INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Multitask Training with Text Data for End-to-End Speech Recognition

Peidong Wang[1,2*], *Tara N. Sainath*[1], *Ron J. Weiss*[1]

[1]Google, New York, NY, USA
[2]The Ohio State University, Columbus, OH, USA
wang.7642@osu.edu, {tsainath, ronw}@google.com

Abstract

We propose a multitask training method for attention-based end-to-end speech recognition models. We regularize the decoder in a listen, attend, and spell model by multitask training it on both audio-text and text-only data. Trained on the 100-hour subset of LibriSpeech, the proposed method, without requiring an additional language model, leads to an 11% relative performance improvement over the baseline and approaches the performance of language model shallow fusion on the test-clean evaluation set. We observe a similar trend on the whole 960-hour LibriSpeech training set. Analyses of different types of errors and sample output sentences demonstrate that the proposed method can incorporate language level information, suggesting its effectiveness in real-world applications.

Index Terms: multitask, text-only, attention, end-to-end

1. Introduction

Attention-based end-to-end (E2E) speech recognition systems map audio features directly to text-level representations [1, 2, 3, 4, 5]. Various model architectures [6, 7, 8, 9, 10] and training schemes [11, 12, 13] were proposed. The models are typically trained on transcribed speech datasets comprised of parallel audio-text pairs. Such audio-text pairs are more difficult and expensive to obtain compared with audio or text data in isolation. Recently, substantial performance improvements have been made by leveraging audio-only data for speech recognition [14, 15, 16].

The most common method for leveraging text-only data is to train a language model (LM) and integrate it into the recognition process using shallow [17], cold [18], or deep fusion [19]. These methods utilize a second neural network model and thus require additional space and computational resources, making them difficult to deploy in resource-constrained environments such as on-device ASR systems.

Another way to use text-only samples is to convert them to audio-text pairs using text-to-speech synthesis (TTS) techniques. Inspired by the back-translation method in neural machine translation [20], Li *et al.* proposed to train the ASR model using audio-text pairs generated from text-only data [21]. Multiple methods were proposed to jointly train the ASR and TTS models in a cycle-consistent manner [22, 23, 24]. Wang *et al.* used a loss term to encourage the ASR model to generate consistent outputs on real and synthesized presentations of the same utterance [25]. These methods face the problem that synthesized audio may bias the ASR model towards unrealistic speech.

As an alternative to LM fusion and TTS, knowledge distillation methods were proposed to transfer the knowledge in an LM to the ASR model [26, 27]. An LM is first trained using a large amount of text-only data. To train the ASR model, LM model outputs on the transcripts of the audio-text data are used as soft labels. This approach uses a pre-trained LM during ASR training and does not explicitly incorporate the large amount of text-only data into the ASR model.

The joint acoustic and text decoder (JATD) model [28] incorporates the text transcribed by a conventional ASR model in order to work in two modes, ASR and language modeling. It is used in a hybrid network designed for two-pass recognition, a transducer for streaming recognition followed by a non-causal rescoring pass using an attention-based decoder [29]. In this study, we propose multitask training with text-only data (MUTE), which differs from JATD in multiple aspects. First, we incorporate reference text directly during training, without using any corresponding audio data and external ASR model. Second, MUTE is designed for one-pass recognition and uses a single attention-based decoder pass during inference. Compared with JATD, a more closely related area of MUTE is the subtraction of internal LMs for end-to-end ASR models [30, 31]. MUTE can be viewed as a method to regularize the internal LM so that it does not overfit smaller training datasets of audio-text pairs. Experimental results on the 100-hour subset of LibriSpeech show that MUTE can effectively incorporate text-only data into E2E models, outperforming the baselines trained using audio-text pairs alone and approaching the performance of LM shallow fusion. We observe a similar trend using the whole 960-hour LibriSpeech training set.

The remainder of this paper is organized as follows. We describe MUTE in Section 2. In Section 3 and 4, we present the experimental setup and evaluation results, respectively. Concluding remarks are given in Section 5.

2. System Description

2.1. Attention-Based End-to-End Speech Recognition

Let us denote the input feature to an attention based end-to-end ASR model as $\boldsymbol{x} \in \mathbb{R}^{T \times F}$ and the corresponding output token sequence as $\boldsymbol{y} \in \mathbb{R}^{U}$, where F is the input feature dimension, and T and U denote the lengths of the input and output, respectively. For audio-text training samples, we denote the output as $\boldsymbol{y}^a \in \mathbb{R}^{U^a}$, where a refers to audio-text. A typical E2E system models the following distribution at output token step u:

$$p(y_u^a | \boldsymbol{x}, \boldsymbol{y}_{1:u-1}^a; \theta) \tag{1}$$

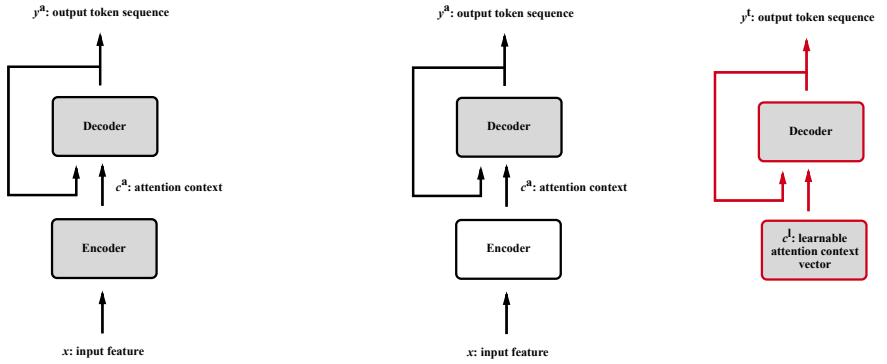
where θ denotes the model parameters.

2.2. Multitask Training with Text Data

Fig. 1 shows the two stages of MUTE training. In the first stage, the whole model is trained using the available audio-text pairs. In the second stage, we retrain the decoder by alternating training steps on audio-text pairs and text-only examples. With

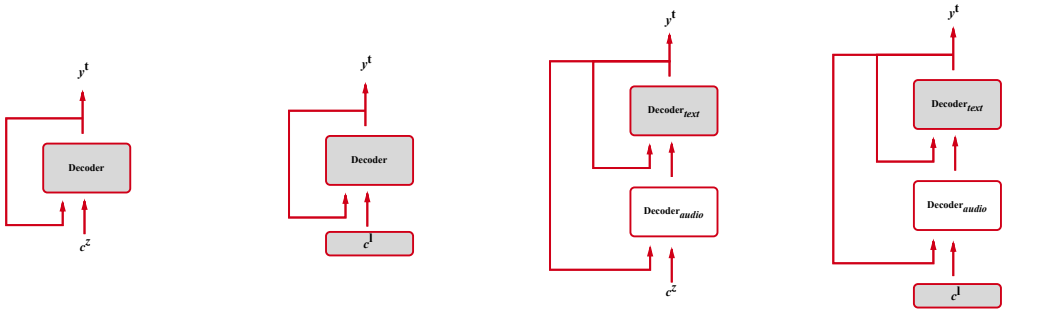
*This work was performed during an internship at Google.

http://dx.doi.org/10.21437/Interspeech.2021-683



(a) *Stage 1* (b) *Stage 2: audio-text* (c) *Stage 2: text-only*

Figure 1: *Illustration of the two stages of MUTE training. Components that are updated during training are in gray, and those which are fixed are white. Black arrows and boxes are used to denote audio-text pairs, while red denotes text-only data.*



(a) *MUTE-Z* (b) *MUTE-L* (c) *MUTE-ZT* (d) *MUTE-LT*

Figure 2: *Illustration for the four variations of MUTE decoders at training stage 2 using text-only data. See Fig. 1 caption for the meaning of different colors.*

audio-text pairs as training data, expression (1) can be simplified to:

$$p(y_u^a | c_u^a, \boldsymbol{y}_{1:u-1}^a; \theta_d) \tag{2}$$

where $c_u^a \in \mathbb{R}^{U^a \times E}$ refers to the context vector extracted from the encoder embedding for step u. E denotes the dimension of the context vector and θ_d denotes the parameters in the decoder. For text-only examples, we denote the output token sequence as $y^t \in \mathbb{R}^{U^t}$:

$$p(y_u^t | c^t, \boldsymbol{y}_{1:u-1}^t; \theta_d) \tag{3}$$

where c^t corresponds to a specialized context vector, used for all output steps, indicating to the decoder layers that there is no audio context for this example and it should instead rely only on its internal state and the auto-regressive input to predict the next token.

Steps taken on text-only data help expose the model to a larger corpus of text data during training. Since the decoder parameters are mostly shared with the ASR task, this next-token prediction LM task prevents the decoder from over-fitting the small amount of audio-text pairs. Meanwhile, steps using audio-text data ensures that the decoder can perform ASR tasks when conditioned on audio features. By sweeping over the mixing ratios of audio-text and text-only data, MUTE takes the best from both ASR and LM tasks.

2.3. Model Variants

We design the model architecture of MUTE from two perspectives, the choice of context vector c^t in equation (3) and the feedback loop for text-only data.

The context vector c^t in equation (3) can be of two types, constant or learnable. We set the values of the constant context vector to zeros. The learnable context vector is shared for different frames and its weights are learned using the ASR objective function. The zero and learnable context vectors are denoted as c^z and c^l, respectively. The corresponding models are referred to as MUTE-Z and MUTE-L, and are shown as Fig. 2-(a) and Fig. 2-(b). In MUTE-Z and MUTE-L, the feedback loop is shared for audio-text pairs and text-only data. For MUTE-ZT and MUTE-LT, we use an additional feedback loop specifically for text-only data. For audio-text training samples, MUTE-ZT and MUTE-LT update the entire decoder, whereas for text-only data, the two models only update the text-only loop. The two architectures are depicted in Fig. 2-(c) and Fig. 2-(d), respectively. Note that the text-only loop is inside the audio-text loop so that the whole decoder can take audio-text pairs during inference.

3. Experimental Setup

3.1. Data and Model

We conduct experiments on the LibriSpeech corpus. The training data for MUTE is a mixture of audio-text pairs and text-only data. For most experiments, the audio-text pairs are from the train-clean-100 subset and the text-only data is from the LibriSpeech-LM corpus, which contains about 40 million sentences. We also conduct experiment on the whole 960-hour Lib-

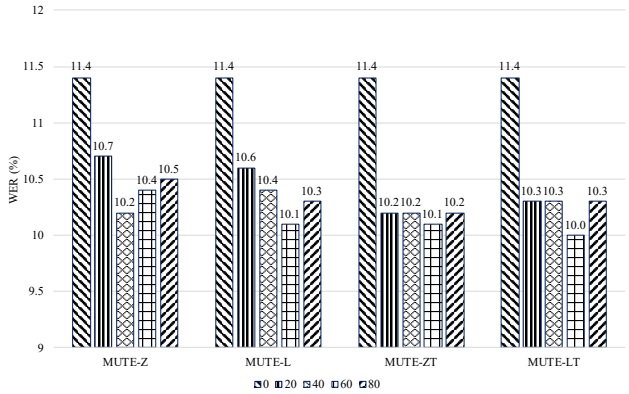


Figure 3: *WERs of MUTE variations on LibriSpeech test-clean. Each bar corresponds to a different text-only data mixing ratio.*

riSpeech training set. The experiments on LibriSpeech 100h may better simulate a realistic low resource condition, where the amount of text-only data is much larger than that of transcribed audio-text pairs. We vary the mixing ratio of text-only data in the whole training set from 0% (i.e. the baseline) to 80%, with a stride of 20%. We use the standard evaluation sets in the LibriSpeech corpus.

The E2E models in our experiments are 8-layer listen, attend, and spell (LAS) models based on the large model from [32]. The encoder consists of 2 batch-normalized convolutional layers, followed by 4 bidirectional long short-term memory (LSTM) layers. The decoder has 4 unidirectional LSTM layers. All LSTM layers contain 1024 nodes. For MUTE-ZT and MUTE-LT, we use the top two layers in the decoder as Decoder$_{text}$ and the remaining two layers as Decoder$_{audio}$. We also use a language model trained on the LibriSpeech-LM corpus for the comparison between MUTE and shallow fusion. The LM is comprised of 2 LSTM layers, each consisting of 2048 nodes. Note that we use LAS models for their simplicity in incorporating text-only data. We do not perform data augmentation to exclude the influence of distorted audio features to our analysis. Note that JATD cannot work in our experimental setup since it requires unlabelled audio features and an external ASR model. We thus do not report the results of JATD.

3.2. Implementation Details

In the first training stage, we train the entire LAS model using audio-text pairs. In the second stage, we fix MUTE encoder parameters to those found during the first stage and randomly re-initialize the decoder. In initial experiments we found similar performance when initializing the decoder parameters using the first stage parameters. It is important to note that the encoder batch normalization layer statistics need to be fixed to ensure convergence. The decoders are trained using a mixture of two types of data, audio-text and text-only. We randomly pick one type of sample at each training step with the mixing ratio described in Section 3.1. Within each batch all samples are of the same type. To minimize the influence of hyperparameter selection, we use exponential moving average with a constant learning rate of 10^{-3} to train all the baseline and MUTE models. All models are trained for extensive number of epochs. The best models on the validation set are used for evaluation.

Table 1: *WER comparisons among MUTEs.*

Model	dev-clean	dev-other	test-clean	test-other
MUTE-Z	9.8	29.6	10.2	31.1
MUTE-L	9.7	29.4	10.1	30.4
MUTE-ZT	9.6	29.4	10.1	31.4
MUTE-LT	9.6	29.5	10.0	30.9

4. Evaluation Results

4.1. Model Variant Selection

We first compare and select one variant of MUTE for further experiments. For each of the model variants, we need to compare the results on different mixing ratios of the text-only and audio-text data. Fig. 3 shows the test-clean WERs of the four types of MUTEs with various mixing ratios. MUTE-Z performs the best with 40% text-only data, and all other MUTEs with 60%.

Table 1 shows the WER comparisons among the MUTEs with their optimal training data mixing ratios. Although MUTE-LT performs the best on test-clean, MUTE-L is comparable to MUTE-LT on test-clean, performs better on test-other, and requires less modification to the decoder architecture. We thus experiment with MUTE-L for the remainder of this paper. Note that the models with the text-only feedback loop perform worse on test-other than those without it. The reason may be that Decoder$_{text}$ is not directly exposed to the audio features during training.

4.2. WER and Oracle WER Comparisons

The top of Table 2 compares baseline and MUTE models trained on LibriSpeech 100h in terms of WER and oracle WER (the minimum WER in the n-best decoding list). For WER, MUTE outperforms the baseline by 11% relatively on test-clean. When both models are shallow fused with the external LM, the relative improvement is 4%. Note that our comparison is mainly on test-clean since we use the clean data (i.e. LibriSpeech 100h) during training. For oracle WER, MUTE without shallow fusion performs better than the baseline with shallow fusion, indicating that the best hypothesis in the beam is improved with the text-only data.

The bottom of Table 2 compares models trained on the whole LibriSpeech 960h training set. For WER, MUTE achieves the same relative improvement 11% over the baseline

Table 2: *WER and oracle WER comparisons between the baseline and MUTE trained using LibriSpeech 100h and 960h. The models decoded using LM shallow fusion are denoted as those with* + LM.

Model	Train Set	WER				Oracle WER			
		dev-clean	dev-other	test-clean	test-other	dev-clean	dev-other	test-clean	test-other
Baseline	100h	10.8	30.8	11.4	32.2	7.7	26.6	8.1	28.0
MUTE	100h	9.7	29.4	10.1	30.4	6.8	25.0	7.3	26.2
Baseline + LM	100h	8.8	27.6	9.7	28.8	7.0	24.7	7.7	25.8
MUTE + LM	100h	8.6	27.4	9.3	28.4	6.5	23.7	7.1	25.1
Baseline	960h	4.5	13.6	4.7	13.7	2.4	9.7	2.6	9.5
MUTE	960h	4.1	11.9	4.2	12.1	2.1	8.2	2.2	8.0
Baseline + LM	960h	3.3	10.3	3.6	10.3	2.0	8.1	2.4	7.8
MUTE + LM	960h	3.4	10.3	3.6	10.3	1.9	7.7	1.9	7.1

on test-clean and a relative improvement of 12% on test-other. The similar improvement to LibriSpeech 100h dataset demonstrates that MUTE is still helpful in a large training corpus setting. For oracle WER, MUTE alone outperforms the baseline using LM shallow fusion on test-clean, which is also consistent with the results on LibriSpeech 100h.

4.3. Error Analysis

We analyze MUTE by comparing different types of errors and sample output sentences using the models trained on LibriSpeech 100h.

Table 3: *Comparisons of different types of errors for models trained on LibriSpeech 100h. The results are shown in the order of deletion/insertion/substitution. See Table 2 caption for acronyms.*

Model	dev-clean	dev-other	test-clean	test-other
Baseline	1.0/1.4/8.4	3.2/3.5/24.1	1.2/1.7/8.6	3.3/3.7/25.2
MUTE	1.3/1.1/7.2	3.9/2.9/22.6	1.4/1.3/7.4	3.9/3.0/23.5
Baseline + LM	1.6/1.0/6.2	4.9/3.1/19.6	2.1/1.1/6.5	5.3/3.3/20.2
MUTE + LM	1.5/1.0/6.2	4.5/2.7/20.2	1.9/1.0/6.4	4.5/2.8/21.2

Table 3 shows the deletion, insertion, and substitution errors of different methods. The results of MUTE and LM shallow fusion have the same trend: deletion errors increase, insertion errors decrease, and substitution errors decrease. This indicates that MUTE has a similar impact to LM shallow fusion on the error distribution. In Table 4, we analyze the ability of MUTE to incorporate language level information by comparing four pairs of sample output sentences on test-clean. Since the goal of this analysis is to understand the sorts of errors that are not made when using MUTE training, the samples are selected such that MUTE contains less errors than the baseline. In the first pair of sentences, the baseline generates "get" and MUTE outputs "yet". These two words are similar in pronunciation but "a chance yet to" is better grammatically. In the second pair, "is lashed to" generated by MUTE uses the correct tense. In the following two output sentence pairs, we show the results of a single utterance using different methods. The "ounce" generated by MUTE fits better into the context "milligram" and "an". In addition to the comparison between MUTE and the baseline, we can observe from the table that baseline + LM may contain more errors than the baseline alone. This indicates that with the language information limited to audio-text pairs, the baseline ASR model could generate incorrect outputs that mislead LM shallow fusion.

Table 4: *Sample output sentences on test-clean. In the first row of each pairs of samples, wrong words are highlighted in red, whereas in the second row, the corresponding correct words are highlighted in green.*

Model	Sentence
Baseline	i haven't had a chance get to tell you what a jolly little place i think this is
MUTE	i haven't had a chance yet to tell you what a jolly little place i think this is
Baseline + LM	each of us is lash to some part of the raft
MUTE + LM	each of us is lashed to some part of the raft
Baseline	milligram roughly one twenty eight thousand of an house
MUTE	milligram roughly one twenty eight thousand of an ounce
Baseline + LM	madame roughly one twenty eight thousand of an house
MUTE + LM	milligram roughly one twenty eight thousand of an ounce

5. Concluding Remarks

We have proposed MUTE, a two-stage multitask training approach for attention-based end-to-end speech recognition models to incorporate language level information. Text-only data is used to regularize the training of the decoder in a multitask manner. Trained using LibriSpeech 100h as audio-text data, MUTE outperforms the baseline by 11% relatively on the test-clean evaluation set. It approaches the performance of shallow fusion and does not need the additional LM. We observe a similar trend on the LibriSpeech 960h training set. Analyses of different types of errors and sample output sentences show that MUTE can incorporate language level information effectively. Future work includes designing test-time training/adaptation methods for MUTE, combining MUTE with audio-only techniques, expanding MUTE to transducer based models, and applying MUTE to deliberation tasks.

6. References

[1] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 577–585.

[2] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 193–199.

[3] C. C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[4] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[5] R. Prabhavalkar, Y. He, D. Rybach, S. Campbell, A. Narayanan, T. Strohman, and T. N. Sainath, "Less is more: Improved rnn-t decoding using limited label context and path merging," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5659–5663.

[6] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[7] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[8] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "Contextnet: Improving convolutional neural networks for automatic speech recognition with global context," in *Proc. of INTERSPEECH*, 2020, pp. 3610–3614.

[9] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. of INTERSPEECH*, 2020, pp. 5036–5040.

[10] B. Li, A. Gulati, J. Yu, T. N. Sainath, C.-C. Chiu, A. Narayanan, S.-Y. Chang, R. Pang, Y. He, J. Qin *et al.*, "A better and faster end-to-end model for streaming ASR," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5634–5638.

[11] J. Cui, C. Weng, G. Wang, J. Wang, P. Wang, C. Yu, D. Su, and D. Yu, "Improving attention-based end-to-end ASR systems with sequence-based loss functions," in *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 353–360.

[12] P. Wang, J. Cui, C. Weng, and D. Yu, "Token-wise training for attention based end-to-end speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6276–6280.

[13] ——, "Large margin training for attention based end-to-end speech recognition." in *Proc. of INTERSPEECH*, 2019, pp. 246–250.

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

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

[16] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[17] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Proc. of INTERSPEECH*, 2017, pp. 523–527.

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

[19] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1–5828.

[20] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proc. of Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 86–96.

[21] B. Li, T. N. Sainath, R. Pang, and Z. Wu, "Semi-supervised training for end-to-end models via weak distillation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2837–2841.

[22] T. Hori, R. Astudillo, T. Hayashi, Y. Zhang, S. Watanabe, and J. Le Roux, "Cycle-consistency training for end-to-end speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6271–6275.

[23] 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. of INTERSPEECH*, 2019, pp. 3790–3794.

[24] Y. Ren, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Y. Liu, "Almost unsupervised text to speech and automatic speech recognition," in *Proc. of International Conference on Machine Learning (ICML)*, 2019, pp. 5410–5419.

[25] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, Y. Wu, and P. Moreno, "Improving speech recognition using consistent predictions on synthesized speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7029–7033.

[26] Y. Bai, J. Yi, J. Tao, Z. Tian, and Z. Wen, "Learn spelling from teachers: Transferring knowledge from language models to sequence-to-sequence speech recognition," in *Proc. of INTERSPEECH*, 2019, pp. 3795–3799.

[27] H. Futami, H. Inaguma, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Distilling the knowledge of bert for sequence-to-sequence asr," in *Proc. of INTERSPEECH*, 2020, pp. 3635–3639.

[28] T. N. Sainath, R. Pang, R. J. Weiss, Y. He, C. C. Chiu, and T. Strohman, "An attention-based joint acoustic and text on-device end-to-end model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7039–7043.

[29] T. N. Sainath, R. Pang, D. Rybach, Y. He, R. Prabhavalkar, W. Li, M. Visontai, Q. Liang, T. Strohman, Y. Wu *et al.*, "Two-pass end-to-end speech recognition," in *Proc. of INTERSPEECH*, 2019, pp. 2773–2777.

[30] E. Variani, D. Rybach, C. Allauzen, and M. Riley, "Hybrid autoregressive transducer (HAT)," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6139–6143.

[31] E. McDermott, H. Sak, and E. Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 434–441.

[32] K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen, "On the choice of modeling unit for sequence-to-sequence speech recognition," in *Proc. of INTERSPEECH*, 2019, pp. 3800–3804.