusing trick. A reasonable schedule of gold rate p during fine-tuning is decreasing from a high value to a low one. Keeping $p > 0$ is another key point. We explain that it can make BERT work better by preserving some anchor tokens.

During inference, we also add some anchor tokens according to the confidence of the acoustic output logits$_{\mathrm{AC}}$, which is similar to the iterative decoding of NAR [24]. Tokens are determined when their post-probabilities surpass a customized threshold TH ($TH = 1.0$ means no anchor tokens). Then the embedding vectors of these tokens are mixed with $\boldsymbol{h}^{\mathrm{LM}}$. As we can see in (7), a proper confidence threshold, $TH = 0.8$ in our best result, contributes to the better performance during inference.

V. CONCLUSION

In this work, we propose an end-to-end model for low-resource ASR tasks. It integrates separately pre-trained acoustic and linguistic modules through a monotonic attention mechanism and minimal parameters. Along with the scheduled modality fusion, our model can achieve remarkable results with a fast convergence speed.

REFERENCES

[1] S. Zhou *et al.*, "Multilingual recurrent neural networks with residual learning for low-resource speech recognition," in *Proc. Interspeech*, 2017, pp. 704–708.

[2] A. Baevski and A. Mohamed, "Effectiveness of self-supervised pre-training for ASR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 7694–7698.

[3] A. Baevski, S. Schneider, and M. Auli, "Vq-wav2vec: Self-supervised learning of discrete speech representations," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 12449–12460.

[4] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 12449–12460.

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

[6] V. Pratap *et al.*, "WAV2LETTER++: The fastest open-source speech recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6460–6464.

[7] L. Dong and B. Xu, "CIF: Continuous integrate-and-fire for end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6079–6083.

[8] S. Zhou, S. Xu, and B. Xu, "Multilingual end-to-end speech recognition with a single transformer on low-resource languages," 2018, *arXiv:1806.05059*.

[9] Y. zhao, J. Li, X. Wang, and Y. Li, "The speechtransformer for large-scale Mandarin Chinese speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 7095–7099.

[10] C. Tran, C. Wang, Y. Tang, Y. Tang, J. Pino, and X. Li, "Cross-modal transfer learning for multilingual speech-to-text translation," 2020, *abs/2010.12829*.

[11] L. Dong *et al.*, "A comparison of label-synchronous and frame-synchronous end-to-end models for speech recognition," 2020, *arXiv:2005.10113*.

[12] G. E. Hinton and R. R. Salakhutdinov, "A better way to pretrain deep Boltzmann machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2447–2455.

[13] Y.-A. Chung, H.-Y. Lee, and J. Glass, "Supervised and unsupervised transfer learning for question answering," in *Proc. Conf. North Amer. Chap. Assoc. Comput. Linguist.: Human Lang. Technol.* (Long Papers)}, 2018, vol.1, pp. 1585–1594.

[14] D. Jiang *et al.*, "Improving transformer-based speech recognition using unsupervised pre-training," 2019, *arXiv:1910.09932*.

[15] J. Gu, J. Bradbury, C. Xiong, V. O. Li, and R. Socher, "Non-autoregressive neural machine translation," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1l8BtlCb

[16] S. Toshniwal, A. Kannan, C. Chiu, Y. Wu, T. N. Sainath, and K. Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2018, pp. 369–375.

[17] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," in *Proc. Interspeech* 2018, pp. 387–391.

[18] C. Shan *et al.*, "Component fusion: Learning replaceable language model component for end-to-end speech recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 5361–5635.

[19] Y. Liu *et al.*, "Multilingual denoising pre-training for neural machine translation," *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 726–742, 2020.

[20] A. H. Liu, H. Lee, and L. Lee, "Adversarial training of end-to-end speech recognition using a criticizing language model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 6176–6180.

[21] S. Karita, S. Watanabe, T. Iwata, A. Ogawa, and M. Delcroix, "Semi-supervised end-to-end speech recognition," in *Proc. Interspeech*, 2018, pp. 2–6.

[22] M. K. Baskar, S. Watanabe, R. Astudillo, T. Hori, L. Burget, and J. Černockỳ, "Semi-supervised sequence-to-sequence ASR using unpaired speech and text," in *Proc. Interspeech*, 2019, pp. 3790–3794.

[23] Y. Chen, W. Wang, and C. Wang, "Semi-supervised asr by end-to-end self-training," in *Proc. Interspeech*, 2020, pp. 2787–2791.

[24] Y. Higuchi, S. Watanabe, N. Chen, T. Ogawa, and T. Kobayashi, "Mask CTC: Non-autoregressive end-to-end ASR with CTC and mask predict," in *Proc. Interspeech*, 2020, pp. 3655–3659.

[25] N. Chen, S. Watanabe, J. Villalba, and N. Dehak, "Listen and fill in the missing letters: Non-autoregressive transformer for speech recognition," 2019, *arXiv:1911.04908*.

[26] C. Yi, F. Wang, and B. Xu, "Ectc-Docd: An end-to-end structure with CTC encoder and OCD decoder for speech recognition," in *Proc. Interspeech*, 2019, pp. 4420–4424.

[27] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[28] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, vol. 1, 2019, pp. 4171–4186.

[30] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=H1eA7AEtvS

[31] Z. Yang *et al.*, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5753–5763.

[32] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 4835–4839.

[33] Y. Liu, P. Fung, Y. Yang, C. Cieri, S. Huang, and D. Graff, "HKUST/MTS: A very large scale Mandarin telephone speech corpus," in *Proc. Int. Symp. Chin. Spoken Lang. Process.*, 2006, pp. 724–735.

[34] M. Ott *et al.*, "FAIRSEQ: A fast, extensible toolkit for sequence modeling," in *Proc. Conf. North Ameri. Chap. Assoc. Comput. Linguist. Human Lang. Technol.*, (Long Papers) 2019, vol. 1, pp. 48–53.

[35] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, "Applying Wav2vec2.0 to speech recognition in various low-resource languages," 2020, *arXiv:2012.12121*.

[36] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[37] S. Zhou, L. Dong, S. Xu, and B. Xu, "A comparison of modeling units in sequence-to-sequence speech recognition with the transformer on Mandarin Chinese," in *Proc. Int. Conf. Neural Inf. Process.*, 2018, pp. 210–220.