SPEECH-TEXT BASED MULTI-MODAL TRAINING WITH BIDIRECTIONAL ATTENTION FOR IMPROVED SPEECH RECOGNITION

Yuhang Yang1, Haihua Xu2*, Hao Huang1,5, Eng Siong Chng3, Sheng Li4

1School of Information Science and Engineering, Xinjiang University, Urumqi, China
2Bytedance
3School of Computer Science and Engineering, Nanyang Technological University, Singapore
4National Institute of Information and Communications Technology (NICT), Kyoto, Japan
5Xinjiang Provincial Key Laboratory of Multi-lingual Information Technology, Urumqi, China

huanghao@xju.edu.cn

ABSTRACT

To let the state-of-the-art end-to-end ASR model enjoy data efficiency, as well as much more unpaired text data by multi-modal training, one needs to address two problems: 1) the synchronicity of feature sampling rates between speech and language (aka text data); 2) the homogeneity of the learned representations from two encoders. In this paper we propose to employ a novel bidirectional attention mechanism (BiAM) to jointly learn both ASR encoder (bottom layers) and text encoder with a multi-modal learning method. The BiAM is to facilitate feature sampling rate exchange, realizing the quality of the transformed features for the one kind to be measured in another space, with diversified objective functions. As a result, the speech representations are enriched with more linguistic information, while the representations generated by the text encoder are more similar to corresponding speech ones, and therefore the shared ASR models are more amenable for unpaired text data pretraining. To validate the efficacy of the proposed method, we perform two categories of experiments with or without extra unpaired text data. Experimental results on Librispeech corpus show it can achieve up to 6.15% word error rate reduction (WERR) with only paired data learning, while 9.23% WERR when more unpaired text data is employed.

Index Terms— Speech recognition, end-to-end, bidirectional attention, forced alignment, multi-modal, representation

1. INTRODUCTION

End-to-end (E2E) automatic speech recognition (ASR) framework [1–5] has now come into predominance in both research and product areas [6–8] thanks to its efficacy in modeling capacity, as well as compactness. However, one of the limitations of E2E ASR modeling is its insatiable data-hungry [9]. To train a decent ASR system, the rule of thumb is always the more data the better.

To get more data, one would first consider collecting more human-transcribed data, the so-called paired data. Unfortunately, such data comes with high costs. As a result, ASR models are usually trained with limited paired data. The alternative is to get more unpaired data at a lower cost instead, in terms of either unpaired speech data or unpaired text data accordingly. For unpaired speech data exploitation, one can employ unsupervised pretraining [10–13] or self-training [14–18] to yield improved ASR performance, while to take advantage of unpaired text data, people have many options for obtaining better ASR models.

In order to well exploit text data, one of the simplest ways is to employ text data to train an external language model (LM) [19, 20] that is fused with ASR system, yielding improved results. Besides, given a unpaired text data set, people can employ a text-to-speech (TTS) system to generate synthesized paired speech-text data [21–23]. However, the challenge is to obtain an off-the-shelf TTS system that yielding diversified speech data is a nontrivial task.

More recently, multi-modal training has been widely explored to realize training an ASR model with both speech and text (either paired or unpaired) data simultaneously [24–26]. The difficulties here lie in two aspects: 1) The synchronicity of feature sampling rates between speech and text/language, namely, speech sampling rate is much faster than language ones, and hence how to synchronize them is a problem, denoted as AliProblem-1 for brevity; 2) The homogeneity of the learned representations from two encoders, that is, since the ASR encoder hidden representations have different distributions with those obtained from the text encoder, how to make the two representations similar is also a problem, and it is denoted as AliProblem-2.

For the above-mentioned multi-modal training, [25] employs a conventional HMM-DNN model to obtain phone level alignment for the transcript of the paired data, and the duration estimation model is used for the unpaired text data to solve AliProblem-1. To address the AliProblem-2, one can introduce diverse objective loss functions, such as masked LM (MLM), connectionist temporal classification (CTC), as well as cosine distance loss functions, etc., to make the two learned representations closer to each other.

In this paper, we propose a novel speech-text based multi-modal training approach to boost ASR performance, using a modified bidirectional attention mechanism (BiAM) [27] that facilitates the solution of both AliProblem-1 and AliProblem-2 with a joint training manner. The framework of the proposed method is illustrated in Figure 1. By BiAM, we can mutually transform one kind of representation (aka embedding) into another representational space. Specifically, we can transform language representation (aka text embedding) into speech space, as well as transform speech representation (aka speech embedding) into language representational space. By such a transformation, we can solve the AliProblem-1. Meanwhile, we employ a series of loss functions, such as CTC loss, cosine distance losses, as well as MLM loss, to make the two transformed features closer, hence addressing the AliProblem-2. Concretely, once we employ the BiAM to transform the text embedding into

* Authors have equal contributions. Hao Huang is the correspondence author. This work was supported by the National Key R&D Program of China (2020AAA0107902).

1Source code: https://github.com/yuhangear/Multi-modal-learning.git
2. RELATION TO PRIOR WORK
Speech-text based multi-modal training for end-to-end ASR has become popular for a while [24–26, 28–32]. [24] directly merge the two embeddings generated from both encoders to train the shared ASR encoders. To solve both problems as mentioned, [29] proposed to use the embeddings from text encoder as query while speech embeddings more appropriate for the speech encoder. Conversely, the transformed speech embedding into language representational space is measured with CTC and MLM losses, respectively, such that the feature similarity issues, the text encoder is learned to generate embeddings enriched with linguistic information.

Our contributions can be summarized as follows: 1) To the best of our knowledge, we are the first to employ the bidirectional attention mechanism for speech-text-based multi-modal training to boost ASR performance. 2) To train text encoder, we advocate grapheme instead of phoneme sequence to learn text encoder, which makes the proposed method language agnostic. 3) We demonstrate its efficacy on Librispeech data with diverse configurations.

3. METHODOLOGY
3.1. Multi-modal learning framework
The whole framework is illustrated in Figure 1, which is composed of three modalities, one is Conformer-based [2] ASR model, and the second is text encoder using Transformer, while the third is the modified bidirectional attention module [27], namely BiAM, which accepts both speech and text encoder embeddings as the inputs.

The entire network is trained with two category losses, one is the ASR loss function, and the others are loss functions denoted as $L_{\text{ALI}}$, facilitating the alignment optimization between two embeddings with BiAM. For clarity, the overall losses are expressed as follows:

$$L_{\text{multi-modal}} = L_{\text{ASR}} + \alpha L_{\text{ALI}}$$  \hspace{1cm} (1)

$$L_{\text{ASR}} = \lambda L_{\text{CTC}} + (1 - \lambda) L_{\text{Attention}}$$  \hspace{1cm} (2)

$$L_{\text{ALI}} = L_{\text{cd}}(Y_{\text{aligned}}, X) + L_{\text{MLM}}(X_{\text{aligned}}, Y_{\text{grapheme}}) + L_{\text{ALI}}(\text{Sampler}(X, Y_{\text{aligned}}), Y_{\text{grapheme}})$$  \hspace{1cm} (3)

where $Y_{\text{grapheme}} \in \mathbb{R}^{n_2}$ is the grapheme sequence generated from the input text data, and $n_2$ is the sequence length in grapheme. Correspondingly, we denote the speech embedding length as $n_1$ in the following. Besides, we fix $\alpha = 0.1$, and $\lambda = 0.3$ in the following experiments.

Similarly in Equation 3, both $X \in \mathbb{R}^{n_1 \times d}$ and $Y \in \mathbb{R}^{n_2 \times d}$ are embedding sequences of the ASR and text encoders respectively, while $X_{\text{aligned}} = \text{BiAM}(X)$, and $Y_{\text{aligned}} = \text{BiAM}(Y)$ with $X_{\text{aligned}} \in \mathbb{R}^{n_1 \times d}$, $Y_{\text{aligned}} \in \mathbb{R}^{n_2 \times d}$, and again $n_1$ and $n_2$ being speech and grapheme embedding length respectively. One can refer to Section 3.2 for the details of the explanation of the BiAM.

Besides, in Equation 3, “cd” stands for cosine distance, and $g$CTC stands for grapheme CTC. For gCTC training, we employ a “Sampler” to sample both $X_{\text{aligned}}$ and $Y'$ for each mini-batch training. As mentioned, the MLM in Equation 3 refers to masked LM.

We note that the speech embeddings are from the bottom 8th layer of the Conformer in practice, while the grapheme embeddings are output from the final layer of the text encoder instead. Finally, after training, only the ASR modality serves for recognition in Figure 1.

3.2. Bidirectional attention mechanism
To solve the alignment problem between the length of the paired speech and text embeddings (AllProblem-1 here), [27] recently proposed a bidirectional attention mechanism (BiAM) realizing a neural forced-alignment (NeuFA) method. Inspired by [27], we propose a simpler one for the speech-text multi-modal training. Specifically, we make $K_1 = V_1$ and $K_2 = V_2$, as well as the compatibility function being defined as matrix dot product in [27]. In other words, we do not generate key-value pairs, and we directly use text and speech embeddings for dot product operation to generate the shared attention matrix instead. Consequently, the BiAM is implemented as follows. Rewrite speech embedding sequence $X \in \mathbb{R}^{n_1 \times d}$ as $X^{n_1 \times d}$ for notational clarity. Likewise, the corresponding text embedding sequence $Y \in \mathbb{R}^{n_2 \times d}$ is rewritten as $Y^{n_2 \times d}$ and $n_1 \neq n_2$. To begin with the bidirectional attention transformation, we first obtain the shared attention matrix $A$ as:

$$A = X^{n_1 \times d} \times (Y^{n_2 \times d})^T$$  \hspace{1cm} (4)

where $A \in \mathbb{R}^{n_1 \times n_2}$. Then we perform the softmax operation on $A$ and $A^T$ to obtain:

$$W_{12}, W_{21} = \text{softmax}(A, A^T)$$  \hspace{1cm} (5)

For simplicity, we ensure the dimension of speech and text embeddings are equal to $d$. 

Fig. 1. Speech-text based multi-modal learning framework with Bidirectional attention mechanism (BiAM). After training, all the stuff in the dashed-line box will be removed.
Fig. 2. The diagram of bidirectional attention mechanism (BiAM), where $X_{n1 \times d}$ and $Y_{n2 \times d}$ are speech and text embedding sequences respectively.

where $W_{12} \in \mathbb{R}^{n1 \times n2}$, and $W_{21} \in \mathbb{R}^{n2 \times n1}$. Now, we can obtain two outputs as aligned embedding with the following transformation:

$$X_{\text{aligned}}^{n2 \times d} = W_{21} \times X_{n1 \times d}$$  \hspace{1cm} (6)

$$Y_{\text{aligned}}^{n1 \times d} = W_{12} \times Y_{n2 \times d}$$  \hspace{1cm} (7)

where $X_{\text{aligned}}^{n2 \times d}$ and $Y_{\text{aligned}}^{n1 \times d}$ are the two final outputs by the BiAM. From Equations 6 and 7, BiAM realizes two transformations $W_{12}$ and $W_{21}$. The former transforms the speech embeddings, yielding the “aligned” speech sequence with the same length as the grapheme embedding length $n_2$. Likewise, the latter do the opposite operation, with the “aligned” text sequence having the same length as the corresponding speech $n_1$. Consequently, once we use the BiAM module to convert text embeddings to speech embeddings and vice versa, we have the flexibility to employ different loss functions, such as $L_{cd}$, $L_{MLM}$, and $L_{gCTC}$ etc. in Equation 3 to measure the quality of these conversions.

The key point of the BiAM lies in the so-called compatibility function definition in [27]. Here, it is defined as two embedding sequence dot product computation as Equation 4, which actually is the pair-wise dot product distance between the two embedding sequences. Once the matrices $A$ and $A^T$ are transformed to posterior matrix using softmax operation, they can act as attention mechanism on the input embeddings, yielding a kind of forced alignment. The details of the BiAM computation are illustrated in Figure 2.

### 3.3. Training process

The whole network in Figure 1 is trained using Equation 1 as the loss function. In practice, we first train the network with paired speech-text data, and both the ASR model and text encoder are jointly trained. During this stage, the cosine distance loss $L_{cd}$ in Equation 3 is only employed at later training steps for the sake of stable training.

Once the training with the paired speech-text data is finished, the embeddings generated with the bottom layer of the ASR encoder have been enriched with more linguistic information that are not only speaker but also ambiance independent, such that it leads to improved ASR performance. After training the network using paired speech-text data, we can optionally use unpaired text data to further improve the model. To do this, we feed the unpaired text data $Y_{\text{unpaired-grapheme}}$ into the text encoder and then replicate the resulting embeddings twice. These replicated embeddings are then used as input to the 8th layer of the ASR encoder. This is possible because our text encoder has already been taught for how to generate grapheme embeddings that are closer to the speech ones with the losses in Equation 3. After the unpaired text data training, we should fine-tune the network using the paired speech-text data again. However, it is not possible to perform BiAM-based training during unpaired training because there is no speech input. Thus, we used the method proposed in [34] and replicated each grapheme embedding twice during unpaired training.

### 4. EXPERIMENTS

#### 4.1. Data

All of the experiments are conducted on the LibriSpeech [35] corpus. Train data consists of 100 hours of train clean data, as well as 960 hours of full train data. Test sets consist of 4 data sets, namely, dev-clean, dev-other, test-clean, and test-other. Overall we conduct two kinds of experiments. One is using 100 hours of train clean data, with or without 960 hours of transcript as unpaired text data. The other experiments are performed on the full 960 hours of train data.

#### 4.2. Modeling

All experiments are conducted with Esptnet toolkit [36]. The ASR model is Conformer with 12-layer encoder and 6-layer Transformer-based decoder. We use a smaller ASR model for 100-hour clean train data, while a bigger one for the 960 hour full training data. The differences lie in the middle layer, attention and word embedding dimensions, as well as multi-head attention heads, {1024, 256, 256, 4} for the smaller model versus {2048, 512, 512, 8} for the bigger model. The input features are 80-dimensional filter-bank, and the output is word piece models with 5000 subwords. The text encoder uses Transformer framework with 3- and 6-layer for 100- and 960 hour train data respectively. The differences between smaller and bigger models are the same as those of the ASR models. We use 0.002 learning rate for the multi-modal training on a single GPU (v100), with the 0.1 dropout. The whole network is trained with 80 epochs, and after 70 epochs the cosine distance loss is enabled with 10 epochs continuing training. For the grapheme-based CTC training, we sample between the aligned speech and the text embeddings in each mini-batch, with each occupying 50% samples. For the MLM training, we randomly mask 20% graphemes for each utterance.

For the unpaired text pretraining, the output text embeddings are taken as input to the 8th layer of the ASR encoder. During training, the parameters of the first 8 layers of the speech encoder are frozen, and the remaining parameters are updated. After that, we use a 0.001 learning rate to fine-tune the ASR network with the paired speech-text data.

For inference, the beam sizes are 20 and 60 for the 100- and 960 hour train data, respectively.

#### 4.3. Results

##### 4.3.1. Results on the 100-hour train data

Table 1 presents the results of the multi-modal training using the 100-hour train data.
Table 1. WERs(%) of the proposed BiAM-based multi-modal training with the 100-hour train data, “cd” refers to cosine distance loss

|                | Dev WER (%) | Test WER (%) |
|----------------|-------------|--------------|
|                | clean       | other        | clean       | other        |
| Baseline       | 6.3         | 17.4         | 6.5         | 17.3         |
| Grapheme CTC   | 6.2         | 17.1         | 6.2         | 17.0         |
| BiAM (w/o cd)  | 6.1         | 16.9         | 6.2         | 16.6         |
| BiAM (w/ cd)   | 6.0         | 16.7         | 6.1         | 16.4         |

From Table 1, the proposed method achieves significant WER reductions (WERR) on the four test sets, with WERRs of 4.76%, 4.02%, 6.15%, and 5.20% on dev-clean, dev-other, test-clean, and test-other, respectively, compared to the baseline. Furthermore, we found that cosine distance loss is very essential to get improved results over that case where only gCTC and MLM losses are employed. We also compare the proposed method with a multi-task learning method, namely intermediate gCTC from the 8th layer of the Conformer encoder, named “Grapheme CTC” in Table 1. From Table 1, though the “Grapheme CTC” is also very effective, the proposed method has achieved consistent performance improvement. In what follows, we abbreviate the proposed BiAM with cosine distance loss as BiAM.

Table 2 reports the WERs of the proposed method with the 100-hour train data using 960 hour train transcript as unpaired text data.

|                | Dev WER (%) | Test WER (%) |
|----------------|-------------|--------------|
|                | clean       | other        | clean       | other        |
| Baseline       | 6.0         | 16.7         | 6.1         | 16.4         |
| BiAM + unpaired text | 6.0       | 16.5         | 5.9         | 16.3         |

Given the unpaired text pretraining, Table 2 reveals the proposed method gets further WERR improvement on 3 out of 4 test sets over the paired speech-text training method (see Table 1). Specifically, the WERRs are 4.76%, 5.17%, 9.23%, and 5.78% on the four test sets over the baseline model. We notice that the unpaired text data pretraining has limited contribution to performance improvement. We think the following reason mainly accounts for this. During the unpaired text pretraining, we cannot get the transform \( W_{12} \) in Eq. 5 for each unpaired utterance. Actually, \( W_{12} \) is not only “diagonal” but also contains duration information for each grapheme. We are putting more effort on this in future.

Table 3 reports WERs of the proposed method using 960 hour train data.

|                | Dev WER (%) | Test WER (%) |
|----------------|-------------|--------------|
|                | clean       | other        | clean       | other        |
| Baseline       | 2.1         | 5.2          | 2.4         | 5.3          |
| Grapheme CTC   | 2.1         | 5.2          | 2.4         | 5.2          |
| BiAM           | 2.0         | 5.0          | 2.3         | 5.0          |

What is shown in Table 3 again validates the efficacy of the proposed method for speech-text-based multi-modal training. It has achieved consistent WERR improvement over the baseline model. The WERRs are 4.76%, 3.85%, 4.17% and 5.66% on the four test sets, respectively. Besides, compared with the intermediate CTC-based multi-task learning method, the proposed method also has a clear improvement margin.

To see if the model has successfully learned the speech-to-text alignment with the help of the BiAM module, Figure 3 plots the alignment matrix after softmax operation, namely \( W_{12} \) in Equation 5. From Figure 3, we can see the clear monotonic alignment pattern between the text and speech sequences, which again validates the effectiveness of the BiAM method. In addition, the breakpoints in the alignment correspond to the “Blank” label in Figure 3.

5. DISCUSSION & CONCLUSION

The above experimental results show that the proposed bidirectional attention mechanism has clear advantages for speech-text forced-alignment learning, yielding improved ASR performance in a speech-text multi-modal training framework. However, the exploration is still far from perfect, and the limitations are at least as follows. 1) The effectiveness of the unpaired text pretraining is not fully demonstrated, particularly for full exploitation of the text data provided by Librispeech corpus is yet to be done. 2) Unpaired text pretraining method also needs a revisit in depth. So far, the pretraining is a mismatched one, yielding under-performed results. To realize a matched pretraining, we need to figure out an approach to reconstruct the transform \( W_{12} \) in Eq. 5 for each unpaired utterance. Actually, \( W_{12} \) is not only “diagonal” but also contains duration information for each grapheme. We are putting more effort on this in future.

To conclude the work in this paper, we have proposed a speech-text-based multi-modal training framework for improving ASR performance via a bidirectional attention mechanism. We demonstrated its efficacy on Librispeech corpus with both 100 and 960 hour train data, respectively. With the paired speech-text-based multi-modal training, the proposed method has achieved up to 6.15% and 5.66% WERR reductions on 4 test sets under the two scenarios. Besides, on the 100-hour low-resource data, we also demonstrated the effectiveness of the proposed method for unpaired text data pretraining. Future work will be focused on efficient unpaired text data pretraining.
6. REFERENCES

[1] W. Chan, N. Jaitly, et al., “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in Proc. of ICASSP, pp. 4960–4964, IEEE, 2016.

[2] A. Gulati, J. Qin, et al., “Conformer: Convolution-augmented Transformer for Speech Recognition,” in Proc. Interspeech 2020, pp. 5036–5040, 2020.

[3] Y. Hu, Hou, et al., “Interactive feature fusion for end-to-end noise-robust speech recognition,” in Proc. of ICASSP, IEEE, 2022.

[4] C. Chen, N. Hou, et al., “Noise-robust speech recognition with 10 minutes unparalleled in-domain data,” in Proc. of ICASSP, IEEE, 2022.

[5] C.-F. Zhang, Y. Liu, et al., “Non-autoregressive transformer with unified bidirectional decoder for automatic speech recognition,” in Proc. of ICASSP, pp. 6527–6531, IEEE, 2022.

[6] Y. He, T. N. Sainath, R. Prabhavalkar, et al., “Streaming end-to-end speech recognition for mobile devices,” in ICASSP, IEEE, 2019.

[7] T. N. Sainath, Y. He, B. Li, et al., “A streaming on-device end-to-end model surpassing server-side conventional model quality and latency,” in Proc. of ICASSP, IEEE, 2020.

[8] B. Li, T. N. Sainath, et al., “A language agnostic multilingual streaming on-device asr system,” arXiv:2208.13916, 2022.

[9] B. Li, R. Pang, Y. Zhang, T. N. Sainath, et al., “Massively multilingual asr: A lifelong learning solution,” in Proc. of ICASSP, IEEE, 2022.

[10] S. Ling, Y. Liu, et al., “Deep contextualized acoustic representations for semi-supervised speech recognition,” in Proc. of ICASSP, pp. 6429–6433, IEEE, 2020.

[11] A. T. Liu, S.-w. Yang, et al., “Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders,” in Proc. of ICASSP, pp. 6419–6423, IEEE, 2020.

[12] W.-N. Hsu, B. Bolte, et al., “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] G. Zheng, Y. Xiao, et al., “Wav-bert: Cooperative acoustic and linguistic representation learning for low-resource speech recognition,” arXiv preprint arXiv:2109.09161, 2021.

[14] Y.-A. Chung, W.-N. Hsu, et al., “An unsupervised autoregressive model for speech representation learning,” 2019.

[15] A. Baevski, S. Schneider, and M. Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” arXiv preprint arXiv:1910.05453, 2019.

[16] A. Baevski, Y. Zhou, A. Mohamed, et al., “wav2vec 2.0: A framework for self-supervised learning of speech representations,” Advances in Neural Information Processing Systems, vol. 33, pp. 12449–12460, 2020.

[17] S. Chen, C. Wang, et al., “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” IEEE Journal of Selected Topics in Signal Processing, 2022.

[18] A. Baevski, W.-N. Hsu, et al., “Data2vec: A general framework for self-supervised learning in speech, vision and language,” arXiv preprint arXiv:2202.03555, 2022.

[19] J. Chorowski and N. Jaitly, “Towards better decoding and language model integration in sequence to sequence models,” arXiv preprint arXiv:1612.02695, 2016.

[20] A. Siriram, H. Jun, et al., “Cold fusion: Training seq2seq models together with language models,” arXiv preprint arXiv:1708.06426, 2017.

[21] T. Hori, R. Astudillo, T. Hayashi, et al., “Cycle-consistency training for end-to-end speech recognition,” in Proc. of ICASSP, pp. 6271–6275, IEEE, 2019.

[22] G. Wang, A. Rosenberg, Z. Chen, et al., “Improving speech recognition using consistent predictions on synthesized speech,” in Proc. of ICASSP, pp. 7029–7033, IEEE, 2020.

[23] Y. Huang, H.-K. Kuo, S. Thomas, et al., “Leveraging unpaired text data for training end-to-end speech-to-intent systems,” arXiv:2010.04284, 2020.

[24] Y. Tang, J. Pino, et al., “A general multi-task learning framework to leverage text data for speech to text tasks,” in Proc. of ICASSP, IEEE, 2021.

[25] W. Wang, S. Ren, Y. Qian, et al., “Optimizing alignment of speech and language latent spaces for end-to-end speech recognition and understanding,” in Proc. of ICASSP, IEEE, 2021.

[26] Z. Chen, Y. Zhang, et al., “Maestro: Matched speech text representations through modality matching,” 2022.

[27] J. Li, Y. Meng, Z. Wu, et al., “Neufa: Neural network based end-to-end forced alignment with bidirectional attention mechanism,” in Proc. of ICASSP, pp. 8007–8011, IEEE, 2022.

[28] A. Renduchintala, S. Ding, et al., “Multi-modal data augmentation for end-to-end asr,” arXiv:1803.10299, 2018.

[29] J. Drexler and J. Glass, “Explicit alignment of text and speech encodings for attention-based end-to-end speech recognition,” in Proc. of ASRU, pp. 913–919, ASRU, 2019.

[30] K. Deng, S. Cao, et al., “Improving ctc-based speech recognition via knowledge transferring from pre-trained language models,” in Proc. of ICASSP, pp. 8517–8521, IEEE, 2022.

[31] J. Ao, R. Wang, et al., “Speech5: Unified-modal encoder-decoder pretraining for spoken language processing,” arXiv:2110.07205, 2021.

[32] Z. Chen, Y. Zhang, A. Rosenberg, et al., “Tts4pretrain 2.0: Advancing the use of text and speech as pretraining with consistency and contrastive losses,” in Proc. of ICASSP, IEEE, 2022.

[33] L. Dong and B. Xu, “Cif: Continuous integrate-and-fire for end-to-end speech recognition,” in Proc. of ICASSP, pp. 6079–6083, IEEE, 2020.

[34] C. Gao, G. Cheng, et al., “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), pp. 6543–6547, IEEE, 2021.

[35] K. Krishna, L. Lu, K. Gimpel, and K. Livescu, “A study of all-convolutional encoders for connectionist temporal classification,” in Proc. ICASSP 2018.

[36] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J.Heymann, M. Wiesner, N. Chen, et al., “Espnet: End-to-end speech processing toolkit,” arXiv preprint arXiv:1804.00015, 2018.