



Knowledge Distillation for Neural Transducer-based Target-Speaker ASR: Exploiting Parallel Mixture/Single-Talker Speech Data

*Takafumi Moriya, Hiroshi Sato, Tsubasa Ochiai, Marc Delcroix, Takanori Ashihara,
Kohei Matsuura, Tomohiro Tanaka, Ryo Masumura, Atsunori Ogawa, Taichi Asami*

NTT Corporation, Japan

takafumi.moriya@ntt.com

Abstract

Neural transducer (RNNT)-based target-speaker speech recognition (TS-RNNT) directly transcribes a target speaker's voice from a multi-talker mixture. It is a promising approach for streaming applications because it does not incur the extra computation costs of a target speech extraction frontend, which is a critical barrier to quick response. TS-RNNT is trained end-to-end given the input speech (i.e., mixtures and enrollment speech) and reference transcriptions. The training mixtures are generally simulated by mixing single-talker signals, but conventional TS-RNNT training does not utilize single-speaker signals. This paper proposes using knowledge distillation (KD) to exploit the parallel mixture/single-talker speech data. Our proposed KD scheme uses an RNNT system pretrained with the target single-talker speech input to generate pseudo labels for the TS-RNNT training. Experimental results show that TS-RNNT systems trained with the proposed KD scheme outperform a baseline TS-RNNT.

Index Terms: target-speaker speech recognition, neural transducer, end-to-end, streaming inference, knowledge distillation

1. Introduction

Although recent advances in end-to-end automatic speech recognition (ASR) frameworks have boosted ASR performance in the single-talker case [1–5], it remains difficult to recognize multi-talker speech if multiple voices are overlapped [6]. Target-speaker ASR (TS-ASR) [7, 8], which recognizes only the target speaker's speech from a mixture, is a promising technology for developing user-dependent voice interactive systems including smart speakers, smartphones, smartwatches, etc. In this paper, we focus on improving the TS-ASR performance.

One conventional approach to realizing TS-ASR is to use a cascade of a target speech extraction (TSE) frontend with an ASR backend [9–14]. First, the TSE frontend extracts the target speaker's voice from the speech mixtures by utilizing a speaker embedding that characterizes the target speaker. This embedding is derived by processing the enrollment utterance of the target speaker in a speaker encoder module. Then, the speech extracted by the TSE module is fed to the ASR module to obtain the target speaker's transcription.

In the training step of TSE, signal-level loss is computed using the estimated target speaker's speech and the reference clean speech of the target speaker. Thus, TSE training needs parallel data consisting of mixtures and the target speaker's reference/enrollment speech. In the inference step, the cascade approach improves the TS-ASR performance. However, the extra computation costs of the TSE module are a critical barrier to quick response, especially for streaming operation.

Recently, [7, 8] proposed an integrated TS-ASR framework that incorporates the TSE essence (i.e., conditioning on speaker embedding) into a neural transducer (RNNT) [15] model, called TS-RNNT. TS-RNNT can identify and transcribe the target speaker's speech while ignoring the interfering speakers' voices, by conditioning an RNNT model on the target speaker embedding. The computation cost is equivalent to that of a conventional RNNT model.

TS-RNNT training requires the input speech signals, i.e., mixtures and target speaker's enrollment utterances, and the target speaker's transcription. Although TS-RNNT training does not require single-talker speech, such clean source data is generally used to generate the simulated mixtures. Thus the use of single-talker speech is logical for TS-RNNT training. However, prior works [7, 8] did not directly utilize the plain single-talker speech for TS-RNNT training. We expect that we could improve TS-ASR performance further if we could utilize the parallel mixture/target speech for TS-RNNT training.

The TS-RNNT system is an end-to-end TS-ASR system that directly performs recognition without explicitly extracting the target speech signal. Therefore, it is not straightforward to utilize parallel mixture and single-talker speech data because it is not possible to define a signal-level loss term as is used in TSE. As an alternative, in this paper, we propose a knowledge distillation (KD)-based training framework to exploit the parallel mixture/single-talker speech data.

We generate pseudo labels (posteriors) by processing the single-talker speech with a teacher RNNT model trained on single-talker data. We then train the student TS-RNNT model using a multi-task loss consisting of the standard RNNT loss and a KD loss. The KD loss is the cross-entropy loss between the posteriors computed with the TS-RNNT (i.e., student) and the pseudo labels. The teacher's guidance in the KD loss may help the student learn posteriors that are more robust to the interference speakers. Thus, we expect that the TS-RNNT trained with the proposed KD will improve the TS-ASR performance.

We conduct experiments on parallel mixture/single-talker speech data, to compare our TS-RNNT proposal, trained with the proposed KD, called TS-RNNT+KD, with conventional TS-RNNT in both offline and streaming modes. Experiments show that TS-RNNT+KD outperforms the variant without KD, especially for streaming TS-ASR.

2. Related Work

While the KD framework is generally used to relieve performance degradation in model compression [5, 16, 17], some previous studies apply it for other purposes [4, 18–28]. In particular, the KD frameworks of [22, 23, 26] have been used for multi-talker ASR systems, where a single-talker ASR system with the target speaker's speech input is used to generate pseudo-labels.

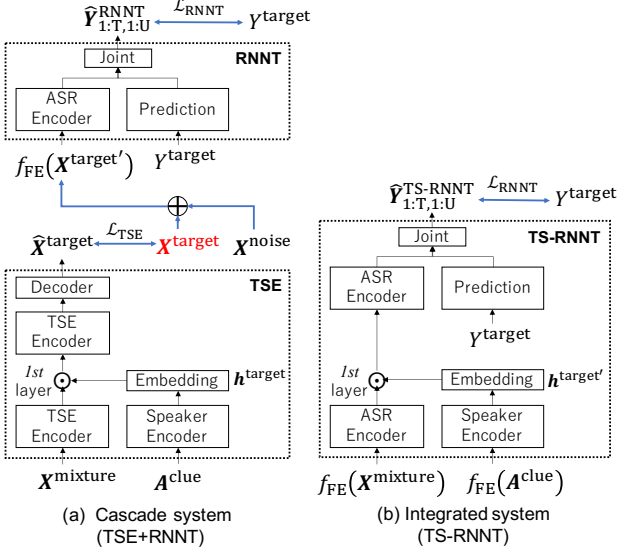


Figure 1: Overview of (a) cascaded and (b) integrated TS-ASR systems. $\hat{\mathbf{X}}^{\text{target}}$ is only used to train the cascaded system.

These works dealt with “separation” based systems that recognize the speech of all speakers in a mixture and identification of a target speaker is not considered. These frameworks require Permutation Invariant Training loss, making the training more complex due to the multiple decoders within the models as well as RNNT-based multi-talker ASR systems [29, 30]. They explored KD for neural network-Hidden Markov Model hybrid [22], and attentional encoder-decoder [23, 26] multi-talker ASR-based systems, and then only for offline settings.

In this work, we use KD for TS-RNNT, which simplifies the training as there is only a single target speaker. Thus TS-RNNT can naturally avoid output-speaker ambiguity and identify the target speaker. We use the approach developed for more recent RNNT systems, which allows the exploration of KD for streaming models.

3. Methods

In this section, we first explain the cascaded TS-ASR, i.e., TSE+RNNT, and the integrated TS-ASR, i.e., TS-RNNT, which are the foundations of this work. Then we introduce our proposed approach, i.e., TS-RNNT+KD.

Let $\mathbf{X}^{\text{mixture}} = [x_1^{\text{mixture}}, \dots, x_{T'}^{\text{mixture}}] \in \mathcal{R}^{T'}$ be the single microphone input speech mixture of duration T' ; it includes the target speech $\mathbf{X}^{\text{target}}$, interfering speakers’ voices $\mathbf{X}^{\text{interference}}$, and background noise $\mathbf{X}^{\text{noise}}$. $\mathbf{Y}^{\text{target}} = [y_1^{\text{target}}, \dots, y_U^{\text{target}}] \in \mathcal{R}^U$ is the sequence of tokens of length U associated with the utterance spoken by the target speaker, where $y_u^{\text{target}} \in \{1, \dots, K\}$. K is the number of tokens in the vocabulary.

3.1. Cascaded TS-ASR (TSE+RNNT)

Figure 1 (a) is a diagram of the cascaded TS-ASR system, which is composed of the TSE and ASR modules described below.

3.1.1. TSE frontend

In this work, we adopt the time-domain TSE module [12, 13] that extracts $\hat{\mathbf{X}}^{\text{target}}$ from $\mathbf{X}^{\text{mixture}}$ using enrollment speech \mathbf{A}^{clue} . First, we extract the embedding of the target-speaker, $\mathbf{h}^{\text{target}}$, from \mathbf{A}^{clue} using speaker encoder $f^{\text{Spk-Enc}}(\cdot)$, which consists of a multi-layer neural network followed by a linear layer and a time-average pooling layer. Then, we use a speech ex-

traction network $f^{\text{SE}}(\cdot)$, which consists of TSE encoder and decoder, to extract the target speech given the embedding, $\mathbf{h}^{\text{target}}$. The above operations, which yield the estimated target speaker’s speech signal $\hat{\mathbf{X}}^{\text{target}}$, are defined as follows:

$$\mathbf{h}^{\text{target}} = f^{\text{Spk-Enc}}(\mathbf{A}^{\text{clue}}; \theta^{\text{Spk-Enc}}), \quad (1)$$

$$\hat{\mathbf{X}}^{\text{target}} = f^{\text{SE}}(\mathbf{X}^{\text{mixture}}, \mathbf{h}^{\text{target}}; \theta^{\text{SE}}). \quad (2)$$

In this work, we utilize the Hadamard product between $\mathbf{h}^{\text{target}}$ and the first encoder layer output of $f^{\text{SE}}(\cdot)$. The parameters $\theta^{\text{TSE}} \triangleq [\theta^{\text{Spk-Enc}}, \theta^{\text{SE}}]$ are jointly optimized with the scale-invariant source-to-noise ratio (SI-SNR) loss \mathcal{L}_{TSE} [31] using $\hat{\mathbf{X}}^{\text{target}}$ and reference $\mathbf{X}^{\text{target}}$.

3.1.2. ASR backend (RNNT)

We use an RNNT-based ASR model [15], which can perform streaming ASR, for the backend module. RNNT learns the mapping between sequences of different lengths. First, in the training step, $\mathbf{X}^{\text{target}'}$ consisting of $\mathbf{X}^{\text{target}}$ with background noise $\mathbf{X}^{\text{noise}}$ to enhance system robustness is transformed into acoustic features using feature extractor $f^{\text{FE}}(\cdot)$. Then, the features are encoded into a length T sequence, \mathbf{H}^{ASR} , via ASR encoder network $f^{\text{ASR-Enc}}(\cdot)$. Next, $\mathbf{Y}^{\text{target}}$ is also encoded into \mathbf{H}^{Pred} via a prediction network $f^{\text{Pred}}(\cdot)$. \mathbf{H}^{ASR} and \mathbf{H}^{Pred} are fed to a joint network, $f^{\text{Joint}}(\cdot)$, to compute the token posterior probabilities, $\hat{\mathbf{Y}}_{1:T,1:U}^{\text{RNNT}} \in \mathcal{R}^{T \times U \times K}$. The above operations can be formularized as follows:

$$\mathbf{H}^{\text{ASR}} = f^{\text{ASR-Enc}}(f^{\text{FE}}(\mathbf{X}^{\text{target}'}); \theta^{\text{ASR-Enc}}), \quad (3)$$

$$\mathbf{H}^{\text{Pred}} = f^{\text{Pred}}(\mathbf{Y}^{\text{target}}; \theta^{\text{Pred}}), \quad (4)$$

$$\hat{\mathbf{Y}}_{1:T,1:U}^{\text{RNNT}} = \text{Softmax}(f^{\text{Joint}}(\mathbf{H}^{\text{ASR}}, \mathbf{H}^{\text{Pred}}; \theta^{\text{Joint}})), \quad (5)$$

where $\text{Softmax}(\cdot)$ performs a softmax operation. All the parameters $\theta^{\text{RNNT}} \triangleq [\theta^{\text{ASR-Enc}}, \theta^{\text{Pred}}, \theta^{\text{Joint}}]$ are optimized with RNNT loss $\mathcal{L}_{\text{RNNT}}$ using $\hat{\mathbf{Y}}_{1:T,1:U}^{\text{RNNT}}$ and $\mathbf{Y}^{\text{target}}$.

For training, we use single-talker speech $\mathbf{X}^{\text{target}'}$ consisting of $\mathbf{X}^{\text{target}}$ mixed with $\mathbf{X}^{\text{noise}}$ and the transcript $\mathbf{Y}^{\text{target}}$. Note that we do not perform joint training of the TSE and RNNT models. In the decoding step, we use $\hat{\mathbf{X}}^{\text{target}}$ extracted from the TSE.

3.2. Integrated TS-ASR with RNNT (TS-RNNT)

Figure 1 (b) is a schematic diagram of the TS-RNNT system. TS-RNNT deals with the target speaker embedding $\mathbf{h}^{\text{target}'}$ to inform which speaker in the mixture to decode; it can directly transduce the mixture to yield the target speaker’s transcription. TS-RNNT encoder $f^{\text{ASR-Enc}'}(\cdot)$ receives $f^{\text{FE}}(\mathbf{X}^{\text{mixture}})$ and $\mathbf{h}^{\text{target}'}$ extracted from $f^{\text{FE}}(\mathbf{A}^{\text{clue}})$ via $f^{\text{Spk-Enc}'}(\cdot)$ of TS-RNNT, and outputs $\mathbf{H}^{\text{ASR}'}$. These functions are defined as follows:

$$\mathbf{h}^{\text{target}'} = f^{\text{Spk-Enc}'}(f^{\text{FE}}(\mathbf{A}^{\text{clue}}); \theta^{\text{Spk-Enc}'}), \quad (6)$$

$$\mathbf{H}^{\text{ASR}'} = f^{\text{ASR-Enc}'}(f^{\text{FE}}(\mathbf{X}^{\text{mixture}}), \mathbf{h}^{\text{target}'}; \theta^{\text{ASR-Enc}'}). \quad (7)$$

$\mathbf{h}^{\text{target}'}$ and the first layer ASR encoder output are multiplied as the Hadamard product within $f^{\text{ASR-Enc}'}(\cdot)$. The prediction and joint networks are the same as in the standard RNNT of 3.1.2, and TS-RNNT also outputs the posterior probabilities in tensor format $\hat{\mathbf{Y}}_{1:T,1:U}^{\text{TS-RNNT}} \in \mathcal{R}^{T \times U \times K}$. All networks with parameters $\theta^{\text{TS-RNNT}} \triangleq [\theta^{\text{Spk-Enc}'}, \theta^{\text{ASR-Enc}'}, \theta^{\text{Pred}}, \theta^{\text{Joint}}]$ are jointly optimized by using RNNT loss $\mathcal{L}_{\text{RNNT}}$.

For decoding, we enroll $\mathbf{h}^{\text{target}'}$ extracted from $f^{\text{FE}}(\mathbf{A}^{\text{clue}})$ in advance. Then, the acoustic feature of the mixture signal,

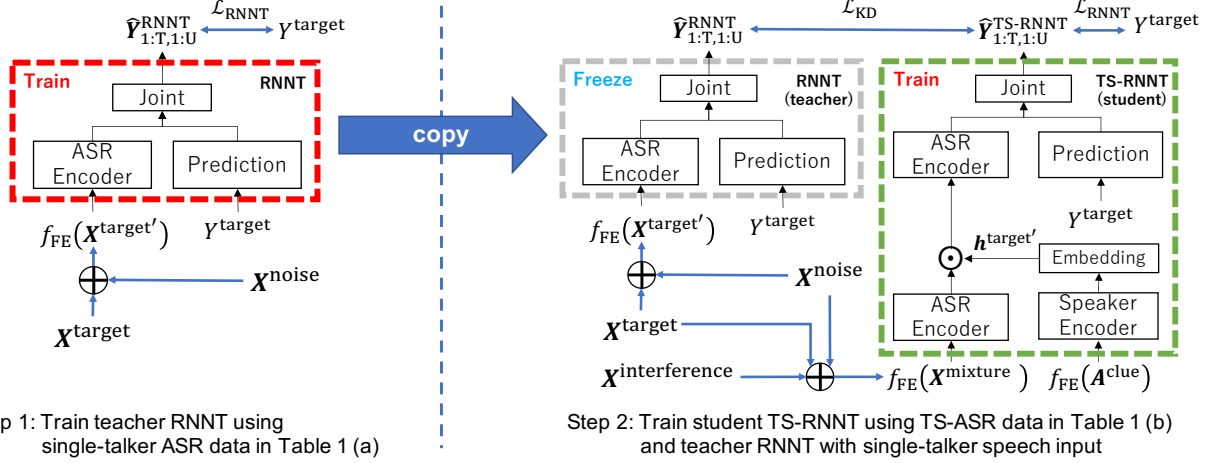


Figure 2: Schematic diagram of TS-RNNT+KD for TS-RNNT. The training process consists of two steps: 1) pretraining the teacher RNNT using single-talker ASR data with $\mathcal{L}_{\text{RNNT}}$ as the loss, and 2) training the student TS-RNNT using the mixture and target speaker’s transcription pair data (TS-ASR data) and the pre-trained RNNT outputs with multi-task loss $\mathcal{L}_{\text{RNNT+KD}}$ combining $\mathcal{L}_{\text{RNNT}}$ and \mathcal{L}_{KD} .

Table 1: Data generation setup. The number of utterances in (a) is double of (b) due to the single-speaker case of (b).

	dataset	mixture type	SIR [dB]	SNR [dB]	#speakers (train set / dev set)	#mixtures or utterances (train set / dev set)
(a)	training data for ASR backend (RNNT)	1 speaker and noise	-	0 - 20	3054 / 160	400,000 / 10,000
(b)	training data for TSE and TS-RNNT	2 speakers and noise	-5 - 5	0 - 20	3054 / 160	200,000 / 5,000
(c)	evaluation data	2 speakers and noise	-5 - 5	0, 5, 10, 15, 20	30	$6,000 \times 5 = 30,000$

$f^{\text{FE}}(\mathbf{X}^{\text{mixture}})$, is directly input to the ASR encoder together with $\mathbf{h}^{\text{target}'}$, and the decoder yields the TS-ASR results. Thanks to the removal of the TSE frontend module, TS-RNNT can perform TS-ASR faster than the cascaded system, while its computation cost equals that of basic RNNT. TS-RNNT training uses the target speakers’ voices and the transcriptions, which are arguably easier to collect for real recordings; it does not need clean target speakers’ speech data as references for TSE module training.

3.3. Proposed KD for TS-RNNT (TS-RNNT+KD)

TS-RNNT does not require single-talker speech in the training step. However, speech mixtures are generally simulated using single-talker speech data. Therefore, target single-speaker speech $\mathbf{X}^{\text{target}}$ is naturally available for training. In this paper, we exploit the parallel speech data, i.e., $\mathbf{X}^{\text{target}}$ and $\mathbf{X}^{\text{mixture}}$, using a KD framework.

The schematic of our proposed KD approach of exploiting parallel speech data for TS-RNNT is illustrated in Figure 2. Our proposed KD for TS-RNNT training consists of two steps; 1) building a single-talker teacher model and 2) distilling knowledge from the pretrained single-talker RNNT with target speaker’s speech input to a TS-RNNT model. The procedure is detailed as follows.

First, we train single-talker RNNT as a teacher model using $\mathbf{X}^{\text{target}'}$, which consists of $\mathbf{X}^{\text{target}}$ mixed with $\mathbf{X}^{\text{noise}}$, and $\mathbf{Y}^{\text{target}}$; the RNNT is optimized by $\mathcal{L}_{\text{RNNT}}$. We assume that the pretrained single-talker RNNT with single-talker speech input can better align the speech and the transcriptions, thus providing more reliable posteriors than those of TS-RNNT. Then, we train a student TS-RNNT using the mixture, and the pretrained teacher RNNT outputs $\hat{\mathbf{Y}}_{1:T,1:U}^{\text{RNNT}}$. This is achieved by feeding the single-talker speech $\mathbf{X}^{\text{target}'}$ to the pretrained single-talker RNNT as the teacher. In this stage, we freeze the parameters of the single-talker RNNT model. We train the student model using multi-task loss $\mathcal{L}_{\text{RNNT+KD}} = \mathcal{L}_{\text{RNNT}} + \lambda \mathcal{L}_{\text{KD}}$, where $\mathcal{L}_{\text{RNNT}}$

is computed using $\hat{\mathbf{Y}}_{1:T,1:U}^{\text{TS-RNNT}}$ and $\mathbf{Y}^{\text{target}}$, and \mathcal{L}_{KD} is the cross-entropy loss between $\hat{\mathbf{Y}}_{1:T,1:U}^{\text{RNNT}}$ and $\hat{\mathbf{Y}}_{1:T,1:U}^{\text{TS-RNNT}}$. λ is a hyperparameter for balancing the losses between $\mathcal{L}_{\text{RNNT}}$ and \mathcal{L}_{KD} . KD loss is defined as follows:

$$\mathcal{L}_{\text{KD}} = - \sum_{t=1}^T \sum_{u=1}^U \sum_{k=1}^K \hat{y}_{t,u,k}^{\text{RNNT}} \log \hat{y}_{t,u,k}^{\text{TS-RNNT}}, \quad (8)$$

where $\hat{y}_{t,u,k}^{\text{RNNT}}$ and $\hat{y}_{t,u,k}^{\text{TS-RNNT}}$ correspond to the k -th class probability of $\hat{\mathbf{Y}}_{1:T,1:U}^{\text{RNNT}}$ and $\hat{\mathbf{Y}}_{1:T,1:U}^{\text{TS-RNNT}}$ at the t -th time and u -th label steps, respectively. The KD loss \mathcal{L}_{KD} replaces the signal-level loss on the TSE output of cascade signals when there is no explicit target speech signal estimation.

4. Experiments

4.1. Data

For the evaluation, we used the Corpus of Spontaneous Japanese (CSJ) [32] and the CHiME-3 corpus [33] to simulate the mixture. The details are shown in Table 1. The mixture consists of speech signals taken from the CSJ corpus, with background noise taken from the CHiME-3 corpus at signal-to-noise ratio (SNR) between 0 and 20 dB. The training data contains samples totaling 800 hours. The overlap ratio of the mixtures in both training and evaluation datasets was about 89% on average. The speakers between training, development, and evaluation datasets are different. In this paper, we adopt 3262 characters for the ASR tasks. We evaluated performance in terms of character error rate (CER) due to the ambiguity of Japanese word boundaries.

4.2. System configuration of TSE module

We adopted a Conv-TasNet [31]-based time-domain Speaker-Beam structure as a TSE frontend [12, 34]. The details of the implementation are similar to those in [12]. In this work, we trained offline and streaming TSE models. All TSE models were trained with dataset (b) in Table 1. Note that the convolution and global layer normalization of offline TSE were re-

Table 2: Comparisons of baseline cascade, baseline integrated, and proposed TS-ASR systems. The system ID uses the following notations, “B”, “P”, “O” “S”, for baseline, proposed, offline, and streaming, respectively.

System	λ	ID	CERs [%] of offline systems on each SNR						Avg.	ID	CERs [%] of streaming systems on each SNR					
			20dB	15dB	10dB	5dB	0dB	Avg.			20dB	15dB	10dB	5dB	0dB	Avg.
TSE+RNNT	-	BO1	7.9	8.8	11.1	18.3	36.3	16.5	BS1	16.8	18.6	23.8	35.6	56.3	30.2	
TS-RNNT	0.0	BO2	8.6	9.3	11.4	17.3	32.7	15.8	BS2	11.9	13.0	16.0	23.8	41.2	21.2	
TS-RNNT+KD (proposed)	1.0	PO1	8.9	9.5	11.6	17.5	34.2	16.3	PS1	11.6	12.9	15.8	24.2	42.2	21.4	
	0.5	PO2	8.5	9.1	11.2	16.7	33.0	15.7	PS2	11.7	12.7	15.7	23.7	41.4	21.0	
	0.1	PO3	8.1	8.7	10.7	16.1	31.4	15.0	PS3	11.0	12.2	15.0	22.4	39.8	20.1	
	0.01	PO4	8.5	9.1	11.0	17.0	32.1	15.5	PS4	10.2	11.3	14.2	21.7	38.7	19.2	
	0.001	PO5	8.8	9.4	11.3	16.8	31.7	15.6	PS5	10.7	11.8	14.7	22.3	39.2	19.7	

placed with causal convolution and channel-wise layer normalization in the streaming one, respectively. The algorithmic latency of the streaming TSE model is 1.25ms, which is negligible in terms of ASR decoding latency. The source-to-distortion improvements in offline and streaming TSE models were 15.1dB and 11.1dB, respectively, which mirrors the tendency reported in prior separation studies [31].

4.3. System configuration of ASR module

We used an 80-dimensional log Mel-filterbank as the input feature of ASR models. SpecAugment [35] was applied to the feature during training. We investigated two versions of (TS-) RNNT. First, we tested with an offline system consisting of the same encoder architecture as Conformer (L) [1] with a kernel size of 15. The ASR encoder contains two-layer 2D-convolutional neural networks (CNNs) followed by 17 Conformer blocks. The stride sizes of both max-pooling layers at each CNN layer were set to 2×2 . The prediction network had a 768-dimensional uni-directional long short-term memory layer. The joint network consisted of a 640-dimensional feed-forward network and output the posterior probabilities. The speaker encoder for TS-RNNT had the same architecture as the ASR encoder, while the number of blocks was reduced from 17 to 6.

We also performed a streaming experiment, and compared our proposed variant with a streaming system that used a similar configuration as the offline system except that the ASR encoder is replaced by the streaming Conformer encoder [3]. We adopted causal depthwise convolution and layer normalization for the streaming Conformer encoder that was trained with an attention mask strategy as in [3]. The history and current chunk sizes of streaming Conformer were set to 68 and 60 frames, respectively. Thus the average latency was 330ms ($= 600\text{ms}/2 + 30\text{ms}$).

“RNNT” and “TS-RNNT” were trained with the single-talker data (a) and the mixture data (b) in Table 1, respectively. The parameters of offline Conformer model were randomly initialized. The streaming Conformer parameters were initialized with those of a trained offline Conformer. We used the Adam optimizer with 25k warmup for a total of 100 epochs, and all models were trained using RNNT loss. The minibatch size was set to 64 in all experiments. For decoding, we performed alignment-length synchronous decoding with beam width of 8 [36]. We used the Kaldi and ESPnet toolkits [37, 38] for all implementation, data preprocessing, training, and evaluation processes.

4.4. Results

4.4.1. Baseline TSE+RNNT vs. Baseline TS-RNNT

First, we compare the baseline cascade system B*1 with baseline integrated systems B*2 for offline and streaming modes. The left and right blocks of Table 2 show the CERs under each SNR condition of the offline and streaming systems, respectively. Although the offline cascade system (BO1) contains the

strong TSE model, the offline integrated system (BO2), i.e., TS-RNNT without any frontend module, achieves comparable or better CERs than BO1. We also compare the streaming cascade system (BS1) with the streaming integrated system (BS2). The CERs of BS2 are much better than those of BS1 under all conditions. The reason is that the performance of BS1 heavily depends on the quality of the TSE output. The TS-RNNT could avoid the performance degradation caused by using the streaming TSE module. Hereafter, the TS-RNNTs (BO2 and BS2) are regarded as the offline and streaming baseline TS-ASR systems, respectively.

4.4.2. Baseline TS-RNNT vs. Proposed TS-RNNT+KD

Next, we compare the baseline TS-RNNT systems (BO2 and BS2) with the proposed systems “TS-RNNT+KD” (PO* and PS*), which were trained with $\mathcal{L}_{\text{RNNT+KD}}$ described in 3.3. The CERs under each SNR condition are shown in Table 2. The results of offline and streaming TS-RNNT+KD are denoted as “PO*” and “PS*”, respectively.

The offline experiment shows that the offline system PO3 achieved the best recognition performance among the TS-RNNT+KD system by tuning the hyperparameter $\lambda = 0.1$ on the development set. The averaged CER of PO3 was better than that of BO2 while retaining the decoding speed. We performed the MAPSSWE significance test [39], and the differences of the CERs between BO2 and PO3 under all conditions were statistically significant, $p < 0.001$. The relative CER reduction was 5.1% on average.

The streaming experiment shows that the proposed streaming system PS4 ($\lambda = 0.01$) greatly outperforms the baseline system BS2. The relative CER reduction between the integrated system BS2 and the proposed system PS4 was 9.4%, and the differences under all conditions were also statistically significant ($p < 0.001$) by performing MAPSSWE significance test. These results show the effectiveness of our proposed loss term $\mathcal{L}_{\text{RNNT+KD}}$ in exploiting parallel mixture/single-talker speech data for TS-RNNT.

5. Conclusion

We have proposed a KD framework that allows us to exploit the parallel mixture/single-talker speech data for improving an end-to-end TS-ASR model, called TS-RNNT. We train a teacher RNNT with single-speaker speech and transcription. The pre-trained model with target single-talker speech input is used to obtain a clearer posterior of the target speaker than is possible with the conventional TS-RNNT. Then the posterior and the TS-RNNT output are used for KD loss computation. The posterior of the single-talker ASR with the target speaker’s speech input guided the TS-RNNT output to behave similarly, resulting in improved TS-ASR performance. Our proposed system TS-RNNT+KD offers better performance than TSE+RNNT and TS-RNNT in both streaming and offline settings. The improvement is particularly significant in streaming operation.

6. References

- [1] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented Transformer for speech recognition,” in *Proc. of INTERSPEECH*, 2020, pp. 5036–5040.
- [2] T. N. Sainath, Y. He, A. Narayanan, R. Botros, R. Pang, D. Rybach, C. Allauzen, E. Variiani, J. Qin, Q.-N. Le-The, S. yiin Chang, B. Li, A. Gulati, J. Yu, C.-C. Chiu, D. Caseiro, W. Li, Q. Liang, and P. Rondon, “An efficient streaming non-recurrent on-device end-to-end model with improvements to rare-word modeling,” in *Proc. of INTERSPEECH*, 2021, pp. 1777–1781.
- [3] X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li, “Developing real-time streaming Transformer Transducer for speech recognition on large-scale dataset,” in *Proc. of ICASSP*, 2021, pp. 5904–5908.
- [4] G. Kurata and G. Saon, “Knowledge distillation from offline to streaming RNN Transducer for end-to-end speech recognition,” in *Proc. of INTERSPEECH*, 2020, pp. 2117–2121.
- [5] S. Panchapagesan, D. S. Park, C. Chiu, Y. Shangguan, Q. Liang, and A. Gruenstein, “Efficient knowledge distillation for RNN-Transducer models,” in *Proc. of ICASSP*, 2021, pp. 5639–5643.
- [6] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, “The fifth ‘CHiME’ speech separation and recognition challenge: dataset, task and baselines,” *Proc. of INTERSPEECH*, pp. 1561–1565, 2018.
- [7] T. Moriya, H. Sato, T. Ochiai, M. Delcroix, and T. Shinozaki, “Streaming target-speaker ASR with Neural Transducer,” in *Proc. of INTERSPEECH*, 2022, pp. 2673–2677.
- [8] —, “Streaming end-to-end target-speaker automatic speech recognition and activity detection,” *IEEE Access*, vol. 11, pp. 13 906–13 917, 2023.
- [9] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černocký, “SpeakerBeam: Speaker aware neural network for target speaker extraction in speech mixtures,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.
- [10] M. Delcroix, K. Žmolíková, K. Kinoshita, A. Ogawa, and T. Nakatani, “Single channel target speaker extraction and recognition with speaker beam,” in *Proc. of ICASSP*, 2018, pp. 5554–5558.
- [11] Q. Wang, I. L. Moreno, M. Saglam, K. Wilson, A. Chiao, R. Liu, Y. He, W. Li, J. Pelecanos, M. Nika *et al.*, “Voicefilter-lite: Streaming targeted voice separation for on-device speech recognition,” in *Proc. of INTERSPEECH*, 2020, pp. 2677–2681.
- [12] H. Sato, T. Ochiai, M. Delcroix, K. Kinoshita, T. Moriya, and N. Kamo, “Should we always separate?: Switching between enhanced and observed signals for overlapping speech recognition,” in *Proc. of INTERSPEECH*, 2021, pp. 1149–1153.
- [13] H. Sato, T. Ochiai, M. Delcroix, K. Kinoshita, N. Kamo, and T. Moriya, “Learning to enhance or not: Neural network-based switching of enhanced and observed signals for overlapping speech recognition,” in *Proc. of ICASSP*, 2022, pp. 6287–6291.
- [14] J. Shi, C. Zhang, C. Weng, S. Watanabe, M. Yu, and D. Yu, “Improving RNN Transducer with target speaker extraction and neural uncertainty estimation,” in *Proc. of ICASSP*, 2021, pp. 6908–6912.
- [15] A. Graves, “Sequence transduction with recurrent neural networks,” in *Proc. of ICML*, 2012.
- [16] G. E. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in *Advances in NeurIPS*, 2014.
- [17] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, “Learning small-size DNN with output-distribution-based criteria,” *Proc. of INTERSPEECH*, pp. 1910–1914, 2014.
- [18] T. Asami, R. Masumura, Y. Yamaguchi, H. Masataki, and Y. Aono, “Domain adaptation of DNN acoustic models using knowledge distillation,” in *Proc. of ICASSP*, 2017, pp. 5185–5189.
- [19] S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, “Student-teacher network learning with enhanced features,” in *Proc. of ICASSP*, 2017, pp. 5275–5279.
- [20] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, “Efficient knowledge distillation from an ensemble of teachers,” in *Proc. of INTERSPEECH*, 2017, pp. 3697–3701.
- [21] W. Li, Y. Zhang, P. Zhang, and F. Ge, “Multichannel ASR with knowledge distillation and generalized cross correlation feature,” in *Proc. of SLT*, 2018, pp. 463–469.
- [22] T. Tan, Y. Qian, and D. Yu, “Knowledge transfer in permutation invariant training for single-channel multi-talker speech recognition,” in *Proc. of ICASSP*, 2018, pp. 571–5718.
- [23] W. Zhang, X. Chang, and Y. Qian, “Knowledge distillation for end-to-end monaural multi-talker ASR system,” in *Proc. of INTERSPEECH*, 2019, pp. 2633–2637.
- [24] Z. Meng, J. Li, Y. Gaur, and Y. Gong, “Domain adaptation via teacher-student learning for end-to-end speech recognition,” in *Proc. of ASRU*, 2019, pp. 268–275.
- [25] T. Moriya, H. Sato, T. Tanaka, T. Ashihara, R. Masumura, and Y. Shinohara, “Distilling attention weights for CTC-based ASR systems,” in *Proc. of ICASSP*, 2020, pp. 6894 – 6898.
- [26] W. Zhang, X. Chang, Y. Qian, and S. Watanabe, “Improving end-to-end single-channel multi-talker speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1385–1394, 2020.
- [27] T. Moriya, T. Ochiai, S. Karita, H. Sato, T. Tanaka, T. Ashihara, R. Masumura, Y. Shinohara, and M. Delcroix, “Self-Distillation for improving CTC-Transformer-based ASR systems,” in *Proc. of INTERSPEECH*, 2020, pp. 546–550.
- [28] A. Kojima, “Knowledge distillation for streaming Transformer-Transducer,” in *Proc. of INTERSPEECH*, 2021, pp. 2841–2845.
- [29] I. Sklyar, A. Piunova, and Y. Liu, “Streaming multi-speaker ASR with RNN-T,” in *Proc. of ICASSP*, 2021, pp. 6903–6907.
- [30] L. Lu, N. Kanda, J. Li, and Y. Gong, “Streaming multi-talker speech recognition with joint speaker identification,” in *Proc. of INTERSPEECH*, 2021, pp. 1782–1786.
- [31] Y. Luo and N. Mesgarani, “Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [32] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, “Spontaneous speech corpus of Japanese,” in *Proc. of LREC*, 2000, pp. 947–9520.
- [33] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, “The third ‘CHiME’ speech separation and recognition challenge: Analysis and outcomes,” *Computer Speech & Language*, vol. 46, pp. 605–626, 2017.
- [34] M. Delcroix, T. Ochiai, K. Zmolikova, K. Kinoshita, N. Tawara, T. Nakatani, and S. Araki, “Improving speaker discrimination of target speech extraction with time-domain SpeakerBeam,” in *Proc. of ICASSP*, 2020, pp. 691–695.
- [35] 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,” *Proc. of INTERSPEECH*, pp. 2613–2617, 2019.
- [36] G. Saon, Z. Tuske, and K. Audhkhasi, “Alignment-length synchronous decoding for RNN Transducer,” in *Proc. of ICASSP*, 2020, pp. 7799–7803.
- [37] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovski, G. Stemmer, and K. Vesel, “The Kaldi speech recognition toolkit,” in *Proc. of ASRU*, 2011.
- [38] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, “ESPnet: End-to-end speech processing toolkit,” in *Proc. of INTERSPEECH*, 2018, pp. 2207–2211.
- [39] L. Gillick and S. Cox, “Some statistical issues in the comparison of speech recognition algorithms,” in *Proc. of ICASSP*, 1989, pp. 532–535.