End-to-End Automatic Speech Recognition with Deep Mutual Learning

Ryo Masumura*, Mana Ihori*, Akihiko Takashima*, Tomohiro Tanaka*, Takanori Ashihara*
* NTT Media Intelligence Laboratories, NTT Corporation, Japan
E-mail: ryou.masumura.ba@hco.ntt.co.jp

***Abstract*—This paper is the first study to apply deep mutual learning (DML) to end-to-end ASR models. In DML, multiple models are trained simultaneously and collaboratively by mimicking each other throughout the training process, which helps to attain the global optimum and prevent models from making over-confident predictions. While previous studies applied DML to simple multi-class classification problems, there are no studies that have used it on more complex sequence-to-sequence mapping problems. For this reason, this paper presents a method to apply DML to state-of-the-art Transformer-based end-to-end ASR models. In particular, we propose to combine DML with recent representative training techniques. i.e., label smoothing, scheduled sampling, and SpecAugment, each of which are essential for powerful end-to-end ASR models. We expect that these training techniques work well with DML because DML has complementary characteristics. We experimented with two setups for Japanese ASR tasks: large-scale modeling and compact modeling. We demonstrate that DML improves the ASR performance of both modeling setups compared with conventional learning methods including knowledge distillation. We also show that combining DML with the existing training techniques effectively improves ASR performance.**

Index Terms: end-to-end ASR, deep mutual learning, Transformer, scheduled sampling, SpecAugment

I. Introduction

In the automatic speech recognition (ASR) field, there has been growing interest in developing end-to-end ASR systems that directly convert input speech into text. While traditional ASR systems have been built from noisy channel formulations using several component models (i.e., an acoustic model, language model, and pronunciation model), end-to-end ASR systems can learn the overall conversion in one step without any intermediate processing.

Modeling methods and training techniques help to achieve powerful end-to-end ASR models. Recent studies have developed modeling methods that include connectionist temporal classification [1], [2], a recurrent neural aligner [3], a recurrent neural network (RNN) transducer [4], and an RNN encoder-decoder [5]–[9]. In particular, Transformer-based modeling methods have shown the strongest performance in recent studies [10]–[15]. In addition, a few effective training techniques are label smoothing [16], scheduled sampling [17], and SpecAugment [18], [19]. These techniques effectively prevent over-fitting problems caused by maximum likelihood estimation, and combining them can improve end-to-end ASR systems [20]. Furthermore, recent studies have focused on building compact models because computation complexity and memory efficiency must be considered in practice. The most representative technique is knowledge distillation [21] (i.e., teacher-student learning) that trains compact student models to mimic a pre-trained large-scale teacher model. In fact, knowledge distillation is an effective compact end-to-end ASR modeling technique [22]–[24].

To achieve a more powerful and compact end-to-end ASR model, we focused on deep mutual learning (DML) [25], one of the most successful learning strategies in recent machine learning studies. In DML, multiple student models simultaneously learn to solve a target task collaboratively without introducing pre-trained teacher models. In fact, each student model is constrained to mimic other student models, thereby helping it to find a global optimum and prevent it from making over-confident predictions. DML enables us to construct stronger models using a unified network structure rather than independent learning. In addition, DML can be used to obtain compact models that perform better than those distilled from a strong but static teacher. In previous studies, DML was used on simple multi-class classification problems, such as image classification [25]–[28]. However, no studies have tried DML on more complex sequence-to-sequence mapping problems.

This paper presents a method to incorporate DML in state-of-the-art Transformer-based end-to-end ASR models. In particular, we propose to combine DML with the existing training techniques for end-to-end ASR models. DML is closely related to label smoothing [16]; both aim to prevent models from making over-confident predictions. While label smoothing uses a uniform distribution to smooth the ground-truth distribution, DML leverages the distributions predicted by other student models. Combining both kinds of smoothing should efficiently prevent over-confident predictions. In addition, DML is related to scheduled sampling [17] and SpecAugment [18], [19]. While scheduled sampling and SpecAugment aim to maintain consistency between similar conditioning contexts, DML aims to maintain consistency between different student models. We expect that these consistency strategies complement each other.

Our experiments using the Corpus of Spontaneous Japanese (CSJ) [29] examined two experimental setups: large-scale modeling and compact modeling. We found that DML improves the ASR performance of both modeling setups compared with conventional learning methods, including knowledge distillation. We also found that combining DML with the existing training techniques effectively improves ASR performance.

II. End-to-End ASR with Transformer

This section briefly describes end-to-end ASR that uses Transformer-based encoder-decoder models based on auto-regressive generative modeling [10]–[14], [30]. The encoder-decoder models predict a generation probability of a text $\boldsymbol{W} = \{w_1, \cdots, w_N\}$ given speech $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M\}$, where w_n is the n-th token in the text and $\boldsymbol{x}_m$ is the m-th acoustic feature in the speech. N is the number of tokens in the text and M is the number of acoustic features in the speech. In the auto-regressive generative models, the generation probability of $\boldsymbol{W}$ is defined as

$$P(\boldsymbol{W}|\boldsymbol{X};\boldsymbol{\Theta}) = \prod_{n=1}^{N} P(w_n|\boldsymbol{W}_{1:n-1}, \boldsymbol{X};\boldsymbol{\Theta}), \quad (1)$$

where $\boldsymbol{\Theta}$ represents the trainable model parameter sets and $\boldsymbol{W}_{1:n-1} = \{w_1, \cdots, w_{n-1}\}$. In our Transformer-based end-to-end ASR models, $P(w_n|\boldsymbol{W}_{1:n-1}\ \boldsymbol{X};\boldsymbol{\Theta})$ is computed using a speech encoder and a text decoder, both of which are composed of a couple of Transformer blocks.

A. Network structure

Speech encoder: The speech encoder converts input acoustic features $\boldsymbol{X}$ into the hidden representations $\boldsymbol{H}^{(I)}$ using I Transformer encoder blocks. The i-th Transformer encoder block composes i-th hidden representations $\boldsymbol{H}^{(i)}$ from the lower layer inputs $\boldsymbol{H}^{(i-1)}$ as

$$\boldsymbol{H}^{(i)} = \texttt{TransformerEncoderBlock}(\boldsymbol{H}^{(i-1)};\boldsymbol{\Theta}), \quad (2)$$

where $\texttt{TransformerEncoderBlock}()$ is a Transformer encoder block that consists of a scaled dot product multi-head self-attention layer and a position-wise feed-forward network [10]. The hidden representations $\boldsymbol{H}^{(0)} = \{\boldsymbol{h}_1^{(0)}, \cdots, \boldsymbol{h}_{M'}^{(0)}\}$ are produced by

$$\boldsymbol{h}_{m'}^{(0)} = \texttt{AddPostionalEncoding}(\boldsymbol{h}_{m'}), \quad (3)$$

where $\texttt{AddPositionalEncoding}()$ is a function that adds a continuous vector in which position information is embedded. $\boldsymbol{H} = \{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{M'}\}$ is produced by

$$\boldsymbol{H} = \texttt{ConvolutionPooling}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M;\boldsymbol{\Theta}), \quad (4)$$

where $\texttt{ConvolutionPooling}()$ is a function composed of convolution layers and pooling layers. M' is the subsampled sequence length depending on the function.

Text decoder: The text decoder computes the generative probability of a token from preceding tokens and the hidden representations of the speech. The predicted probabilities of the n-th token w_n are calculated as

$$P(w_n|\boldsymbol{W}_{1:n-1}, \boldsymbol{X};\boldsymbol{\Theta}) = \texttt{Softmax}(\boldsymbol{u}_{n-1}^{(J)};\boldsymbol{\Theta}), \quad (5)$$

where $\texttt{Softmax}()$ is a softmax layer with a linear transformation. The input hidden vector $\boldsymbol{u}_{n-1}^{(J)}$ is computed from J Transformer decoder blocks. The j-th Transformer decoder block composes j-th hidden representation $\boldsymbol{u}_{n-1}^{(j)}$ from the lower layer inputs $\boldsymbol{U}_{1:n-1}^{(j-1)} = \{\boldsymbol{u}_1^{(j-1)}, \cdots, \boldsymbol{u}_{n-1}^{(j-1)}\}$ as

$$\boldsymbol{u}_{n-1}^{(j)} = \texttt{TransformerDecoderBlock}(\boldsymbol{U}_{1:n-1}^{(j-1)}, \boldsymbol{H}^{(I)};\boldsymbol{\Theta}), \quad (6)$$

where $\texttt{TransformerDecoderBlock}()$ is a Transformer decoder block that consists of a scaled dot product multi-head masked self-attention layer, a scaled dot product multi-head source-target attention layer, and a position-wise feed-forward network [10]. The hidden representations $\boldsymbol{U}_{1:n-1}^{(0)} = \{\boldsymbol{u}_1^{(0)}, \cdots, \boldsymbol{u}_{n-1}^{(0)}\}$ are produced by

$$\boldsymbol{u}_{n-1}^{(0)} = \texttt{AddPositionalEncoding}(\boldsymbol{w}_{n-1}), \quad (7)$$

$$\boldsymbol{w}_{n-1} = \texttt{Embedding}(w_{n-1};\boldsymbol{\Theta}), \quad (8)$$

where $\texttt{Embedding}()$ is a linear layer that embeds input token in a continuous vector.

B. Typical objective function

In end-to-end ASR, a model parameter set can be optimized from the utterance-level training data set $\mathcal{U} = \{(\boldsymbol{X}^1, \boldsymbol{W}^1), \cdots, (\boldsymbol{X}^T, \boldsymbol{W}^T)\}$, where T is the number of utterances in the training data set. An objective function based on the maximum likelihood estimation is defined as

$$\mathcal{L}_{\texttt{mle}}(\boldsymbol{\Theta}) = -\sum_{t=1}^{T}\sum_{n=1}^{N^t}\sum_{w_n^t \in \mathcal{V}} \hat{P}(w_n^t|\boldsymbol{W}_{1:n-1}^t, \boldsymbol{X}^t) \log P(w_n^t|\boldsymbol{W}_{1:n-1}^t, \boldsymbol{X}^t;\boldsymbol{\Theta}), \quad (9)$$

where w_n^t is the n-th token for the t-th utterance and $\boldsymbol{W}_{1:n-1}^t = \{w_1^t, \cdots, w_{n-1}^t\}$. $\mathcal{V}$ represents the vocabulary sets, and N^t is the number of tokens in the t-th utterance. The ground-truth probability $\hat{P}(w_n^t|\boldsymbol{W}_{1:n-1}^t, \boldsymbol{X}^t)$ is defined as

$$\hat{P}(w_n^t|\boldsymbol{W}_{1:n-1}^t, \boldsymbol{X}^t) = \begin{cases} 1 & (w_n^t = \hat{w}_n^t) \\ 0 & (w_n^t \neq \hat{w}_n^t), \end{cases} \quad (10)$$

where $\hat{w}_n^t$ is the n-th reference token in the t-th utterance.

C. Training techniques

There are several training techniques in the end-to-end ASR modeling. This paper introduces the following three techniques.

Label smoothing: Label smoothing is a regularization technique that can prevent the model from making over-confident predictions [16]. This encourages the model to have higher entropy at its prediction. This paper introduces a uniform distribution to all tokens in vocabulary that smooths the ground-truth probabilities. Thus, an objective function that uses the label smoothing is defined as

$$\mathcal{L}_{\texttt{ls}}(\boldsymbol{\Theta}) = -\sum_{t=1}^{T}\sum_{n=1}^{N^k}\sum_{w_n^t \in \mathcal{V}} \tilde{P}(w_n^t|\boldsymbol{W}_{1:n-1}^t, \boldsymbol{X}^t) \log P(w_n^t|\boldsymbol{W}_{1:n-1}^t, \boldsymbol{X}^t;\boldsymbol{\Theta}), \quad (11)$$

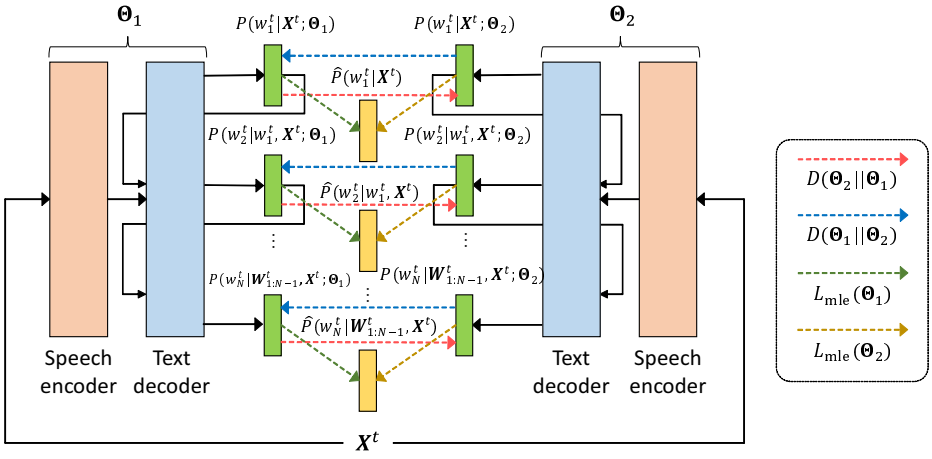

Fig. 1. Deep mutual learning using two student models.

$$\tilde{P}(w_n^t|w_1^t,\cdots,w_{n-1}^t,\boldsymbol{X}^t) = (1-\alpha)\hat{P}(w_n^t|\boldsymbol{W}_{1:n-1}^t,\boldsymbol{X}^t)+\alpha\frac{1}{|\mathcal{V}|}, \quad (12)$$

where α is a smoothing weight to adjust the smoothing term.

Scheduled sampling: Scheduled sampling is a technique that randomly uses predicted tokens as conditioning tokens in the text decoder [17]. This technique helps reduce the gap between teacher forcing in a training phase and free running in a testing phase. An objective function that uses scheduled sampling is defined as

$$\mathcal{L}_{\text{ss}}(\boldsymbol{\Theta}) = -\sum_{t=1}^{T}\sum_{n=1}^{N^k}\sum_{w_n^t\in\mathcal{V}}\hat{P}(w_n^t|\boldsymbol{W}_{1:n-1}^t,\boldsymbol{X}^t) \log P(w_n^t|\mathcal{S}(\boldsymbol{W}_{1:n-1}^t),\boldsymbol{X}^t;\boldsymbol{\Theta}), \quad (13)$$

where $\mathcal{S}()$ is a scheduled sampling function with random behavior for the conditioning tokens.

SpecAugment: SpecAugment is a technique that augments input acoustic feature representations [18], [19]. This technique consists of three kinds of deformations: time warping, time masking, and frequency masking. Time warping is a deformation of the acoustic features in the time direction. Time masking and frequency masking mask a block of consecutive time steps or frequency channels. An objective function that uses SpecAugment is defined as

$$\mathcal{L}_{\text{sa}}(\boldsymbol{\Theta}) = -\sum_{t=1}^{T}\sum_{n=1}^{N^k}\sum_{w_n^t\in\mathcal{V}}\hat{P}(w_n^t|\boldsymbol{W}_{1:n-1}^t,\boldsymbol{X}^t) \log P(w_n^t|\boldsymbol{W}_{1:n-1}^t,\mathcal{G}(\boldsymbol{X}^t);\boldsymbol{\Theta}), \quad (14)$$

where $\mathcal{G}()$ is the SpecAugment deformation function with random behavior for the input acoustic features.

III. Proposed Method

This section details deep mutual learning (DML) for end-to-end ASR. In addition, we present objective functions when combining DML with several training techniques.

A. Deep mutual learning for end-to-end ASR

In DML, K different model parameters $\{\boldsymbol{\Theta}_1,\cdots,\boldsymbol{\Theta}_K\}$ are simultaneously trained to mimic each other, while the conventional training method learns the model parameters to predict ground-truth probabilities for the training instances. Figure 1 represents DML using two student model parameters. A DML-based objective function for training the k-th model parameter $\boldsymbol{\Theta}_k$ is defined as

$$\mathcal{L}_{\text{dml}}(\boldsymbol{\Theta}_k) = (1-\lambda)\mathcal{L}_{\text{mle}}(\boldsymbol{\Theta}_k)+\lambda\frac{1}{K-1}\sum_{i=1,i\neq k}^{K}\mathcal{D}(\boldsymbol{\Theta}_i||\boldsymbol{\Theta}_k), \quad (15)$$

where $D(\boldsymbol{\Theta}_i||\boldsymbol{\Theta}_k)$ is a mimicry loss to mimic the i-th model, and λ is an interpolation weight to adjust the influence of the mimicry loss. The mimicry loss is computed from

$$\mathcal{D}(\boldsymbol{\Theta}_i||\boldsymbol{\Theta}_k) = -\sum_{t=1}^{T}\sum_{n=1}^{N^k}\sum_{w_n^t\in\mathcal{V}}P(w_n^t|\boldsymbol{W}_{1:n-1}^t,\boldsymbol{X}^t;\boldsymbol{\Theta}_i) \log P(w_n^t|\boldsymbol{W}_{1:n-1}^t,\boldsymbol{X}^t;\boldsymbol{\Theta}_k). \quad (16)$$

In a mini-batch training, K model parameters are optimized jointly and collaboratively. Thus, K models are learned with the same mini-batches. In each mini-batch step, we compute predicted probability distributions using the K models and update each parameter according to the predicted probability distributions of the others. These optimizations are conducted iteratively until convergence. We finally pick up the single model with the smallest validation loss or a pre-defined compact model.

B. Deep mutual learning with training techniques

DML can be combined with existing training techniques for end-to-end ASR. This paper proposes new objective functions specific to using DML with label smoothing, scheduled sampling, and SpecAugment. Note that all techniques can be simultaneously combined with DML.

Deep mutual learning with label smoothing: Both label smoothing and DML avoid peaky predictions with very low entropy. When combining label smoothing with DML, we define an objective function that trains k-th model parameter $\boldsymbol{\Theta}_k$ as

$$\mathcal{L}_{\texttt{dml+ls}}(\boldsymbol{\Theta}_k) = (1-\lambda)\mathcal{L}_{\texttt{ls}}(\boldsymbol{\Theta}_k)+\lambda\frac{1}{K-1}\sum_{i=1,i\neq k}^{K}\mathcal{D}(\boldsymbol{\Theta}_i||\boldsymbol{\Theta}_k), \tag{17}$$

where $\mathcal{L}_{\texttt{ls}}$ is the same as Eq. (11).

Deep mutual learning with scheduled sampling: When combining scheduled sampling with DML, we aim to make model more robust to various conditioning tokens by maintaining consistency between different models with different conditioning contexts. Thus, an objective function for the k-th model parameter is defined as

$$\mathcal{L}_{\texttt{dml+ss}}(\boldsymbol{\Theta}_k) = (1-\lambda)\mathcal{L}_{\texttt{ss}}(\boldsymbol{\Theta}_k) + \lambda\frac{1}{K-1}\sum_{i=1,i\neq k}^{K}\mathcal{D}_{\texttt{ss}}(\boldsymbol{\Theta}_i||\boldsymbol{\Theta}_k), \tag{18}$$

$$\mathcal{D}_{\texttt{ss}}(\boldsymbol{\Theta}_i||\boldsymbol{\Theta}_k) = -\sum_{t=1}^{T}\sum_{n=1}^{N^k}\sum_{w_n^t\in\mathcal{V}} P(w_n^t|\mathcal{S}_i(\boldsymbol{W}_{1:n-1}^t),\boldsymbol{X}^t;\boldsymbol{\Theta}_i) \log P(w_n^t|\mathcal{S}_k(\boldsymbol{W}_{1:n-1}^t),\boldsymbol{X}^t;\boldsymbol{\Theta}_k), \tag{19}$$

where $\mathcal{L}_{\texttt{ss}}$ is the same as Eq. (13). $\mathcal{S}_i()$ and $\mathcal{S}_k()$ are the functions for the scheduled sampling with different random seeds.

Deep mutual learning with SpecAugment: When combining SpecAugment with DML, we aim to make model more robust to various acoustic feature examples by maintaining consistency between different models with different deformation. An objective function for the k-th model parameter is defined as

$$\mathcal{L}_{\texttt{dml+sa}}(\boldsymbol{\Theta}_k) = (1-\lambda)\mathcal{L}_{\texttt{sa}}(\boldsymbol{\Theta}_k) + \lambda\frac{1}{K-1}\sum_{i=1,i\neq k}^{K}\mathcal{D}_{\texttt{sa}}(\boldsymbol{\Theta}_i||\boldsymbol{\Theta}_k), \tag{20}$$

$$\mathcal{D}_{\texttt{sa}}(\boldsymbol{\Theta}_i||\boldsymbol{\Theta}_k) = -\sum_{t=1}^{T}\sum_{n=1}^{N^k}\sum_{w_n^t\in\mathcal{V}} P(w_n^t|\boldsymbol{W}_{1:n-1}^t,\mathcal{G}_i(\boldsymbol{X}^t);\boldsymbol{\Theta}_i) \log P(w_n^t|\boldsymbol{W}_{1:n-1}^t,\mathcal{G}_k(\boldsymbol{X}^t);\boldsymbol{\Theta}_k), \tag{21}$$

where $\mathcal{L}_{\texttt{sa}}$ is the same as Eq. (14). $\mathcal{G}_i()$ and $\mathcal{G}_k()$ are the functions for the SpecAugment with different random seeds.

IV. Experiments

We experimented using CSJ [29]. We divided the CSJ into a training set (512.6 hours), a validation set (4.8 hours), and three test sets (1.8 hours, 1.9 hours, and 1.3 hours). We used the validation set to choose several hyper parameters and to conduct early stopping. Each discourse-level speech was segmented into utterances in accordance with our previous work [31]. We used characters as the tokens.

A. Setups

We examined two types of experimental setups: large-scale modeling and compact modeling.

- **Large-scale modeling**: We set $I = 8$ for the encoder blocks and $J = 6$ for the decoder blocks. When introducing DML, we prepared 4 large-scale models and evaluated the single model with the least validation loss.
- **Compact modeling**: We set $I = 2$ for the encoder blocks and $J = 1$ for the decoder blocks where other parameters were the same as the large-scale modeling. When introducing the knowledge distillation [21] or the deep mutual learning, we prepared 1 compact model and 3 large-scale models and evaluated the compact model.

In both setups, Transformer blocks were composed using the following conditions: the dimensions of the output continuous representations were set to 256, the dimensions of the inner outputs in the position-wise feed forward networks were set to 2,048, and the number of heads in the multi-head attentions was set to 4. For the speech encoder, we used 40 log mel-scale filterbank coefficients appended with delta and acceleration coefficients as acoustic features. The frame shift was 10 ms. The acoustic features passed two convolution and max pooling layers with a stride of 2, so we downsampled them to $1/4$ along with the time-axis. In the text decoder, we used 256-dimensional word embeddings. We set the vocabulary size to 3,262.

For the optimization, we used the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$ and varied the learning rate based on the update rule presented in previous studies [10]. The training steps were stopped based on early stopping using the validation set. We set the mini-batch size to 32 utterances and the dropout rate in the Transformer blocks to 0.1. When we introduced label smoothing, we set α as 0.1. Our scheduled sampling-based optimization process used the teacher forcing at the beginning of the training steps, and we linearly ramped up the probability of sampling to the specified probability at the specified epoch (20 epoch). Our SpecAugment only applied frequency masking and time masking where the number of frequency masks and time step masks were set to 2, the frequency masking width was randomly chosen from 0 to 20 frequency bins, and the time masking width was randomly chosen from 0 to 100 frames. λ was set to 0.4 in DML. We used a beam search algorithm in which the beam size was set to 20.

B. Results

We evaluated various setups using DML and the training techniques in large-scale modeling and compact modeling setups. Table 1 shows experimental results in terms of character error rate.

First, in the large-scale modeling setup, the results show that each training tip improves Transformer-based end-to-end ASR performance, and combining the techniques effectively improved ASR performance. SpecAugment significantly improved performance in particular. These results indicate

TABLE I
EXPERIMENTAL RESULTS IN TERMS OF CHARACTER ERROR RATE (%).

	Label smoothing	Scheduled sampling	SpecAugment	Knowledge distillation	Deep mutual learning (DML)	Test 1	Test 2	Test 3
Large-scale modeling	-	-	-	-	-	8.83	6.49	7.19
	√	-	-	-	-	8.41	6.23	6.74
	-	√	-	-	-	8.59	6.31	6.30
	-	-	√	-	-	7.48	5.59	5.86
	√	√	√	-	-	7.24	5.13	5.40
	-	-	-	-	√	8.19	5.78	6.37
	√	-	-	-	√	8.05	5.67	6.30
	-	√	-	-	√	7.90	5.57	5.62
	-	-	√	-	√	7.02	4.92	5.28
	√	√	√	-	√	**6.87**	**4.73**	**5.02**
Compact modeling	-	-	-	-	-	12.80	9.43	10.01
	√	√	√	-	-	11.37	7.88	8.44
	-	-	-	√	-	11.67	8.28	9.08
	√	√	√	√	-	11.15	7.58	8.31
	-	-	-	-	√	11.23	7.82	8.74
	√	√	√	-	√	**10.65**	**7.19**	**7.93**

that training techniques are important for the Transformer-based end-to-end ASR models. In addition, we improved ASR performance by introducing DML into the Transformer-based end-to-end ASR models, both with and without training techniques. It is thought that DML could help discover the global optimum and prevent models from making over-confident predictions. The highest results were attained by combining DML and all the training techniques. This suggests that combining DML with existing training techniques effectively improves ASR performance.

Next, in the compact modeling setups, the results show that DML improved performance even more than knowledge distillation. This indicates that DML in which student models interact with each other during all the training steps effectively transfers knowledge in large-scale end-to-end ASR models to the compact end-to-end ASR models. These results confirm that DML is a good solution to build Transformer-based end-to-end ASR models.

V. CONCLUSIONS

We have presented a method to incorporate deep mutual learning (DML) in Transformer-based end-to-end automatic speech recognition models. The key advance of our method is to introduce combined training strategies of DML with representative training techniques (label smoothing, scheduled sampling, and SpecAugment) for end-to-end ASR models. Our experiments demonstrated that the DML improves ASR performance of both large-scale modeling and compact modeling setups compared with conventional learning methods, including knowledge distillation. We also showed that combining DML with existing training techniques effectively improves ASR performance.

REFERENCES

[1] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4805–4809, 2017.

[2] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for English conversational speech recognition," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 959–963, 2017.

[3] H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1298–1302, 2017.

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

[5] 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, 2015.

[6] L. Lu, X. Zhang, K. Cho, and S. Renals, "A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 3249–3253, 2015.

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

[8] L. Lu, X. Zhang, and S. Renals, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 5060–5064, 2016.

[9] R. Masumura, T. Tanaka, T. Moriya, Y. Shinohara, T. Oba, and Y. Aono, "Large context end-to-end automatic speech recognition via extension of hierarchical recurrent encoder-decoder models," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 5661–5665, 2019.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *In Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 5998–6008, 2017.

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

[12] S. Zhou, L. Dong, S. Xu, and B. Xu, "Syllable-based sequence-to-sequence speech recognition with the Transformer in mandarin chinese," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 791–795, 2018.

[13] Y. Zhao, J. Li, X. Wang, and Y. Li, "The SpeechTransformer for large-scale mandarin chinese speech recognition," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 7095–7099, 2019.

[14] S. Li, D. Raj, X. Lu, P. Shen, T. Kawahara, and H. Kawai, "Improving

Transformer-based speech recognition systems with compressed structure and speech attribute augmentation," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 4400–4404, 2019.

[15] R. Masumura, N. Makishima, M. Ihori, A. Takashima, T. Tanaka, and S. Orihashi, "Phoneme-to-grapheme conversion based large-scale pre-training for end-to-end automatic speech recognition," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2822–2826, 2020.

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

[17] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *In Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 1171–1179, 2015.

[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 Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2613–2617, 2019.

[19] D. S. Park, Y. Zhang, C.-C. Chiu, Y. Chen, B. Li, W. Chan, Q. V. Le, and Y. Wu, "Spacaugment on large scale datasets," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6874–6878, 2020.

[20] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4774–4778, 2018.

[21] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *In Proc. NIPS Deep Learning and Representation Learning Workshop*, 2015.

[22] M. Huang, Y. You, Z. Chen, Y. Qian, and K. Yu, "Knowledge distillation for sequence model," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 3703–3707, 2018.

[23] H.-G. Kim, H. Na, H. Lee, J. Lee, T. G. Kang, M.-J. Lee, and Y. S. Choi, "Knowledge distillation using output errors for self-attention end-to-end models," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6181–6185, 2019.

[24] 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 Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 7049–7053, 2020.

[25] Y. Zhang, T. Xiang, T. M. Hospedales, , and H. Lu, "Deep mutual learning," *In Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4320–4328, 2018.

[26] H. Zhao, G. Yang, D. Wang, and H. Lu, "Lightweight deep neural network for real-time visual tracking with mutual learning," *In Proc. IEEE International Conference on Image Processing (ICIP)*, pp. 3063–3067, 2019.

[27] R. Wu, M. Feng, W. Guan, D. Wang, H. Lu, and E. Ding, "A mutual learning method for salient object detection with intertwined multi-supervision," *In Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8150–8159, 2019.

[28] F. M. Thoker and J. Gall, "Cross-modal knowledge distillation for action recognition," *IEEE International Conference on Image Processing (ICIP)*, pp. 6–10, 2019.

[29] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," *In Proc. International Conference on Language Resources and Evaluation (LREC)*, pp. 947–952, 2000.

[30] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving Transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1408–1412, 2019.

[31] R. Masumura, H. Sato, T. Tanaka, T. Moriya, Y. Ijima, and T. Oba, "End-to-end automatic speech recognition with a reconstruction criterion using speech-to-text and text-to-speech encoder-decoders," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1606–1610, 2019.