End-to-End Rich Transcription-Style Automatic Speech Recognition with Semi-Supervised Learning

Tomohiro Tanaka, Ryo Masumura, Mana Ihori,
Akihiko Takashima, Shota Orihashi, Naoki Makishima

NTT Media Intelligence Laboratories, NTT Corporation, Japan

tomohiro.tanaka.ht@hco.ntt.co.jp

Abstract

We propose a semi-supervised learning method for building end-to-end rich transcription-style automatic speech recognition (RT-ASR) systems from small-scale rich transcription-style and large-scale common transcription-style datasets. In spontaneous speech tasks, various speech phenomena such as fillers, word fragments, laughter and coughs, etc. are often included. While common transcriptions do not give special awareness to these phenomena, rich transcriptions explicitly convert them into special phenomenon tokens as well as textual tokens. In previous studies, the textual and phenomenon tokens were simultaneously estimated in an end-to-end manner. However, it is difficult to build accurate RT-ASR systems because large-scale rich transcription-style datasets are often unavailable. To solve this problem, our training method uses a limited rich transcription-style dataset and common transcription-style dataset simultaneously. The Key process in our semi-supervised learning is to convert the common transcription-style dataset into a pseudo-rich transcription-style dataset. To this end, we introduce style tokens which control phenomenon tokens are generated or not into transformer-based autoregressive modeling. We use this modeling for generating the pseudo-rich transcription-style datasets and for building RT-ASR system from the pseudo and original datasets. Our experiments on spontaneous ASR tasks showed the effectiveness of the proposed method.

Index Terms: automatic speech recognition, rich transcription, semi-supervised learning, pseudo-labeling

1. Introduction

There has been increasing progress in end-to-end automatic speech recognition (ASR) that directly converts a speech into textual tokens (characters, words, etc.). End-to-end ASR is an approach with simpler training and faster decoding compared with traditional deep-neural-network hidden-Markov-model hybrid ASR. Various end-to-end ASR methods including connectionist temporal classification [1–3], recurrent neural network (RNN) encoder-decoder models [4–7], RNN transducers [8,9], and transformer encoder-decoder models [10,11], have been investigated. These methods have been successful in recognizing textual information in speech.

General ASR systems usually transcribe speech into only textual information, but spontaneous speech includes rich information. For example, spontaneous speech often includes speech phenomena such as fillers, word fragments, laughter, and coughs. Since the speech changes due to the occurrence of such phenomena, ASR errors are induced in general ASR systems. Laughter obscures speech, and fillers and word fragments make connections of textual tokens irregular. Therefore, it is important to consider these phenomena when estimating textual information for accurate ASR.

In previous studies, speech recognition and detecting speech phenomena were jointly modeled for forming an end-to-end rich transcription-style ASR (RT-ASR) system [12, 13]. Simultaneous estimation of textual information and speech phenomenon was achieved by treating the speech phenomena as phenomenon tokens in the same way as textual tokens. RT-ASR systems execute speech recognition by taking into account what kind of speech phenomena are occurring and can output textual and phenomenon tokens at the same time. The models in previous studies were trained from only rich transcription-style datasets including speech, textual tokens, and phenomenon tokens. However, it is difficult to build highly accurate RT-ASR systems only from the rich transcription-style dataset because collecting large-scale rich transcription-style datasets is costly and time-consuming.

In this paper, we propose a semi-supervised learning method for constructing RT-ASR systems using small-scale rich and large-scale common transcription-style datasets consisting of only speech and textual tokens. In contrast to previous studies, the key advance of our method is to handle both rich and common transcription-style datasets. In our semi-supervised learning method, we convert the common transcription-style dataset into a pseudo-rich transcription-style dataset and use it for further model training. Our key idea for handling two different style datasets is to introduce style tokens that indicated rich or common transcription style into transformer-based autoregressive modeling. By explicitly considering a transcription-style from the style token, the model can switch whether to output the phenomenon tokens. We apply this modeling to generating a pseudo-rich transcription-style dataset and building a final RT-ASR system. This RT-ASR system transcribes input speech into a token sequence that includes both textual tokens and phenomenon tokens in an end-to-end manner, the same as previous RT-ASR systems.

There have been several attempts to build an ASR system based on a semi-supervised learning approach [14–20]. For end-to-end ASR, pseudo-labeling methods that generate pseudo-transcriptions against unlabeled data have been increasingly popular [18–20]. Previous pseudo-labeling methods for end-to-end ASR involve generating pseudo-labels of textual tokens for unlabeled speech. In our method, we generate pseudo-transcriptions with phenomenon tokens for speech and textual token pairs. In other words, previous semi-supervised learning methods focused on general end-to-end ASR, whereas our method focuses on end-to-end RT-ASR.

We conduct experiments to evaluate the proposed method on Japanese spontaneous speech recognition tasks. The experimental results indicate that the proposed method enables better ASR performance in terms of the standard character er-

Table 1: *Speech phenomena in rich transcription-style dataset. Number of phenomena is that counted in 24-hour dataset.*

Phenomenon	Type	# phenomena
Fragment	single token	4,361
Laugh	single token	4,388
Cough	single token	73
Sigh	single token	25
Filler	enclosed tokens	18,558
Repetition	enclosed tokens	487
Misstatement	enclosed tokens	143
Stretch	enclosed tokens	3,296
Laughing	enclosed tokens	2,334
Hard to hear	enclosed tokens	3,176
Emphasis	enclosed tokens	821

ror rate (CER) and the CER including phenomenon tokens. Our method enables us to construct highly accurate RT-ASR systems since both a small-scale rich transcription-style and a large-scale common transcription-style datasets can be used for the training by using semi-supervised learning. We verify that RT-ASR system built with our method outperforms general end-to-end ASR systems when the same amount of data is used.

2. Rich Transcription-Style Dataset

A common transcription-style dataset is annotated with textual tokens but not with speech phenomena appearing in spontaneous speech. To handle various speech phenomena including fillers, repetitions, and laughing utterances, a rich transcription-style dataset in which such phenomena are annotated, is required. We use our natural two-person dialogue corpus (NTDC) in our experiments. The two individuals were given the roles of questioner and respondent. The questioner asked questions on a topic we determined beforehand. We recorded the dialogues and annotated textual and phenomenon tokens. The total amount of speech in NTDC is 24 hours, and the total number of textual tokens is 125,376.

It is thought that various types of speech phenomena are included in spontaneous speech. We roughly divided the phenomena into two types to create a rich transcription-style dataset. One type can be expressed as a single appearance (single token), and the other is something that can be expressed so as to enclose the textual tokens (enclosed tokens). Examples include fillers, repetitions, and laughing utterances. Table 1 shows the phenomena considered for RT-ASR and their frequency in the data. We define 4 types of single tokens and 7 types of enclosed tokens.

3. Proposed Semi-Supervised Learning Method for RT-ASR

3.1. Strategy

We take a semi-supervised learning approach to efficiently utilize a small-scale rich transcription-style dataset and large-scale common transcription-style dataset. Figure 1 illustrates flow of our semi-supervised learning method for RT-ASR. In our semi-supervised method, we convert common transcription-style dataset into pseudo-rich transcription-style dataset. The final model is trained using the pseudo-rich and rich transcription-style datasets.

For the semi-supervised learning, we introduce style tokens that indicate the rich or common transcription-style into
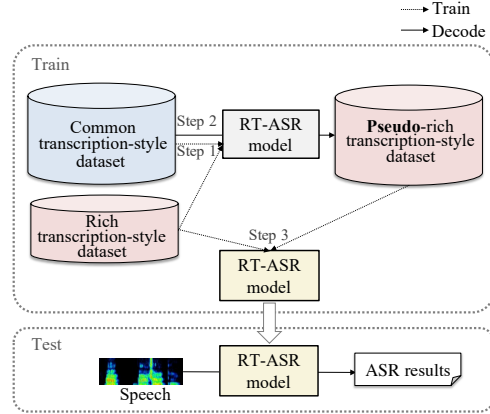

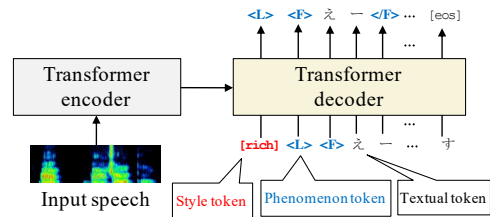Figure 1: *Flow of proposed semi-supervised learning method for RT-ASR.*

Figure 2: *End-to-end rich transcrption-style automatic speech recognition. This is the example of inference with style token of rich transcription-style.*

transformer-based autoregressive modeling. The modeling is used for generating pseudo-rich transcription-style dataset and building a final RT-ASR system. In training, we use a rich transcription-style dataset with a rich-style style token and a common transcription-style dataset with a common-style style token. The style tokens allow the model to learn whether to output the phenomenon tokens. In decoding, the model can output rich-style transcription by feeding the rich-style style token. This enables us to achieve efficient conversion of common transcription-style into rich transcription style.

3.2. Modeling

Our model directly converts acoustic features of input speech into a sequence of mixed textual tokens and phenomenon tokens. Given speech $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_I\}$ and style token $s \in \{[\texttt{rich}], [\texttt{common}]\}$ that represent rich transcription-style or common transcription-style, the encoder-decoder estimates the generative probability of a token sequence $\boldsymbol{W} = \{w_1, \cdots, w_T\}$, where I is the number of the acoustic features in the input speech and T is the number of the tokens in the token sequence. The generative probability of a sequence is defined as

$$P(\boldsymbol{W}|\boldsymbol{X}, s; \boldsymbol{\Theta}) = \prod_{t=1}^{T} P(w_t|w_{1:t-1}, \boldsymbol{X}, s; \boldsymbol{\Theta}), \quad (1)$$

where $\boldsymbol{\Theta}$ represents the trainable parameters.

We use transformer-based autoregressive models that predict the probability of a token given the previous predicted tokens. The encoder converts the input acoustic features $\boldsymbol{X}$ into hidden representations $\boldsymbol{f}^M$ by using M transformer encoder

blocks. The m-th transformer encoder block composes the m-th hidden representations $\boldsymbol{f}^m$ from the lower layer inputs $\boldsymbol{f}^{m-1}$ as

$$\boldsymbol{f}^m = \texttt{TransformerEncoderBlock}(\boldsymbol{f}^{m-1}; \theta_f), \quad (2)$$

where $\texttt{TransformerEncoderBlock}(\cdot)$ is a transformer encoder including a scaled dot-product multi-head self-attention layer and a position-wise feed-forward network and θ_f is a trainable parameter. The first input for the encoder $\boldsymbol{f}^0 = \{\boldsymbol{f}_1, \cdots \boldsymbol{f}_{I'}\}$ is computed from the input acoustic features as

$$\overline{\boldsymbol{f}}_{i'} = \texttt{PositionalEncoding}(\overline{\boldsymbol{x}}_{i'}), \quad (3)$$

$$\{\overline{\boldsymbol{x}}_1, \cdots, \overline{\boldsymbol{x}}_{I'}\} = \texttt{Convolution}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_I; \theta_c), \quad (4)$$

where $\texttt{PositionalEncoding}(\cdot)$ is a function that adds a continuous vector in which position information is embedded, $\texttt{Convolution}(\cdot)$ is a convolutional neural network consisting of convolutional and pooling layers, and θ_x is the trainable parameter.

The hidden representations of the final block $\boldsymbol{f}^M$ in the encoder is fed into the transformer decoder. When the output of the t-th time step for the n-th transformer block in the decoder is $\boldsymbol{g}_t^n$, the transformer decoder constructs a hidden representation from the lower output of the decoder $\boldsymbol{g}_{0:t-1}^{n-1} = \{\boldsymbol{g}_0^{n-1}, \cdots, \boldsymbol{g}_{t-1}^{n-1}\}$ as

$$\boldsymbol{g}_t^n = \texttt{TransformerDecoderBlock}(\boldsymbol{g}_{0:t-1}^{n-1}; \boldsymbol{f}^M, \theta_g), \quad (5)$$

where $\texttt{TransformerDecoderBlock}(\cdot)$ is a transformer decoder including a scaled dot-product multi-head masked self-attention layer, position-wise feed-forward network, and scaled dot product multi-head source-target attention layer, and θ_g is the trainable parameter. The input of the first transformer block is token embedding, which is calculated as

$$\boldsymbol{g}_t^0 = \texttt{PositionalEncoding}(\overline{\boldsymbol{w}}_t), \quad (6)$$

$$\overline{\boldsymbol{w}}_t = \texttt{Embedding}(w_t; \theta_w), \quad (7)$$

where $\texttt{Embedding}(\cdot)$ is a function that converts a token into a continuous representation and θ_w is the trainable parameter. In a typical encoder-decoder-based ASR model, the first token input into the decoder is $[\texttt{sos}]$ that represent the start-of-sentence. We use special style tokens s as the first input. In short, the first input is written as $w_0 = s$. Finally, the network estimates the probabilities of a distribution of the output tokens as

$$P(w_t|w_{1:t-1}; \boldsymbol{\Theta}) = \texttt{Softmax}(\boldsymbol{g}_{0:t-1}^N; \theta_o), \quad (8)$$

where $\texttt{Softmax}(\cdot)$ represents the softmax function with linear Transformation, N is the number of blocks in the transformer decoder, and θ_o is the trainable parameter. The model parameters can be summarized as $\boldsymbol{\Theta} = \{\theta_f, \theta_g, \theta_c, \theta_w, \theta_o\}$.

3.3. Semi-supervised learning

Step 1: First, we train a model to convert common transcription-style dataset into pseudo-rich transcription-style dataset. We use common transcription-style dataset $\mathcal{D}_c$ and rich transcription-style dataset $\mathcal{D}_r$ for the training. The model parameters $\boldsymbol{\Theta}$ are optimized so as to maximize the generative probability in the decoder when given an input speech and a style token. Thus, the model parameters are optimized by minimizing the cross entropy loss function as

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}}$$
$$- \sum_{\mathcal{D} \in \{\mathcal{D}_c, \mathcal{D}_r\}} \sum_{(\boldsymbol{W}', \boldsymbol{X}') \in \mathcal{D}} \log P(\boldsymbol{W}'|\boldsymbol{X}', s; \boldsymbol{\Theta}), \quad (9)$$

Table 2: *Details of datasets.*

Data	Corpus	# of textual tokens	# of phenomenon tokens	hours
Train	NTDC	548,886	116,906	22
	CSJ	12,568,607	-	545
Dev	NTDC	21268	4,083	1
	CSJ	68458	-	2
Eval	NTDC	22579	4387	1

$$s = \begin{cases} [\texttt{rich}] & \text{if } \mathcal{D} = \mathcal{D}_r, & (10a) \\ [\texttt{common}] & \text{if } \mathcal{D} = \mathcal{D}_c, & (10b) \end{cases}$$

where $\mathcal{D}$ is the training set.

Step 2: We convert a common transcription-style dataset into a pseudo-rich transcription-style dataset. The probabilities of textual and phenomenon tokens are estimated from input speech and a style token that indicates the common or rich transcription-style dataset. The generative probabilities of mixed textual and pseudo-phenomenon tokens $\bar{\boldsymbol{W}}$ are estimated with the trained parameters mentioned in **Step 1** by using following criterion:

$$\bar{\boldsymbol{W}} = \arg\max_{\boldsymbol{W}} P(\boldsymbol{W}|\boldsymbol{X}, s = [\texttt{rich}], \hat{\boldsymbol{\Theta}}). \quad (11)$$

This process is performed on all the data in common transcription-style dataset. The generated pseudo-rich transcription-style dataset with the same acoustic features as common transcription-style dataset is written as $\mathcal{D}_{pr} = \{(\bar{\boldsymbol{W}}^1, \boldsymbol{X}_c^1), \cdots, (\bar{\boldsymbol{W}}^{K_c}, \boldsymbol{X}_c^{K_c})\}$.

Step 3: At this point, we can use three datasets: common transcription-style dataset $\mathcal{D}_c$, rich transcription-style dataset $\mathcal{D}_r$ and pseudo-rich transcription-style dataset $\mathcal{D}_{pr}$. The model parameters $\boldsymbol{\Theta}$ are optimized by minimizing the cross entropy loss function with the three datasets as

$$\tilde{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}}$$
$$- \sum_{\mathcal{D} \in \{\mathcal{D}_c, \mathcal{D}_r, \mathcal{D}_{pr}\}} \sum_{(\boldsymbol{W}', \boldsymbol{X}') \in \mathcal{D}} \log P(\boldsymbol{W}'|\boldsymbol{X}', s; \boldsymbol{\Theta}), \quad (12)$$

$$s = \begin{cases} [\texttt{rich}] & \text{if } \mathcal{D} = \mathcal{D}_r \text{ or } \mathcal{D}_{pr}, & (13a) \\ [\texttt{common}] & \text{if } \mathcal{D} = \mathcal{D}_c. & (13b) \end{cases}$$

3.4. Decoding

In the decoding, beam search decoding is conducted with the trained parameters $\tilde{\boldsymbol{\Theta}}$ mentioned in **Step 3**. We use the following criterion during beam-search decoding with a rich-style style token:

$$\hat{\boldsymbol{W}} = \arg\max_{\boldsymbol{W}} P(\boldsymbol{W}|\boldsymbol{X}, s = [\texttt{rich}]; \tilde{\boldsymbol{\Theta}}). \quad (14)$$

The model can output both textual and phenomenon tokens by receiving the style token $[\texttt{rich}]$.

4. Experiments

4.1. Setups

4.1.1. Dataset

Table 2 shows details of the training, development, and evaluation sets. We used two corpora: the corpus of spontaneous

Table 3: *%CERs in different training data and input style tokens. C, R, and PR mean common transcription-style, rich transcription-style, and pseudo rich transcription-style datasets respectively. The style tokens are shown in parentheses in each dataset. SSL denote semi-supervised learning. Column of style token indicates whether or not a style token was used for models.*

System	Proposed SSL	Training Data	Style-Token	%CER w/o phenomena	%CER w/ phenomena
ASR	-	C	-	24.7	31.2
	-	C+R	-	23.5	34.1
RT-ASR	no	R	no	46.7	48.2
	no	C+R	yes	23.2	26.8
	yes	C+R+PR	yes	**22.7**	**26.2**

Japanese (CSJ) [21] as the common transcription-style dataset and NTDC as the rich transcription-style dataset. We split NTDC into 22 hours for the training set, 1 hour for the development set, and 1 hour for the evaluation set. The details of NTDC is described in Section 2. For CSJ, we used 545 hours for the training set and 2 hours for the development set.

4.1.2. ASR systems

We use 40-dimensional log Mel-filterbank with delta and acceleration coefficients as the acoustic features. We applied SpecAugment [22] to the training data. The vocabulary size was 3307 characters including textual tokens, phenomenon tokens, and style tokens for common transcription-style and rich transcription-style. The RT-ASR system was a transformer encoder-decoder. The encoder and decoder each had six transformer blocks. The token embedding dimension, hidden state dimension, non-linear layer dimension, and the number of heads were 256, 256, 2048, and 4, respectively. The acoustic features were transformed using a two-layers 2D convolutional neural network. We set the mini-batch size to 32. For regularization, we use label smoothing [23] with a smoothing parameter of 0.1 and set the dropout rate in the transformer blocks to 0.1 [24]. We used Adam optimizer [25] with Noam learning rate scheduler with 25000 warmup steps. Early stopping was applied if no best model was found in the development set for five epochs. Gradient clipping with a maximum norm of 5.0 was applied. When decoding by beam search, the beam size was set to 20. We prepared a general end-to-end ASR system with a transformer encoder-decoder for comparison with our RT-ASR system. The general end-to-to ASR system had the same configurations as the RT-ASR system. In the training of the general ASR system, the phenomenon tokes were deleted from the NTDC.

4.2. Results

Table 3 shows the speech recognition accuracies of the general end-to-end ASR system and the RT-ASR system in terms of CER with different training data. Evaluation was conducted using general CER for textual tokens and the CER for textual and phenomenon tokens. The results show that the RT-ASR system was more accurate than the general ASR system when the same common and rich transcription-style datasets (C+R) were used. This indicates that the style tokens allowed us to explicitly consider the transcription-styles as the contexts. The RT-ASR system with our proposed semi-supervised learning method showed better CER than the RT-ASR system trained from only rich transcription-style dataset (R). It was difficult to build an RT-ASR system with high accuracy when only the rich transcription-style dataset was used, but was easy with the pseudo-rich transcription-style dataset. The RT-ASR sys-
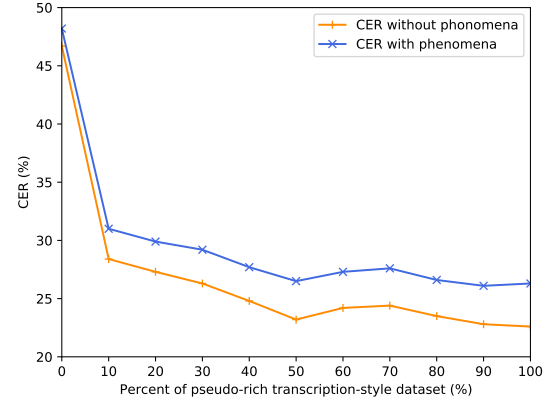

Figure 3: *CERs on evaluation set with different sizes of data of pseudo-rich transcription-style dataset.*

tem with our proposed semi-supervised learning method outperformed the general ASR and RT-ASR systems trained from common and rich transcription-style datasets (C+R). This system was trained from the same data as the system trained from common and rich transcription-style datasets (C+R) because pseudo-rich transcription-style dataset was generated from the common transcription-style dataset. This confirmed the effectiveness of our semi-supervised learning method.

We constructed models by changing the size of the pseudo-rich transcription-style dataset to confirm the effect of the size of this dataset on ASR performance. Figure 3 shows the CERs with different sizes of the pseudo-rich transcription-style dataset. The speech recognition accuracy improved as the dataset size increased from the case of 0% on the horizontal axis (only using the rich transcription-style dataset). These results confirmed that highly accurate RT-ASR systems can be built even with half the amount of the pseudo-rich transcription dataset.

5. Conclusion

We proposed a semi-supervised learning method for building RT-ASR systems from small-scale rich transcription-style dataset and large-scale common transcription-style datasets. With our semi-supervised learning method, we generate a pseudo-rich transcription-style dataset from a common transcription-style dataset. We introduced style tokens indicated rich transcription-style or common transcription-style to handle the two types of datasets efficiently. By considering speech phenomena in spontaneous speech and applying our semi-supervised learning method, the resulting RT-ASR system outperformed a general end-to-end ASR system with the same training data.

6. References

[1] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," *In Proc. International Conference on Machine Learning, (ICML)*, pp. 1764–1772, 2014.

[2] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *arXiv: 1412.5567*, 2014.

[3] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: end-to-end speech recognition using deep RNN models and wfst-based decoding," *In Proc. Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 167–174, 2015.

[4] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *In Proc. Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 577–585, 2015.

[5] L. Lu, X. Zhang, K. Cho, and S. Renals, "A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition," pp. 3249–3253, 2015.

[6] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," *In Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4945–4949, 2016.

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

[8] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," *In Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6645–6649, 2013.

[9] 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 Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 193–199, 2017.

[10] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," *In Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5884–5888, 2018.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 5998–6008, 2017.

[12] H. Inaguma, K. Inoue, M. Mimura, and T. Kawahara, "Social signal detection in spontaneous dialogue using bidirectional LSTM-CTC," *In Proc. of the International Speech Communication Association (INTERSPEECH)*, pp. 1691–1695, 2017.

[13] H. Fujimura, M. Nagao, and T. Masuko, "Simultaneous speech recognition and acoustic event detection using an LSTM-CTC acoustic model and a WFST decoder," *In Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5834–5838, 2018.

[14] L. Lamel, J. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 115–129, 2002.

[15] K. Veselý, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," *In Proc. of Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 267–272, 2013.

[16] S. Novotney, R. M. Schwartz, and J. Z. Ma, "Unsupervised acoustic and language model training with small amounts of labelled data," pp. 4297–4300, 2009.

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

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

[19] Y. Chen, W. Wang, and C. Wang, "Semi-supervised ASR by end-to-end self-training," *In Proc. of Conference of the International Speech Communication Association (INTERSPEEH)*, pp. 2787–2791, 2020.

[20] F. Weninger, F. Mana, R. Gemello, J. Andrés-Ferrer, and P. Zhan, "Semi-supervised learning with data augmentation for end-to-end ASR," *In Proc. of Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2802–2806, 2020.

[21] S. Furui, K. Maekawa, and H. Isahara, "A Japanese national project on spontaneous speech corpus and processing technology," *In Proc. ASR2000 - Automatic Speech Recognition: Challenges for the new Millenium*, pp. 244–248, 2000.

[22] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *In Proc. of Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2613–2617, 2019.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *In Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980