INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Semi-supervised Sequence-to-sequence ASR using Unpaired Speech and Text

Murali Karthick Baskar[ϕ,†], *Shinji Watanabe*[†], *Ramon Astudillo*[π], *Takaaki Hori*[ϒ],
Lukáš Burget[ϕ], *Jan Černocký*[ϕ]

[ϕ]Brno University of Technology, Czech Republic
[†]Johns Hopkins University, USA
[π]IBM T. J. Watson Research Center, USA
[ϒ]MERL, USA

baskar@fit.vutbr.cz

Abstract

Sequence-to-sequence automatic speech recognition (ASR) models require large quantities of data to attain high performance. For this reason, there has been a recent surge in interest for unsupervised and semi-supervised training in such models. This work builds upon recent results showing notable improvements in semi-supervised training using cycle-consistency and related techniques. Such techniques derive training procedures and losses able to leverage unpaired speech and/or text data by combining ASR with Text-to-Speech (TTS) models. In particular, this work proposes a new semi-supervised loss combining an end-to-end differentiable ASR→TTS loss with TTS→ASR loss. The method is able to leverage both unpaired speech and text data to outperform recently proposed related techniques in terms of %WER. We provide extensive results analyzing the impact of data quantity and speech and text modalities and show consistent gains across WSJ and Librispeech corpora. Our code is provided in ESPnet to reproduce the experiments.

Index Terms: Sequence-to-sequence, end-to-end, ASR, TTS, semi-supervised, unsupervised, cycle consistency

1. Introduction

Sequence-to-sequence (seq2seq) ASR training directly maps a speech input to an output character sequence using a neural network [1–3], similar to those used in machine translation [4, 5]. The model requires a considerable amount of paired speech and text to learn alignment and classification [6, 7], which limits its use in under-resourced domains. On the other hand, unpaired speech and text can be obtained for most domains in large quantities making training with unpaired data particularly relevant for seq2seq models.

Recent works have shown that the problem of seq2seq training in low-resource conditions can be tackled using unpaired data. These methods can be classified into three categories according to the type of data used. First, methods utilizing only unpaired speech for unsupervised training. In this category, [8] proposes an end-to-end differentiable loss integrating ASR and TTS models by the straight-through estimator. The work in [9] also proposes an end-to-end differentiable loss integrating ASR and a Text-to-Encoder (TTE) model. Both works, showed that connecting ASR with TTS/TTE can handle unpaired speech data as well as reduce ASR recognition errors. The second category concerns methods leveraging unpaired text data, which focus on enhancing the decoder component of seq2seq ASR [10] or moving the encoder representation closer to text representation [11]. The former approach [10] is implemented by feeding the text data to TTE-to-ASR [10] model, where the TTE converts the text directly to encoder representations and then is fed to the decoder. In the latter model, an encoder component is shared to have common representation across both speech and text data [11]. In addition to these works, [12] used a language model (LM) built with unpaired text-only data to jointly train with an ASR. In the third category, both unpaired speech and text data are exploited. Examples of this are the speech-chain framework [13] and adversarial training schemes [14, 15]. The former proposes a dual pipeline of ASR→TTS and TTS→ASR systems. Here, the backpropagation only takes place in the secondary stage of the chain. The latter utilizes, adversarial learning objectives to match the speech and text distributions.

Outside of the speech processing area, cycle-consistency can be related to successful progress in machine translation [16], which derive end-to-end differentiability by using N-best approach. Also, improvements are observed in image processing tasks using cycle-consistency with an adversarial objective [17].

In this paper, we borrow ideas from the above mentioned techniques to handle unpaired data using cycle-consistency. The basic idea of this approach is that if a model converts an input data to output data and another model reconstructs the input data from the output data, then the input data and its reconstruction should be similar. We use this idea to derive a new loss able to use both unpaired speech and text data. The contributions of this paper are the following

- A speech-only end-to-end differentiable loss is proposed using ASR→TTS. This improves the TTE cycle loss in [9] and is related to the straight-through loss in [8]
- We complement this with a text-only pipeline loss using TTS→ASR. This improves TTE backtranslation [10] and is related to the speech-chain in [13]
- We combine this and a shallow integrated RNNLM [18, 19] to obtain a high performance on unpaired data.

Our experimental analysis using WSJ and Librispeech corpus described in section 4 substantiate our hypothesis that unpaired speech and text can individually improve ASR performance and combining them provides additional gains.

2. Cycle consistency training

2.1. ASR and TTS with seq2seq models

Sequence-to-sequence ASR systems [2] model directly a probability distribution over C-length character or phoneme

http://dx.doi.org/10.21437/Interspeech.2019-3167

sequence $\mathbf{Y} = \{y_c\}_{c=1}^{C}$ given T-length speech features $\mathbf{X} = \{x_t\}_{t=1}^{T}$ i.e. Mel-filterbank as

$$p_{\text{ASR}}(\mathbf{Y} \mid \mathbf{X}) = \prod_{c}^{C} p(y_c \mid y_{1:c-1}, \mathbf{X}). \quad (1)$$

Each character prediction depends on the entire input and all previously generated characters. To account for this complex relation, a neural network with attention mechanism is used [4]. This is composed of an encoder network, typically one or more bi-directional LSTMs layers [20], generating an encoder sequence $\mathbf{H} = \{h_t\}_{t=1}^{T}$ as

$$\mathbf{H} = \text{encoder}(\mathbf{X}). \quad (2)$$

A decoder, also modeled by one or more LSTM networks, utilizes a matching function to select the relevant encoded input features$\{h_t\}_{t=t_c}^{t'_c}$ for each step c given its internal state. This matching function is normalized to resemble a probability distribution over the input features and is termed attention. Decoder internal state, attended input and previously produced character are used to predict the current character.

When paired data is available, the seq2seq models are trained with the cross-entropy (CE) criterion given correct output y^*:

$$\begin{aligned}\mathcal{L}_{\text{ASR}} &= -\log p_{\text{ASR}}(y^* \mid X) \quad (3)\\ &= -\sum_{l=1} \log p_{\text{ASR}}(y_l^* \mid y_{1:l-1}^*, X).\end{aligned}$$

TTS seq2seq models use a very similar architecture, but receive the characters as input $\mathbf{Y} = \{y_c\}_{c=1}^{C}$ and predict the speech features with a regression layer $\mathbf{X} = \{x_t\}_{t=1}^{T}$. There are some needed modifications with respect to ASR, such as the need to predict continuous data and the end of utterance. In this work, we used the Tacotron2 architecture, detailed in [21]. The loss of the TTS system is divided into three terms:

$$\mathcal{L}_{\text{TTS}} = \mathcal{L}_{\text{MSE}} + \mathcal{L}_{L1} + \mathcal{L}_{BCE} \quad (4)$$

where the mean square error (squared L2) and L1 norms are over a regression estimate $\hat{\mathbf{X}}$ and BCE is the binary cross entropy loss for the end of sentence prediction. MSE and L1 components of the loss can be interpreted as the minus log-probability of the speech features for Gaussian and Laplace distributions for constant scale parameters i.e.

$$\begin{aligned}\mathcal{L}_{\text{TTS}} &= -\log p_{\text{TTS}}(\mathbf{X}^* \mid \mathbf{Y}) \quad (5)\\ &= -\sum_{t=1} \log p_{\text{TTS}}(\mathbf{X}_t^* \mid \mathbf{X}_{1:t-1}^*, \mathbf{Y})\end{aligned}$$

considering the MSE only, we have

$$-\log p_{\text{TTS}}(\mathbf{X}_t^* \mid \mathbf{X}_{1:t-1}^*, \mathbf{Y}) \propto \|\mathbf{X}_t^* - \mathbf{g}(\mathbf{Y}, \mathbf{X}_{1:t-1}^*)\|^2 \quad (6)$$

where $\hat{\mathbf{X}}_t = \mathbf{g}(\mathbf{Y}, \mathbf{X}_{1:t-1}^*)$ is the Tacotron2 auto-regressive estimate at time t.

2.2. Unsupervised training with unpaired speech data

The previous formulations concern the case in which both paired speech $\mathbf{X}$ and text $\mathbf{Y}$ are available. In this section we propose an improvement over recent work in [9] that provides an unsupervised loss for ASR while being more performant. A central problem to the ASR→TTS pipeline is the fact that the text bottleneck eliminates a lot of information from speech e.g. speaker identity. The work in [9] solves this by training the TTS model to predict encoded speech, $\mathbf{H}$ in eq. (2), instead of speech $\mathbf{X}$ directly. Since the encoded speech keeps some speech characteristics, it is possible to define the following loss based on the Cycle-Consistency (CC) criterion

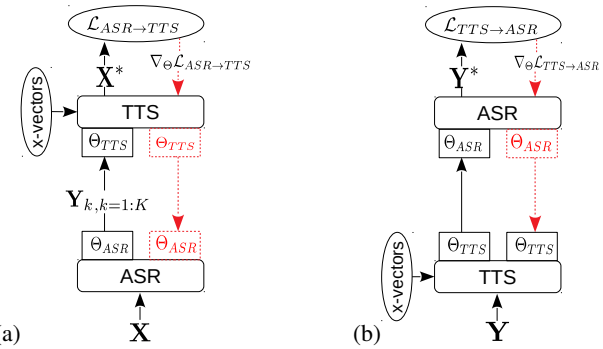

Figure 1: *Simplified representation of the cycle-consistency training for unpaired speech and text. The flow of the diagram is bottom-to-top. The red dotted lines denotes the gradient propagation and the updated parameters. Figure 1a denotes the ASR→TTS pipeline where $\mathbf{Y}_k$ is the k^{th} randomly sampled ASR output. Figure 1b shows the TTS→ASR pipeline.*

$$\mathcal{L}_{\text{ASR}\to\text{TTE}} = E_{p_{\text{ASR}}(\mathbf{Y}|\mathbf{X})}\{\mathcal{L}_{\text{TTE}}\} \quad (7)$$

where $\mathcal{L}_{\text{TTE}}$ is the same as $\mathcal{L}_{\text{TTS}}$ but replacing $\mathbf{X}$ by the encoded speech $\mathbf{H}$. This loss penalizes the ASR system for transcriptions that, once transformed back into an estimate of the encoded speech $\hat{\mathbf{H}}$ by the TTE, differ from the original encoded speech $\mathbf{H}$. Such loss does not require to have the correct text output y^* and is end-to-end differentiable.

TTE-CC has the disadvantage of having to train an specific network to predict $\mathbf{H}$. The encoded speech, may also already eliminate some of the speech characteristics making the CC loss less powerful. In this work, we propose to use the TTS loss instead of the TTE loss, to realize

$$\mathcal{L}_{\text{ASR}\to\text{TTS}} = E_{p_{\text{ASR}}(\mathbf{Y}|\mathbf{X})}\{\mathcal{L}_{\text{TTS}}\}. \quad (8)$$

Aside from being computationally more intensive, this requires solving the problem of passing speaker characteristics through the text bottleneck, see Figure 1(a). In order to do so, inspired by [13], we augment the TTS model with speaker vectors obtained from a x-vector network [22]. For example, for the MSE component of $\mathcal{L}_{\text{TTS}}$ we have

$$\mathcal{L}_{\text{MSE}} = -\log p_{\text{TTS}}(\mathbf{X}^* \mid \mathbf{Y}, \mathbf{f}(\mathbf{X})) \quad (9)$$

where $\mathbf{f}$ implements the x-vector. Note that x-vectors are designed to retain speaker characteristics but not general structure of the speech signal. In that sense, the model can not learn to copy directly $\mathbf{X}$ from input to output.

Similarly to [9] we use policy-gradient to back-propagate through the expectation in the loss and update the ASR parameters. This yields the following gradient update formula

$$\nabla_\theta \mathcal{L}_{\text{ASR}\to\text{TTS}} \approx \sum_{\mathbf{Y}^n \sim p_{\text{ASR}}} \frac{1}{N} R(\mathbf{Y}^n, \mathbf{X}) \nabla_\theta \log p_{\text{ASR}}(\mathbf{Y}^n \mid \mathbf{X}) \quad (10)$$

for N samples. The reward R is obtained by subtracting a bias term from the TTS loss to reduce variance $R(\mathbf{Y}^n, \mathbf{X}) = \mathcal{L}_{\text{TTS}} - B(\mathbf{X})$. The bias term is calculated as the mean value of $\mathbf{X}$ for each sample. In practical terms, using policy gradient amounts to sampling multiple sentences from the ASR distribution and backpropagating each of them as if they were the ground truth, but weighted by the reconstruction loss.

2.3. Unsupervised training with unpaired text data

As with speech, unsupervised training of ASR using only text and cycle-consistency is possible by using a TTS→ASR chain, see Figure 1(b). Since our target in this work is to increase ASR performance, the resulting computation graph is simpler than in the ASR→TTS case. There is no need to backpropagate into the TTS module and thus the chain can be realized by forming a non-differentiable TTS→ASR pipeline as

$$\mathcal{L}_{\text{TTS}\rightarrow\text{ASR}} = -\log p_{\text{ASR}}(\mathbf{Y}^* \mid \hat{\mathbf{X}}). \tag{11}$$

The point estimate is obtained by providing the TTS system with the input text $\mathbf{Y}$ along with randomly chosen x-vectors to generate

$$\hat{\mathbf{X}} = \arg\max_{\mathbf{X}} \{p_{\text{TTS}}(\mathbf{X} \mid \mathbf{Y})\}. \tag{12}$$

Thus the pipeline can act as an autoencoder and allows training of unpaired text only data. Using a non-differentiable pipeline where TTS generates synthetic speech sentences for which we only have a transcription is akin to back-translation in machine translation [23]. The work in [10] proposes a similar idea utilizing TTE instead of TTS. The TTS→ASR pipeline is also used in [13].

2.4. Unsupervised training with both unpaired speech and text data

In the case in which both paired speech and text are available, both ASR→TTS and TTS→ASR can be trained jointly, see figure 1. The final loss $\mathcal{L}_{\text{both}}$ is a linear interpolation of the loss functions defined in equations (10) and (8).

$$\mathcal{L}_{\text{both}} = \alpha * \mathcal{L}_{\text{ASR}\rightarrow\text{TTS}} + (1-\alpha) * \mathcal{L}_{\text{TTS}\rightarrow\text{ASR}} \tag{13}$$

where α is a hyper-parameter set by default to $\alpha = 0.5$.

3. Experimental Setup

An extensive experimental setup leveraging the Librispeech [24] and Wall Street Journal (WSJ) [25] corpora was used in this work. For each corpus, a sub-set of the data was used to simulate unpaired speech and text conditions.

Training utilized 83-dimensional filter-bank with pitch features as input. The encoder-decoder network utilized location aware attention [2]. For WSJ and Librispeech experiments, the encoder comprises 8 bi-directional LSTM layers [20, 26] each with 320 units and the decoder comprises 1 (uni-directional) LSTM layer with 320 units. The CE objective is optimized using AdaDelta [27] with an initial learning rate set to 1.0. The training batch size is 30 and the number of training epochs is 20. The learning rate decay is based on the validation performance computed using the character error rate (min. edit distance). ESPnet [28] is used to implement and execute all our experiments. For TTS, Tacotron2 [21] model is used and is configured as described in our previous work [9] differing only with the output targets. Here, we use 83 dimensional log-Mel filterbank features as targets and x-vectors [22] are fed as auxiliary information to provide speaker characteristics of each utterance. Detailed experimental setup can be obtained from our github codebase in ESPnet. Unsupervised training was performed after conventional supervised training of each model. Cycle-consistency utilized five samples in Eq. 10. A small amount of paired data was also used to regularize the model during the unsupervised stage. For WSJ, 81 hours were split into 2, 5, 10 and 14 hours of paired data and the remaining 67 hours was used as unpaired data. Table 1 shows the effect of the amount of parallel data on WSJ and Librispeech using conventional supervised training regime. The eval92 test set was kept for evaluations. In Librispeech, the 100 hours is used as paired data and the 360 hours set as unpaired data as in [9].

Table 1: *Baseline performance of paired data (semi-supervised) across WSJ-SI84 and Librispeech by using pre-trained and randomly initialized ASR model on eval-92 test for WSJ and test-clean for Librispeech*

Paired data Corpus info			
Name	# Hours	% CER	% WER
WSJ	2	27.7	68.2
	5	13.2	41.5
	10	10.8	33.7
	14	10.2	31.5
Librispeech	100	8.9	21.0

4. Analysis and Results

The initial experiments compared the performance of the TTS loss introduced in this work, to the TTE loss from previous work [9]. The analysis involved Librispeech setup using 100 hours of paired data with 180 and 360 hours of unpaired data for a fair comparison. The results shows that TTS approach improves %WER over TTE by a 2.5% relative (20.7% to 20.1%) on 360 hours of unpaired data and 2.9% relative (19.9% to 19.4%) on 180 hours set.

4.1. Impact of amount of paired data (semi-supervision)

In this section, the model is tested with varying amounts of unpaired data such as 14 hours and 67 hours along with certain amount of semi-supervision such as 2, 5, 10 and 14 hours of data. Table 2, shows the %WER obtained by using varying amounts of unpaired speech and text data and different amounts of paired data. The experiments are done starting from 2 hours of parallel data and 14 hours of unpaired data. With only 2 hours of parallel data, the model %WER performance improves from 68.2 in Table 1 to 49.8 by only aiding 14 hours of unpaired speech and improved to 51.9 with 67 hours of unpaired speech.

Overfitting is observed in the case of unpaired speech, while in case of text only data, the model improved consistently from 63 (pair 14 hours) to 39.6 (pair 67 hours). This showcases the importance of text only data under very low-resource scenario. Interestingly, including both unpaired speech and text of 14 hours, the %WER improved to 43.7 and with 67 hours of data the model obtained 41.4 %WER. The pattern emerging here is that, under very low-resource scenario, the model benefits from large amounts of text. But, adding more speech only data leads to slight degradation in performance. This pattern is not observed with 5, 10 and 14 hours of parallel data conditions as

increasing the amount of speech data from 14 hours to 67 hours improved by ∼1% absolute WER. With 14 hours of supervision and 67 hours of unpaired speech, text and both, the %WER improves to 28.0, 27.1 and 26.2 respectively.

Table 2: *%WER performance analysis on varying the amount of paired data (semi-supervision) and unpaired data on eval-92 test set using WSJ. The Type refers to type unpaired data used and it can be either "speech/text/both"*

Unpaired data		Paired data (#hrs)			
#hrs	Type	2	5	10	14
14	Speech	49.8	39.9	29.8	-
14	Text	63.0	43.6	34.6	-
14	Both	43.7	35.5	28.3	-
67	Speech	51.9	38.8	28.4	28.0
67	Text	39.6	36.8	29.6	27.1
67	Both	41.4	34.2	27.7	26.2

Table 3: *Unsupervised ASR performance across best results in literature. The Type refers to type unpaired data used and it can be either "speech/text/both"*

WSJ-SI84 (parallel) + WSJ-SI284 (unpaired)				
Model	Type	RNNLM	%CER	%WER
Speech chain [13]	Both	-	9.9	-
Adversarial [14]	Both	yes	-	24.9
this work	Both	-	9.1	26.1
this work	Both	yes	**7.8**	**20.3**
oracle	-	-	5.5	16.4
oracle [29]	-	yes	2.0	4.8
Librispeech 100 h (parallel) + 360 h (unpaired)				
Backtranslation-TTE [10]	Text	-	10.0	22.0
this work	Text	-	8.0	17.9
Crictizing-LM [12]	Text	yes	9.1	17.3
this work	Text	yes	8.0	17.0
Cycle-TTE [9]	Speech	yes	9.9	19.5
this work	Speech	yes	7.8	16.8
Adversarial-AE [15]	Both	yes	8.4	18.0
this work	Both	-	7.6	17.5
this work	Both	yes	**7.6**	**16.6**
oracle [9]	-	-	4.6	11.8

4.2. Model comparison with literature

Table 3, shows the results of using both unpaired speech and text data from WSJ-SI284 and Librispeech 360 hours across literature. In WSJ corpus, the model achieves a %WER of 20.3 with RNNLM and 26.1 without RNNLM. This leaves a relative difference of 37.1% compared to oracle performance of 16.4%. The oracle result is our baseline performance using WSJ-SI284. Note that the performance on WSJ-SI284 is inferior to our previously reported baseline [29]. This is due to a difference in architecture necessary to fit ASR and TTS models into the GPU. In case of Librispeech, the table 3 shows that training with unpaired audio and text data can achieve a %WER of 17.5 leading to 32.5% relative difference with the oracle performance. Table 3 also shows that our approach only using unpaired text gains 18.0% relative improvement over backtranslation-TTE [10] approach. Complementary gains of absolute 0.9% were observed by integrating RNNLM with this approach and the result is compared with crictizing-LM [12]. The effectiveness of our approach using unpaired speech only data is shown in table 3 by obtaining 16.8% WER. Jointly training unpaired speech and text provided modest gains with a %WER improvement from 16.8 to 16.6. From these results, one can infer that training with unpaired speech only data has major benefits over text only data in large corpus such as Librispeech.

5. Related Work

As indicated in the previous sections, the work here proposed improvements over recent related approaches and integrated some of them into a single loss. Back-translation style TTE [10] synthesized encoder can be related to the TTS→ASR loss here used, as both utilize a pipeline that generates synthetic train data. The latter has the advantage of utilizing directly speech and has higher performance overall. The speech-chain framework [13] was the first work to integrate ASR and TTS to train using unpaired speech and text. It also utilizes a TTS→ASR pipeline loss but this is not learnt jointly with the end-to-end differentiable ASR→TTS loss. One limitation of the speech-chain is the fact that is not end-to-end differentiable, having to rely in the alternate training of a TTS→ASR and ASR→TTS loss to update both models. This limitation was addressed in [9] proposing an end-to-end differentiable loss for speech only and based on TTE. The work presented here improves upon this by extending TTE to TTS with x-vectors and combining this with a TTS→ASR point-estimate loss. Finally [8] is similar to [9] but applies straight-through estimators to construct end-to-end differentiable ASR→TTS losses based on argmax and a expected loss similar to Eq. 8.

6. Conclusions

This paper presented a new approach to exploit the information in unpaired speech and/or unpaired text to improve the performance of seq2seq ASR systems. We show that under low-resource conditions such as WSJ corpus the performance improvements are relatively higher compared to corpus with sufficient amount of data such as Librispeech. We show as well that integrating unpaired speech and text, both as a pipeline loss and through shallow integration with a RNN language model, provides additional gains and competitive results. Future work will be focused on cycle-consistency approaches where ASR and TTS do not have matching conditions. Preliminary experiments show that this is a limitation of current systems.

7. Acknowledgements

This work was supported by Czech Ministry of Education, Youth and Sports from the National Program of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602" and by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) MATERIAL program, via Air Force Research Laboratory (AFRL) contract # FA8650-17-C-9118 and U.S. DARPA LORELEI contract No. HR0011-15-C-0115. The views and conclusions contained herein are those of the authors and should not be interpreted as official policies, either expressed or implied, of ODNI, IARPA, AFRL or the U.S. Government. The work was also supported by Facebook (Research Award on Speech and Audio Technology for Voice Interaction and Video Understanding.

8. References

[1] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks." in *ICML*, vol. 14, 2014, pp. 1764–1772.

[2] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *ICASSP*, 2016, pp. 4945–4949.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016, pp. 4960–4964.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014, pp. 3104–3112.

[6] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *ICML*, 2016, pp. 173–182.

[7] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition." in *Interspeech*, 2017, pp. 939–943.

[8] A. Tjandra, S. Sakti, and S. Nakamura, "End-to-end feedback loss in speech chain framework via straight-through estimator," *arXiv preprint arXiv:1810.13107*, 2018.

[9] T. Hori, R. Astudillo, T. Hayashi, Y. Zhang, S. Watanabe, and J. L. Roux, "Cycle-consistency training for end-to-end speech recognition," in *ICASSP*, 2019.

[10] T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, and K. Takeda, "Back-translation-style data augmentation for end-to-end asr," in *SLT*, 2018, pp. 426–433.

[11] A. Renduchintala, S. Ding, M. Wiesner, and S. Watanabe, "Multi-modal data augmentation for end-to-end asr," in *Interspeech*, 2018, pp. 2394–2398.

[12] A. Liu, H.-y. Lee, and L.-s. Lee, "Adversarial training of end-to-end speech recognition using a criticizing language model," in *ICASSP*, 2019.

[13] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *ASRU*, 2017, pp. 301–308.

[14] J. Drexler and J. Glass, "Combining end-to-end and adversarial training for low-resource speech recognition," in *SLT*, 2018, pp. 361–368.

[15] S. Karita, A. Ogawa, M. Delcroix, and T. Nakatani, "Sequence training of encoder-decoder model using policy gradient for end-to-end speech recognition," in *ICASSP*, 2018, pp. 5839–5843.

[16] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma, "Dual learning for machine translation," in *NIPS*, 2016, pp. 820–828.

[17] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[18] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," in *Interspeech*, 2017, pp. 949–953.

[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 *ICASSP*, 2018, pp. 1–5828.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*, 2018, pp. 4779–4783.

[22] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP*, 2018, pp. 5329–5333.

[23] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.

[25] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proc. of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[26] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[27] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[28] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *Interspeech*, 2018, pp. 2207–2211.

[29] M. K. Baskar, L. Burget, S. Watanabe, M. Karafiát, T. Hori, and J. H. Černockỳ, "Promising accurate prefix boosting for sequence-to-sequence asr," *arXiv preprint arXiv:1811.02770*, 2018.