OPTIMIZING ALIGNMENT OF SPEECH AND LANGUAGE LATENT SPACES FOR END-TO-END SPEECH RECOGNITION AND UNDERSTANDING

Wei Wang[1,2,*] *, Shuo Ren*[2]*, Yao Qian*[2]*, Shujie Liu*[2]*, Yu Shi*[2]*, Yanmin Qian*[1]*, Michael Zeng*[2]

[1] MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Microsoft Corporation

ABSTRACT

The advances in attention-based encoder-decoder (AED) networks have brought great progress to end-to-end (E2E) automatic speech recognition (ASR). One way to further improve the performance of AED-based E2E ASR is to introduce an extra text encoder for leveraging extensive text data and thus capture more context-aware linguistic information. However, this approach brings a mismatch problem between the speech encoder and the text encoder due to the different units used for modeling. In this paper, we propose an embedding aligner and modality switch training to better align the speech and text latent spaces. The embedding aligner is a shared linear projection between text encoder and speech encoder trained by masked language modeling (MLM) loss and connectionist temporal classification (CTC), respectively. The modality switch training randomly swaps speech and text embeddings based on the forced alignment result to learn a joint representation space. Experimental results show that our proposed approach achieves a relative 14% to 19% word error rate (WER) reduction on LIBRISPEECH ASR task. We further verify its effectiveness on spoken language understanding (SLU), i.e., an absolute 2.5% to 2.8% F1 score improvement on SNIPS slot filling task.

Index Terms— speech recognition, multi-modality, end-to-end

1. INTRODUCTION

Since the emergence of end-to-end (E2E) models, the automatic speech recognition (ASR) pipeline has been greatly simplified, and ASR tasks can be accomplished with a unified model architecture [1, 2]. The most commonly adopted E2E ASR architectures are attention-based encoder-decoder models [3, 4]. In these models, the encoder plays an essential role, which converts the acoustic information to context-aware linguistic features. The decoder then generates the formal text output based on the linguistic features.

In recent years, many attempts have been made to assist the encoder in learning context-aware linguistic information with large text corpora [5, 6, 7, 8, 9]. An E2E ASR model with an extra text encoder network is a commonly used architecture to integrate more linguistic information into the ASR encoder. [7] incorporates a smoothed L1 loss with a multi-stage training scheme and trains the text encoder and speech encoder to match their output to each other, pushing the speech embeddings closer to the text embedding space.

Instead of casting explicit constraints, in [5], the text encoder shares part of its layers with the speech encoder and is trained on large text corpora with an extra text denoising autoencoder task. Under this multi-task framework, the shared encoder is capable of encoding both speech and text embedding from their corresponding encoder, mitigating the mismatch between the embedding from the shared encoder and text decoder. [6] examins different synthetic input generation schemes for text data and concludes that repeating phoneme input by their relative duration to each other achieves the best performance. Rather than introducing another encoder for text data, [10] utilizes traditional acoustic and language modeling techniques in hybrid systems. An acoustic-to-phoneme module and a phoneme-to-word module are trained separately but decoded in an E2E fashion. [11, 12, 13] adopt text-to-speech (TTS) technique to utilize the extensive text corpora to generate labeled speech and improve the generalization ability of ASR models without a modification to the model structure.

Following previous work, we also leverage an extra text encoder network to learn contextual representations from a large text corpus in this paper. While the framework is similar to the multi-modality framework used in [5], we propose an embedding aligner to reduce the mismatch between speech and text embeddings, which was not considered in these earlier studies. Under this refined framework, the text encoder trained with MLM target becomes a simplified phoneme bert [14], injecting phoneme information into the embedding aligner. The speech encoder trained with phoneme CTC target learns from the aligner and encodes speech into embeddings that are easier to align with those from the text decoder. Meanwhile, dot product is replaced with Euclidean distance to calculate the pairwise distance between input embedding and embedding aligner for making the embeddings from speech and text further closer. [6] only took advantage of the extensive text data and the text encoder to mimic speech input for data augmentation. [7] performed experiments under a low-resource setup and adopted a different target to enforce the speech-language alignment. Inspired by the code-switch methods in machine translation [15, 16] for better alignment, we extend it to a modality switch training (MST) method to swap speech and text embedding for enhancing speech-text alignment in ASR training, based on the forced alignment result. We conduct experiments on LIBRISPEECH ASR tasks and SNIPS slot filling (SF) task of spoken language understanding (SLU). The experimental results from both tasks show our proposed methods are promising.

Our contributions can be summarized as: 1) We propose an embedding aligner method to explicitly align the speech and text hidden spaces by sharing the same weights optimized by both the MLM and CTC losses; 2) Euclidean distance is employed to make the speech and language spaces tied closer; 3) A modality switch training (MST) method is proposed to fuse the embeddings generated from speech and text encoders to enhance the alignment between them; 4) The effectiveness of our proposed methods has been demonstrated on both ASR and SLU tasks, i.e., significantly reduce the WER of ASR on LIBRISPEECH and improve the F1 score of SF

*Work done during an internship at Microsoft.

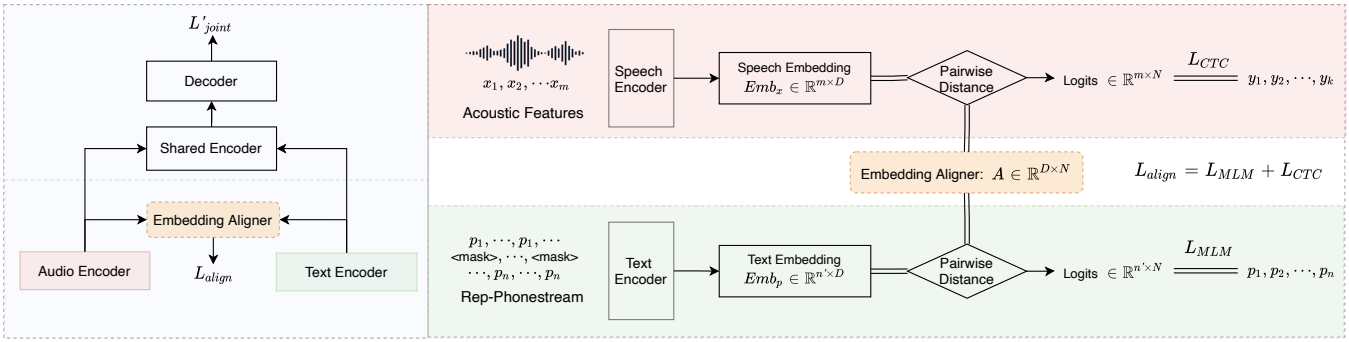

Fig. 1: The proposed framework to enhance the speech-text embedding alignment. The left part shows the overall structure of the framework. We adopt joint CTC-attention multi-task loss in the decoder side and attach our proposed alignment loss on the encoder side. The right part shows the details of the encoders and the embedding aligner. We calculate the pairwise distance between embeddings from two encoders and the embedding aligner. The output logits are used for MLM and CTC target computation. Two pairwise distance metrics are compared: (1) Dot product (2) Euclidean distance.

on SNIPs.

2. METHODOLOGY

The whole network structure is illustrated in the left part of Fig. 1. For the speech input, an audio encoder is used to extract speech embeddings, and the shared encoder is used to learn context-aware linguistic features, followed by a decoder to generate the ASR output. For the text data, the masked text input is processed with the text encoder, the shared encoder and the shared decoder to recover the original text data. To deal with the mismatch problem between the audio encoder and the text encoder, in this section, we introduce two approaches, the embedding aligner, and the modality switch training, to explicitly align the hidden spaces of speech and text.

2.1. Embedding Aligner

As shown in the right part of Fig.1, we trained the text encoder with MLM target and the speech encoder with CTC target on a text corpus. To extract linguistic information from phoneme sequences and to model acoustic information with phoneme tokens, both MLM and CTC targets adopt phoneme representations of text sequences.

The embedding aligner is a linear projection shared between MLM and CTC targets, which is used for calculating the logits in the softmax layer. Denote the dimension of speech and text embedding as D and the size of the phoneme dictionary as N. The embedding aligner is denoted as A ($A \in \mathbb{R}^{D \times N}$), representing the learnable embedding for each entrance in the phoneme dictionary. To deal with the length mismatch between speech and text, we model phoneme relative duration by Rep-Phonestream [6], and repeat phonemes by their relative duration to each other. Denote $P = (p_1, p_2, \cdots, p_n)$ as the phoneme representation of a text sequence with n phoneme tokens, where 20% of the phonemes are randomly replaced with <mask> symbols. After repeating phonemes, we have $P_{rep} = (p_1, \cdots, p_1, p_2, \cdots, p_2, p_n, \cdots, p_n)$, whose length is denoted as n'. The phoneme MLM target can be calculated as:

$$Emb_p = \text{Text-Encoder}(P_{rep}) \tag{1}$$

$$\hat{P} = \text{Softmax}(Emb_p \cdot A) \tag{2}$$

$$L_{MLM} = \text{MLMLoss}(\hat{P}, P) \tag{3}$$

where Emb_p is the embedding of P extracted by text encoder. Denote $X = (x_1, x_2, \cdots, x_m)$ as the acoustic features of a speech sample with m frames. Its corresponding phoneme transcripts is denoted as $Y = (y_1, y_2, \cdots, y_k)$. The phoneme CTC target can be calculated as:

$$Emb_x = \text{Speech-Encoder}(X) \tag{4}$$

$$\hat{Y} = \text{Softmax}(Emb_x \cdot A) \tag{5}$$

$$L_{CTC} = \text{CTCLoss}(\hat{Y}, Y) \tag{6}$$

where Emb_x is the embedding of X extracted by speech encoder. Note that in Eq.(2) and Eq.(5), the logits in the softmax layer is calculated as the pairwise dot product distance between the input embedding and each entrance of the embedding aligner. We replace the dot product distance with euclidean distance to cast a stronger restriction on the pairwise distance between input embedding and embedding aligner in both direction and scale. Concretely, Eq.(2) and Eq.(5) are rewritten as:

$$\hat{P} = \text{Softmax}(\text{Pairwise-Euclidean}(E_p, A)) \tag{7}$$

$$\hat{Y} = \text{Softmax}(\text{Pairwise-Euclidean}(E_x, A)) \tag{8}$$

It should be noted that, $\hat{P}$ and $\hat{Y}$ share the same weight A, and the two objectives L_{MLM} and L_{CTC} try to push both the hidden states E_p and E_x to A. The loss of embedding aligner becomes:

$$L_{align} = L_{MLM} + L_{CTC} \tag{9}$$

Both dot product and Euclidean distance can measure the similarity between two vectors. These two similarity measurements are equivalent if the vectors are unit-length, but the dot product is proportional to the vector length. We conducted experiments with dot product and length normalization for the embeddings E_p and E_x, but cannot get any promising improvement.

For the decoder part, we adopt a joint CTC attention training framework in [4] and subwords as modeling units for text [17] denoted as L'_{joint}. Note that this multitask target is applied for both paired speech data and unpaired text data. Combining encoder and decoder, the loss under our proposed framework is:

$$L = \alpha L_{align} + (1 - \alpha) L'_{joint} \tag{10}$$

where α is the weight of the embedding alignment loss.

2.2. Modality Switch Training

Aligning the embedding space of input and output sequence by randomly replacing input tokens with their corresponding output tokens has been proved effective in multi-lingual machine translation [15, 16]. The success of this approach in machine translation can be ascribed to two reasons: (1) The input and output spaces are discrete and share the same token list, which reduces the mismatch between the embeddings of source and target languages. (2) Most input and output tokens correspond one by one. Thus the replacement won't make a big difference in sequence length. Regarding the ASR tasks, for the first one, the mismatch between input and output spaces can be minimized by our proposed embedding aligner. For the second, given a pair of speech and phoneme sequences, we repeat each phoneme by its duration according to the force alignment results, and generate a new phoneme sequence whose length is exactly the same as speech frames. We compare two strategies regarding the swapping methods: phoneme-unaware and phoneme-aware, as illustrated in Fig.2:

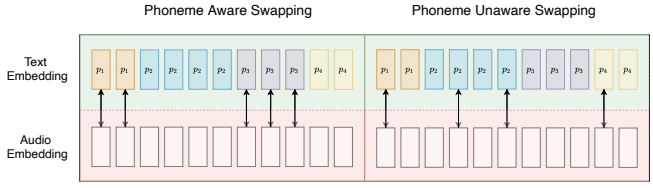

Fig. 2: Modality switch training with phoneme aware and phoneme unaware strategy: (1) phoneme-aware: embedding swapping is performed on spans of frames that correspond to the same phoneme token in forced alignment. (2) phoneme-unaware: random frames are selected to perform the embedding swapping regardless of their correspondence to phonemes. MST is only applied when training the model on speech data with transcription, where the transcripts are used as inputs for the text encoder. 10% of phonemes or frames are randomly selected for MST.

3. EXPERIMENTS

3.1. Experiment Setup

3.1.1. ASR task

ASR experiments are carried out on 960 hours of LIBRISPEECH dataset and a text corpus, which contains 14500 public domain books and comes with LIBRISPEECH for language model training. We adopt 10 layers of speech encoder, 4 layers of text encoder, 2 layers of shared encoder and 6 layers of the decoder with 2048 hidden units. Each layer is a Transformer block with 8 heads of 64 dimension self-attention layer [18]. The weight α in Eq.(10) is set to 0.2. The weight for CTC and attention is set to 0.3 and 0.7. We use an 80-dimensional log Mel-filterbank with 25ms window length computed every 10ms as inputs of speech encoder. Spec-augment with policy LD [19] is applied during training. The model is trained on both unpaired text data and pair speech data alternately for each batch. The Adam [20] optimizer is adopted with 0.001 initial learning rate and 20,000 warmup steps. All models are trained until convergence.

For the text encoder, the input is position-dependent phonemes generated with Rep-Phonestream strategy for the text corpus. The force alignment results obtained from the TDNN chain model in Kaldi [21, 22] are used as inputs to text encoder for transcripts of LIBRISPEECH audios. And the relative duration of phoneme estimated from the force alignment results for speech data are used to repeat phonemes in unpaired text data. We use the official LIBRISPEECH lexicon to perform grapheme-to-phoneme transduction (G2P). There are over 300 position-dependent phonemes in the phoneme dictionary. The number of modeling units (BPE) for text sequence in the decoder is 10,000 [17]. Experiments are carried out with ESPnets toolkit [23].

3.1.2. SLU task

SLU experiments are carried out on SNIPS dataset and perform a slot filling (SF) task. We employ the same data split as in Audio SNIPS corpus, which contains synthesized multi-speaker utterances for SNIPS and was used for SF task in SUPERB Benchmark [24]. The SF task requires a model to extract slot type and slot value pairs directly from speech inputs. The SF task here is reformulated as a sequence-to-sequence problem by predicting a sequence in which the slot value is surrounded by slot type boundaries. For example, a slot value and slot type pair "served_dish : maple syrup" is converted to "B-served_dish maple syrup E-served_dish". We also follow SUPERB to use Character Error Rate (CER) for slot value and F1 score for slot type to evaluate the performance of SF task. We adopt Transformer with 12 layers of encoder and 2 layers of decoders for SLU task. The CTC weight in MTL is set to zero in this task.

3.2. Experiment Results and analysis

The ASR performance for different system setups is shown in Table 1. It also lists the performance of reference systems from [5], which leveraged text data in a multi-modality transformer framework. We reproduced the results in [5] and used them as our baseline systems. Note that the joint-CTC-attention framework was not adopted in [5] since the text outputs modeled with subword units were not necessarily shorter than the text inputs modeled with phoneme units. We repeated the phoneme inputs to similar lengths as speech inputs, enabling the application of an extra CTC loss on the decoder side. The corresponding WERs are shown in the second row and fifth row. We got the same findings as those reported in [5], i.e., adding a text encoder trained with the phoneme inputs is beneficial to ASR performance and the number of shared layers between audio encoder and text encoder plays a marginal impact on WER.

3.2.1. Embedding aligner

The effectiveness of our proposed embedding aligner is shown in the bottom part of Table 1. Embedding aligner can improve the ASR performance by relative 1.3% to 11.4% WER reductions on different evaluation sets. In addition, replacing the dot product with Euclidean distance as the similarity measurement among the embeddings from both encoders and the embedding aligner can further improve the performance by a relative 3.6% WER reduction, averaged over four evaluation sets. We observe that the relative improvement on clean sets is larger than on other sets. A possible explanation is that the matched condition between training and testing on the clean set makes the speech encoder relatively easy to generate embeddings closer to its correct entrance of the embedding aligner.

3.2.2. Modality switch training

We compared the phoneme-aware and phoneme-unaware strategies of MST. The results are shown in the last two rows of Table 1. Phoneme-unaware MST did not improve the performance. One possible reason is that among a span of frames that corresponds to the same phoneme, only a few frames were replaced by the text embeddings. Thus the encoder was still able to utilize speech embedding from other frames to perform the phoneme recognition. Under such circumstances, phone-unaware MST has similar effects as the time-warp strategy in spec-augment, which has already been employed in

Multi-Modal	Num of Shared Layers in Encoder	Embedding Aligner (Distance Metrics)	Modality Switch Training	Dev		Test	
				clean	other	clean	other
✗ [5]	N/A	N/A	N/A	3.5	8.1	3.7	8.1
✗	N/A	N/A	N/A	3.3	8.0	3.6	8.0
✓ [5]	6	N/A	N/A	3.0	7.4	3.3	7.6
✓ [5]	0	N/A	N/A	3.0	7.4	-	-
✓	2	N/A	N/A	3.1	7.6	3.5	7.3
✓	2	Dot Product	N/A	2.9	7.2	3.1	7.2
✓	2	Euclidean	N/A	2.7	7.1	3.0	7.0
✓	4	Euclidean	N/A	2.7	7.2	2.9	7.0
✓	2	Euclidean	Phoneme Unaware	2.8	7.1	3.1	6.9
✓	2	Euclidean	Phoneme Aware	**2.7**	**6.9**	**2.9**	**6.8**

Table 1: Performance comparison (WER%) of different setups

our system. With phoneme-aware MST, the encoder must learn to use text embeddings of the entire span of frames that corresponds to the selected phoneme to reconstruct the phoneme. The phoneme-aware MST is similar to the semantic mask proposed in [25] except that the masked embeddings are substituted with corresponding text embeddings. Therefore, the phoneme-aware MST can further improve the robustness of the embedding alignment and obtain a slight improvement on both dev-other and test-other sets.

3.2.3. Text data usage

Since in our system, text data for training LIBRISPEECH LM was used as multi-modality input, for a fair comparison, we further conducted experiments that trained an LM[1] on the same text data and applied shallow fusion with a single-modality ASR model. The results are shown in Table 2. It indicates that utilizing the text data to train an embedding aligner that optimizes the speech-text alignment on the encoder side yields better results than a shallow fusion with LM trained on the same text data on the decoder side. It also shows that our proposed method of using text data is complementary to LM shallow fusion, and the performance of the multi-modality trained model can be further improved by LM rescoring.

System	Dev		Test	
	clean	other	clean	other
Single-Modal	3.3	8.0	3.6	8.0
Single-Modal + LM	3.2	7.7	3.4	7.4
Multi-Modal	2.7	6.9	2.9	6.8
Multi-Modal + LM	2.6	6.6	2.9	6.6

Table 2: Comparison between LM and multi-modal training

3.2.4. Slot filling task

To validate that the encoder trained under our proposed framework is richer in text information. We set up experiments on the SF task of SLU, and the results are shown in Table.3. It shows the CERs of slot value and the F1s of slot type for the sequences generated by a transformer trained from scratch or the encoders of the transformer initialized by the parameters from a pretrained transformer on ASR task with single or multiple modalities. We use the encoder of a single modality transformer trained on LIBRISPEECH dataset for initialization, which is exactly the model for the approach named transformer from scratch in Table 3. A significant improvement can be observed by pretraining the transformer encoder on a larger speech corpus. Initialization with multi-modality ASR model can yield a consistent improvement over that with single modality on SF task in both valid and test sets. With decoder initialization, the performance can be further improved. SF performance we achieved is comparable with those reported in [24], which leveraged much larger size unlabeled data to get better-generalized representations for SLU.

Initialization	Valid		Test	
	CER	F1	CER	F1
Transformer from scratch	55.80	67.30	57.91	67.14
Transformer Single Init Enc	33.03	90.01	32.85	89.35
Transformer Multi Init Enc	27.91	92.83	26.03	91.80
+ Multi Init Dec	25.77	94.29	25.01	92.85

Table 3: Comparison between different initialization approaches on SNIPS slot filling task.

4. CONCLUSIONS

In this paper, we propose two approaches to optimize the alignment of the speech and language latent spaces under the multi-modality E2E ASR framework. We introduce a learnable embedding aligner, which is a shared linear projection between text encoder and speech encoder trained by MLM loss and CTC at phoneme level, respectively. The speech and text embeddings are expected to be pushed closer to the same latent space by the embedding aligner. We further enhance the speech-text embedding alignment with modality switch training, which randomly swaps speech and text embeddings based on the forced alignment results. Phoneme aware and phoneme unaware strategies are compared for MST. Finally, we validated that the semantic information is injected into the speech embeddings with our proposed approach by conducting experiments on SF task of SLU. Experiments show that our proposed method achieves a relative 14% to 19% WER reduction on LIBRISPEECH ASR task and an absolute 2.5% to 2.8% F1 score improvement on SNIPS SF task.

[1]We adopted the same LM configuration as in https://zenodo.org/record/3966501/

5. REFERENCES

[1] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[2] Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney, "Improved training of end-to-end attention models for speech recognition," *Interspeech 2018*, Sep 2018.

[3] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv:1506.07503*, 2015.

[4] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[5] Yun Tang, Juan Pino, Changhan Wang, Xutai Ma, and Dmitriy Genzel, "A general multi-task learning framework to leverage text data for speech to text tasks," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6209–6213.

[6] Adithya Renduchintala, Shuoyang Ding, Matthew Wiesner, and Shinji Watanabe, "Multi-modal data augmentation for end-to-end asr," *arXiv preprint arXiv:1803.10299*, 2018.

[7] Jennifer Drexler and James Glass, "Explicit alignment of text and speech encodings for attention-based end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 913–919.

[8] Yinghui Huang, Hong-Kwang Kuo, Samuel Thomas, Zvi Kons, Kartik Audhkhasi, Brian Kingsbury, Ron Hoory, and Michael Picheny, "Leveraging unpaired text data for training end-to-end speech-to-intent systems," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7984–7988.

[9] Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou, "Bridging the gap between pre-training and fine-tuning for end-to-end speech translation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 9161–9168.

[10] Zhehuai Chen, Qi Liu, Hao Li, and Kai Yu, "On modular training of neural acoustics-to-word model for lvcsr," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4754–4758.

[11] Zhehuai Chen, Yu Zhang, Andrew Rosenberg, Bhuvana Ramabhadran, Gary Wang, and Pedro Moreno, "Injecting text in self-supervised speech pretraining," *arXiv preprint arXiv:2108.12226*, 2021.

[12] Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu, "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7180–7184.

[13] Aleksandr Laptev, Roman Korostik, Aleksey Svischev, Andrei Andrusenko, Ivan Medennikov, and Sergey Rybin, "You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation," *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct 2020.

[14] Mukuntha Narayanan Sundararaman, Ayush Kumar, and Jithendra Vepa, "Phoneme-bert: Joint language modelling of phoneme sequence and asr transcript," *arXiv preprint arXiv:2102.00804*, 2021.

[15] Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju, "Code-switching pre-training for neural machine translation," *arXiv preprint arXiv:2009.08088*, 2020.

[16] Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li, "Pre-training multilingual neural machine translation by leveraging alignment information," *arXiv preprint arXiv:2010.03142*, 2020.

[17] Taku Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," *arXiv preprint arXiv:1804.10959*, 2018.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[19] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Interspeech 2019*, Sep 2019.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Vesel, "The kaldi speech recognition toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 01 2011.

[22] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi.," in *Interspeech*, 2016, pp. 2751–2755.

[23] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[24] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al., "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[25] Chengyi Wang, Yu Wu, Yujiao Du, Jinyu Li, Shujie Liu, Liang Lu, Shuo Ren, Guoli Ye, Sheng Zhao, and Ming Zhou, "Semantic mask for transformer based end-to-end speech recognition," *arXiv preprint arXiv:1912.03010*, 2019.