Interspeech 2022
18-22 September 2022, Incheon, Korea

Improved Consistency Training for Semi-Supervised Sequence-to-Sequence ASR via Speech Chain Reconstruction and Self-Transcribing

Heli Qi[1], *Sashi Novitasari*[1], *Sakriani Sakti*[2], *Satoshi Nakamura*[1]

[1]Nara Institute of Science and Technology, Japan
[2]Japan Advanced Institute of Science and Technology, Japan
{qi.heli.qi9, sashi.novitasari.si3, s-nakamura}@is.naist.jp, ssakti@jaist.ac.jp

Abstract

Consistency regularization has recently been applied to semi-supervised sequence-to-sequence (S2S) automatic speech recognition (ASR). This principle encourages an ASR model to output similar predictions for the same input speech with different perturbations. The existing paradigm of semi-supervised S2S ASR utilizes SpecAugment as data augmentation and requires a static teacher model to produce pseudo transcripts for untranscribed speech. However, this paradigm fails to take full advantage of consistency regularization. First, the masking operations of SpecAugment may damage the linguistic contents of the speech, thus influencing the quality of pseudo labels. Second, S2S ASR requires both input speech and prefix tokens to make the next prediction. The static prefix tokens made by the offline teacher model cannot match dynamic pseudo labels during consistency training. In this work, we propose an improved consistency training paradigm of semi-supervised S2S ASR. We utilize speech chain reconstruction as the weak augmentation to generate high-quality pseudo labels. Moreover, we demonstrate that dynamic pseudo transcripts produced by the student ASR model benefit the consistency training. Experiments on LJSpeech and LibriSpeech corpora show that compared to supervised baselines, our improved paradigm achieves a 12.2% CER improvement in the single-speaker setting and 38.6% in the multi-speaker setting.

Index Terms: semi-supervised learning, consistency regularization, FixMatch algorithm, speech chain reconstruction, ASR, TTS

1. Introduction

In recent years, sequence-to-sequence (S2S) ASR has made significant progress thanks to the advancement of deep neural networks. S2S ASR models are designed for directly converting the input speech into transcripts [1–3]. However, a large amount of transcribed speech data is essential for training S2S ASR models to achieve state-of-the-art performance. Thus, many semi-supervised learning algorithms have been proposed to efficiently train ASR models with the help of untranscribed speech [4–11].

Consistency regularization [12] is an important principle of semi-supervised learning algorithms. This principle was originally designed for semi-supervised image classification [13–16] and it has recently been extended to semi-supervised S2S ASR [4–7, 17]. Consistency regularization assumes that an ASR model should output similar predictions for the same input speech with various perturbations. Also, these perturbations should change the distribution of input speech without altering the corresponding transcripts [15]. In the literature, SpecAugment [18] is commonly used to perturb speech features due to its simplicity. Its time-frequency masking plays a major role in improving the robustness of ASR [17]. However, randomly removing continuous frequency bins or temporal frames may damage the semantics of the input speech. Incomplete linguistic contents will further accumulate errors in pseudo labels and thus influence ASR performance during consistency training.

Different from image classification, S2S ASR models require both input speech and prefix tokens to make the prediction at each time step. In the consistency training paradigm for S2S ASR [4–9], a teacher model trained on transcribed speech is used to produce pseudo transcripts for untranscribed speech. The pseudo transcripts are then fed into the student ASR model as prefix tokens during consistency training. However, there exist some errors in predictions of the teacher model because of the limited training set [7]. These errors remain in the static pseudo transcripts and make pseudo labels poorer in quality. Moreover, since the static pseudo transcripts are made by original speech before consistency training, the mismatch between these transcripts and perturbed speech will further influence the student model during consistency training.

This paper presents an improved consistency training paradigm of S2S ASR. In previous work, machine speech chain [19, 20] was designed to jointly train ASR and TTS by reconstructing unlabeled speech and text data. In this work, we adopt the speech chain reconstruction as a data augmentation method and focus on the FixMatch algorithm [15] which has recently been applied on S2S ASR [5]. Our contributions are as follow:

- We propose the self-transcribing scheme where the student model serves as its own teacher. Specifically, the pseudo transcripts are produced dynamically by the student model with perturbed speech as input.
- We demonstrate that speech chain reconstruction is superior than SpecAugment as the weak augmentation for making pseudo labels.
- We conducted constrast experiments that covers most of the factors that may have an impact on the consistency training, including number of speakers, confidence threshold, ratio of unlabeled data to labeled data, data augmentation methods, and pseudo transcript generation.

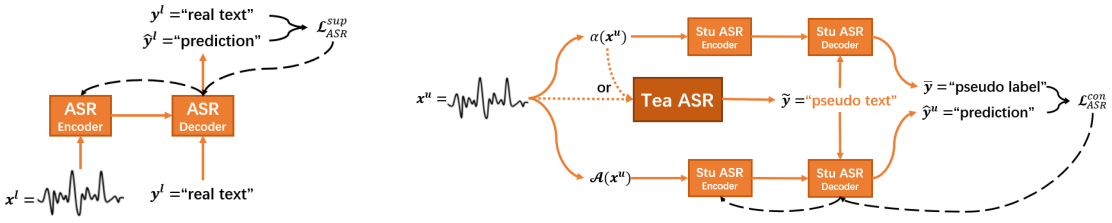
2. Semi-supervised consistency training for S2S ASR

2.1. Supervised training for a base ASR model

S2S ASR models are designed to directly predict the conditional probability $P(\hat{\boldsymbol{y}}|\boldsymbol{x};\theta_{ASR})$ of a sequence of predicted tokens $\hat{\boldsymbol{y}} = [\hat{y}_1, ..., \hat{y}_T]$ given a sequence of speech features $\boldsymbol{x} = [x_1, ..., x_S]$. Here, θ_{ASR} represents the parameters of

Copyright © 2022 ISCA

10.21437/Interspeech.2022-11169



(a) (b)

Figure 1: *"Tea" denotes the teacher model and "Stu" denotes the student model. $\alpha(\boldsymbol{x^u})$ denotes the weakly-perturbed speech and $\mathcal{A}(\boldsymbol{x^u})$ denote the strongly-perturbed speech. Black dash lines represent the back-propagation of the gradients. (a) Supervised training for sequence-to-sequence ASR. (b) FixMatch-based consistency training for sequence-to-sequence ASR. Orange dash lines represent the different input speech for making pseudo transcripts.*

ASR, S is the length of the input sequence, and T is the length of the output sequence. Our ASR models are based on Listen-Attend-Spell (LAS) [1] which has an encoder-decoder architecture as shown in Fig.1a.

The encoder converts the input sequence of speech features into a sequence of hidden representations $\boldsymbol{h}^e = [h_1^e, ..., h_S^e]$. The decoder receives $\boldsymbol{h}^e$ and outputs the token probability $p(\hat{y}_t|\hat{y}_{1:t-1}, \boldsymbol{x}; \theta_{ASR})$ at time step t based on the prefix tokens $\hat{y}_{1:t-1}$. The probability of the generated transcript $P(\hat{\boldsymbol{y}}|\boldsymbol{x}; \theta_{ASR})$ is calculated by the product of the token probability at each time step as

$$P(\hat{\boldsymbol{y}}|\boldsymbol{x}; \theta_{ASR}) = \prod_{t=1}^{T} p(\hat{y}_t|\hat{y}_{1:t-1}, \boldsymbol{x}; \theta_{ASR}). \quad (1)$$

During training, the prefix tokens $\hat{y}_{1:t-1}$ are replaced with the ground-truth labels. Given a speech-text pair $(\boldsymbol{x^l}, \boldsymbol{y^l})$, a base ASR model is trained by the following supervised ASR loss:

$$\mathcal{L}_{ASR}^{sup} = -\frac{1}{T}\sum_{t=1}^{T} \log p(y_t^l|y_{1:t-1}^l, \boldsymbol{x}; \theta_{ASR}). \quad (2)$$

2.2. Consistency training for S2S ASR based on FixMatch algorithm [5]

FixMatch algorithm [15] is a semi-supervised algorithm for image classification that combines consistency regularization with pseudo-labeling. In its existing application to S2S ASR [5], a teacher ASR model is initialized on the parameters of the base ASR model and used to produce the pseudo transcript $\tilde{\boldsymbol{y}}$ for the untranscribed speech $\boldsymbol{x^u}$ as

$$\tilde{\boldsymbol{y}} = \arg\max_{\boldsymbol{z}} P(\boldsymbol{z}|\boldsymbol{x^u}; \theta_{ASR}^{Tea}). \quad (3)$$

As shown in Fig.1b, untranscribed speech $\boldsymbol{x^u}$ is separately perturbed by a weak augmentation function $\alpha(\cdot)$ and a strong augmentation function $\mathcal{A}(\cdot)$. The weakly-perturbed speech $\alpha(\boldsymbol{x^u})$ is fed into the student ASR model to obtain the pseudo label $\bar{y}_t$ at time step t given $\tilde{y}_{1:t-1}$ as prefix tokens by

$$\bar{y}_t = \arg\max_{z} p(z|\tilde{y}_{1:t-1}, \alpha(\boldsymbol{x^u}); \theta_{ASR}^{Stu}). \quad (4)$$

The consistency ASR loss $\mathcal{L}_{ASR}^{con}$ is calculated on the strongly-perturbed speech $\mathcal{A}(\boldsymbol{x^u})$ and pseudo label $\bar{y}_t$ given the same pseudo transcripts $\tilde{y}_{1:t-1}$ as prefix tokens by

$$\mathcal{L}_{ASR}^{con} = -\frac{1}{T}\sum_{t=1}^{T} \mathbb{1}(q_t > \tau) \log p(\bar{y}_t|\tilde{y}_{1:t-1}, \mathcal{A}(\boldsymbol{x^u}); \theta_{ASR}^{Stu}), \quad (5)$$

where $q_t = p(\bar{y}_t|\tilde{y}_{1:t-1}, \alpha(\boldsymbol{x^u}); \theta_{ASR}^{Stu})$ is the confidence of the student model on $\bar{y}_t$ and τ is the confidence threshold. The value of τ restricts $\mathcal{L}_{ASR}^{con}$ to be calculated only by the pseudo labels that the student model is confident of. The larger τ is, the fewer pseudo labels $\bar{y}$ are used to calculate $\mathcal{L}_{ASR}^{con}$.

Finally, the student model is trained by the following loss function:

$$\mathcal{L}_{ASR} = \mathcal{L}_{ASR}^{sup} + \lambda_{con}\mathcal{L}_{ASR}^{con}. \quad (6)$$

2.3. Proposed self-transcribing scheme

On top of the existing paradigm, we made two improvements. First, the weakly-perturbed speech $\alpha(\boldsymbol{x^u})$ is used to produce the pseudo transcripts $\tilde{\boldsymbol{y}}$. Second, the student model is initialized by the base ASR and dynamically produces the pseudo transcripts by itself during the consistency training. Our improvements can be formualted as follow:

$$\tilde{\boldsymbol{y}} = \arg\max_{\boldsymbol{z}} P(\boldsymbol{z}|\alpha(\boldsymbol{x^u}); \theta_{ASR}^{Stu}). \quad (7)$$

3. Speech chain reconstruction

3.1. Semi-supervised TTS based on pseudo transcribing

Sequence-to-sequence TTS models can be considered the reverse case of ASR that directly predicts the conditional probability $P(\hat{\boldsymbol{x}}|\boldsymbol{y}, \theta_{TTS})$ of a sequence of speech features $\hat{\boldsymbol{x}} = [\hat{x}_1, ..., \hat{x}_S]$ given a sequence of tokens $\boldsymbol{y} = [y_1, ..., y_T]$. Our TTS model is based on Tacotron2 [21] which has a similar architecture to LAS. We provide the decoder of TTS with the speaker embedding $f(\boldsymbol{x})$ extracted from the input speech, which enables TTS to synthesize speech in a multi-speaker setting. Our TTS model is trained by the loss function $\mathcal{L}_{TTS}$:

$$\mathcal{L}_{TTS} = \mathcal{L}_{TTS}^{sup} + \mathcal{L}_{TTS}^{unsup}, \quad (8)$$

$$\mathcal{L}_{TTS}^{sup} = \frac{1}{S}\sum_{s=1}^{S}(x_s^l - \hat{x}_s^l)^2 - (b_s^l \log(\hat{b}_s^l) + (1 - b_s^l)\log(1 - \hat{b}_s^l)), \quad (9)$$

$$\mathcal{L}_{TTS}^{unsup} = \frac{1}{S}\sum_{s=1}^{S}(x_s^u - \hat{x}_s^u)^2 - (b_s^u \log(\hat{b}_s^u) + (1 - b_s^u)\log(1 - \hat{b}_s^u)), \quad (10)$$

where $\hat{x}_s, x_s$ are the predicted and ground-truth log Mel-scale spectrograms at time s and $\hat{b}_s, b_s$ are the predicted and ground-truth end-of-frame probabilities. The pseudo transcripts $\tilde{\boldsymbol{y}}$ used for training the TTS model is produced by the base ASR model before ASR consistency training as Eq.3 does.

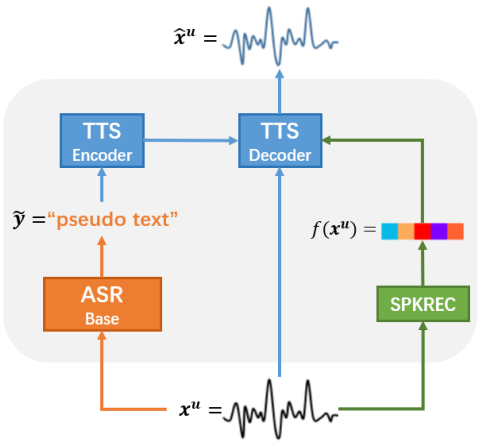


Figure 2: *Proposed speech chain reconstruction. The grey box can be viewed as the weak augmentation $\alpha(\cdot)$. SPKREC is short for speaker recognition model which is used to provide the TTS decoder with speaker embedding vectors in the multi-speaker experiments.*

3.2. Speech chain reconstruction for untranscribed speech

Different from our previous work that generates synthetic speech from real texts for semi-supervised ASR training [19, 20], this work applied the speech chain reconstruction as a data augmentation method for the untranscribed speech by the teacher-forcing technique. This technique prevents TTS decoding from mispronunciation and early stopping, thus protecting the linguistic contents. Fig.2 shows the process of our proposed speech chain reconstruction that can be considered as a frame-level speech transformation.

Untranscribed speech $\boldsymbol{x^u}$ is utilized three times before generating reconstructed speech $\hat{\boldsymbol{x}}^{\boldsymbol{u}}$. First, the base ASR model converts $\boldsymbol{x^u}$ into pseudo transcripts $\tilde{\boldsymbol{y}}$ as the input of our TTS encoder. Then, the speaker embedding $f(\boldsymbol{x^u})$ is extracted from $\boldsymbol{x^u}$ to provide our TTS decoder with the speaker information. Finally, our TTS decoder generates the reconstructed speech $\hat{\boldsymbol{x}}^{\boldsymbol{u}}$ with original speech $\boldsymbol{x^u}$ as the prefix speech features at each time step. The reconstructed speech $\hat{\boldsymbol{x}}^{\boldsymbol{u}}$ is treated as the weakly-perturbed speech $\alpha(\boldsymbol{x^u})$ during the subsequent ASR consistency training.

4. Experiments

4.1. Datasets

We conducted experiments in both single-speaker and multi-speaker settings. We use the LJSpeech corpus [22] for the single-speaker setting. We take 12,600 utterances as the training set, 250 utterances as the development set and 250 utterances as the test set. There is no overlap in the division of data. We treat the first 50% of the training set as labeled data and the last 50% as unlabeled data. In the multi-speaker setting, we use the LibriSpeech corpus [23]. We take the "train-clean-100" set as the training set, the "dev-clean" set as the development set and the "test-clean" set as the test set. We treat the first 75 speakers (8,570 utts) of the training set as labeled data and the last 176 speakers (19,968 utts) as unlabeled data.

As for acoustic features, we used 80-dimensional log Mel-spectrograms extracted at a 50-ms frame length and a 12.5-ms frame shift. The English letters in the transcripts of all utterances are normalized into their lowercase forms. All transcripts are mapped into a 29-token set: 26 (a-z) letters of the alphabet, apostrophes, space, and "*sos/eos*" [1].

4.2. Model configuration

4.2.1. ASR

In the single-speaker setting, our ASR encoder is composed of three bidirectional LSTM layers with 256 hidden units for each direction (totally 512 hidden units for each Bi-LSTM layer). Hierarchical sub-sampling [1,2] was used on the last two layers to reduce the sequence length by a factor of four. In the multi-speaker setting, we added two extra Bi-LSTM layers at the beginning of the encoder. On the decoder sides, the single-speaker and multi-speaker settings shared the same configuration: an embedding layer followed by a unidirectional LSTM layer with 512 hidden units. We selected Additive Attention [24] as the attention module of our ASR models. The beam searching technique was adopted to generate pseudo transcripts during the consistency training and the beam size was set to 4.

AdaDelta [25] was adopted to train our ASR models. During the base training, the initial learning rate was set to 1.0 and the decay rate was set to 0.1. During the consistency training, we set a smaller initial learning rate of 0.5 and a larger decay rate of 0.2. The learning rate decay was based on the accuracy calculated on the development set during training and the minimal learning rate was set to 1% of the initial value. We used early stopping to prevent the models from overfitting. The weight of the consistency loss $\mathcal{L}_{ASR}^{con}$ is set to 0.1.

4.2.2. TTS

The hyperparameters for our TTS model were generally the same as those for the original Tacotron2, except we concatenated the encoder hidden representations with the speaker embedding vectors in the multi-speaker experiments. We extracted X-vectors [26] from the input speech as the speaker embedding vectors and the extraction was done using SpeechBrain [27].

Adam [28] was adopted to train our TTS model with an initial learning rate of 0.001 and a decay rate of 0.1. The learning rate decay was based on the loss calculated on the development set during training and the minimal learning rate was set to 1% of the initial value. Early stopping was also used to avoid overfitting. For each time-step, our model generated two consecutive frames to reduce the number of steps in the decoding process.

4.3. Experimental setting

ESPNET2 [29, 30] was used to perform our experiments. Our supervised baselines were the base ASR models trained using only the labeled data. We designed four scenarios and conducted contrast experiments where the weak augmentation $\alpha(\cdot)$ was either SpecAugment or speech chain reconstruction. For all experiments, the strong augmentation $\mathcal{A}(\cdot)$ was implemented by SpecAugment. Our weak SpecAugment applied one time of time-frequency masking while the strong SpecAugment applied two times. Maximal frequency masking width was 5 bins for $\alpha(\cdot)$ and 20 bins for $\mathcal{A}(\cdot)$. For the LJSpeech corpus, maximal time masking width was 10 frames for $\alpha(\cdot)$ and 50 frames for $\mathcal{A}(\cdot)$. For LibriSpeech corpus, maximal time masking width was 20 frames for $\alpha(\cdot)$ and 100 frames for $\mathcal{A}(\cdot)$.

[1] We combined *sos* and *eos* into one token.

Table 1: *CER results of our constrast experiments. We designed 4 scenarios in 4 different ways of generating pseudo transcripts $\tilde{y}$. In each scenario, we set different weak augmentations $\alpha(\cdot)$ and confidence thresholds τ for comparison. The strong augmentation $\mathcal{A}(\cdot)$ is fixed to strong SpecAugment for all experiments. Underlined numbers denote the best performance in each scenario.*

	LJSpeech					LibriSpeech		
$\alpha(\cdot)$	τ=0.5	τ=0.6	τ=0.7	τ=0.8	τ=0.9	τ=0.5	τ=0.7	τ=0.9
Supervised Baseline								
–	8.2	8.2	8.2	8.2	8.2	28.0	28.0	28.0
Static $\tilde{\boldsymbol{y}}$ produced by $\boldsymbol{x^u}$ (the existing paradigm [5])								
Weak SpecAugment	8.3	7.8	7.7	7.5	7.7	18.3	19.6	20.8
Speech Chain Reconstruction	7.9	7.6	<u>7.4</u>	7.5	7.8	<u>18.2</u>	19.8	18.5
Static $\tilde{\boldsymbol{y}}$ produced by $\alpha(\boldsymbol{x^u})$								
Weak SpecAugment	7.8	7.7	7.7	7.8	7.6	18.8	19.6	20.3
Speech Chain Reconstruction	7.9	7.7	<u>7.2</u>	<u>7.2</u>	7.6	<u>17.2</u>	18.4	18.3
Dynamic $\tilde{\boldsymbol{y}}$ produced by $\boldsymbol{x^u}$								
Weak SpecAugment	7.9	7.9	7.6	7.4	7.6	19.1	19.1	19.9
Speech Chain Reconstruction	8.2	<u>7.2</u>	7.5	7.4	7.6	<u>18.1</u>	18.4	18.4
Dynamic $\tilde{\boldsymbol{y}}$ produced by $\alpha(\boldsymbol{x^u})$								
Weak SpecAugment	7.5	7.4	8.1	7.6	8.0	19.8	20.5	18.9
Speech Chain Reconstruction	7.7	7.7	7.6	7.4	<u>7.2</u>	20.0	19.1	<u>18.5</u>

5. Results and analysis

5.1. Single-speaker experiments

In the single-speaker setting, the ratio of labeled data to unlabeled data was set to 1:1. From Tab.1, we observed that speech chain reconstruction outperforms the weak SpecAugment in all scenarios. Our consistency training paradigm achieved the best CER performance of 7.2%, which has a 12.2% CER improvement on the supervised baseline. This showcases how speech chain reconstruction keeps more linguistic information than SpecAugment, and thus it is more suitable to be the weak augmentation.

In the single-speaker setting, dynamic pseudo transcripts produced by the student model achieved a 2.7% CER improvement on the existing paradigm. This indicates that the student model gradually corrects the errors in the pretrained base model during consistency training. It can be seen that the pseudo transcripts produced by the weakly-perturbed input speech $\alpha(\boldsymbol{x^u})$ also resulted in a 2.7% CER improvement. It supports our idea that prefix tokens should match the input speech during ASR consistency training. As for the reason why the relative improvements on the existing paradigm were not large in scale, we hypothesized that the base model is already good enough to produce understandable transcripts because 50% of the training set was used as the labeled data.

5.2. Multi-speaker experiments

In the multi-speaker setting, we simulated a harsher condition where the ratio of labeled data to unlabeled data is 3:7. From Tab.1, speech chain reconstruction still outperformed the weak SpecAugment in all scenarios. With more unlabeled data, the improvement of our paradigm over the supervised baseline surged to 38.6%, which indicates that our paradigm benefits from a large amount of untranscribed speech.

On top of the existing paradigm, a 5.5% CER improvement was achieved when the input speech used to produce the pseudo transcripts was changed from the original speech $\boldsymbol{x^u}$ to weak-perturbed speech $\alpha(\boldsymbol{x^u})$. On the other hand, only a 0.5% CER improvement was observed when we set the teacher model in the existing paradigm to the student model itself during consistency training. This indicates that the mismatch between the input speech and pseudo transcripts has a stronger influence on our ASR models than the errors in the pretrained base model.

According to the right part of Tab.1, our ASR models performed better when we set the confidence threshold to a smaller value. Since we only used the first 30% of "train-clean-100" as the labeled data to train the base model, our student ASR models are not very confident on the untranscribed speech and thus output relatively lower token probability at each time step. With a higher confidence threshold, only a small fraction of untranscribed speech is utilized to calculate the final ASR loss, hence seriously restricting the potential of consistency training for improving ASR performance.

6. Conclusions

In this work, we proposed an improved consistency training paradigm for S2S ASR. We took the FixMatch algorithm as the proving ground and presented comprehensive constrast experiments covering most of the factors in the semi-supervised ASR training. Our results show that speech chain reconstruction protects more linguistic contents than SpecAugment and produces pseudo labels with higher quality. Moreover, the proposed self-transcribing method helps the student model correct the errors in the pretrained base model and eliminate the mismatch between the input speech and prefix tokens. Our future work involves applications of other semi-supervised algorithms on S2S ASR with various data augmentation methods for speech data.

7. Acknowledgements

Part of this work is supported by JSPS KAKENHI Grant Number JP21H05054 and JP21H03467.

8. References

[1] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[2] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[3] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Advances in neural information processing systems*, vol. 28, 2015.

[4] Z.-q. Zhang, Y. Song, J.-s. Zhang, I. V. McLoughlin, and L.-r. Dai, "Semi-supervised end-to-end asr via teacher-student learning with conditional posterior distribution." in *INTERSPEECH*, 2020, pp. 3580–3584.

[5] F. Weninger, F. Mana, R. Gemello, J. Andrés-Ferrer, and P. Zhan, "Semi-supervised learning with data augmentation for end-to-end asr," in *INTERSPEECH*, 2020.

[6] R. Masumura, M. Ihori, A. Takashima, T. Moriya, A. Ando, and Y. Shinohara, "Sequence-level consistency training for semi-supervised end-to-end automatic speech recognition," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7054–7058.

[7] Z. Chen, A. Rosenberg, Y. Zhang, H. Zen, M. Ghodsi, Y. Huang, J. Emond, G. Wang, B. Ramabhadran, and P. J. M. Mengibar, "Semi-supervision in asr: Sequential mixmatch and factorized tts-based augmentation," in *INTERSPEECH*, 2021.

[8] J. Kahn, A. Lee, and A. Hannun, "Self-training for end-to-end speech recognition," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7084–7088.

[9] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," in *INTERSPEECH*, 2020.

[10] Y. Higuchi, N. Moritz, J. L. Roux, and T. Hori, "Momentum pseudo-labeling for semi-supervised speech recognition," in *INTERSPEECH*, 2021.

[11] A. Xiao, C. Fuegen, and A. Mohamed, "Contrastive semi-supervised learning for asr," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3870–3874.

[12] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," *Advances in neural information processing systems*, vol. 27, 2014.

[13] L. Samuli and A. Timo, "Temporal ensembling for semi-supervised learning," in *International Conference on Learning Representations (ICLR)*, vol. 4, no. 5, 2017, p. 6.

[14] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," *Advances in neural information processing systems*, vol. 29, 2016.

[15] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems*, vol. 33, pp. 596–608, 2020.

[16] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[17] 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 *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7029–7033.

[18] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *INTERSPEECH*, 2019, pp. 2613–2617.

[19] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 301–308.

[20] ——, "Machine speech chain," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 976–989, 2020.

[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 *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[22] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[24] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations (ICLR)*, 2015.

[25] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[26] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[27] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] 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. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[30] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "Espnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2020, pp. 7654–7658.