Semi-Supervised End-to-End Speech-to-Text Translation with Joint Text-to-Text and Speech-to-Text Decoding

Tomohiro Tanaka, Ryo Masumura, Naoki Makishima, Mana Ihori, Shota Orihashi, Satoshi Suzuki, Taiga Yamane
NTT, Inc., Japan
E-mail: tomohiro.tanaka@ntt.com

***Abstract*—This paper proposes a novel semi-supervised end-to-end speech-to-text translation (S2TT) approach that leverages pseudo-labeling (PL) through joint decoding with both a text-to-text translation (T2TT) model and an S2TT model. While pseudo-labeling has proven effective in mitigating the data scarcity issue in S2TT, conventional PL approaches typically rely either on a standalone S2TT model or a cascade system combining automatic speech recognition (ASR) with a T2TT model. However, cascade systems trained on out-of-domain data often perform poorly on target domain inputs, and S2TT models fine-tuned on limited in-domain data tend to struggle with low-occurrence tokens. To address these limitations, our method employs both the S2TT model and the cascade system for PL, using the T2TT model to complement the S2TT model by improving accuracy on rare tokens. Experimental results demonstrate that our proposed method outperforms conventional PL strategies that utilize only a T2TT or S2TT model.**

I. Introduction

Translation is a key technology for facilitating human-to-human multilingual communication. Text-to-text translation (T2TT), which converts source language text into target language text, has been widely studied. When speech is used as input, a combination of automatic speech recognition (ASR) and T2TT can be used to convert source language speech into target language text. End-to-end speech-to-text translation (S2TT), which directly translates source language speech into target language text, has gained attention as a simpler and faster alternative to the ASR-T2TT pipeline. However, building an end-to-end S2TT model requires a large amount of data, and a major challenge is the limited availability of data compared with ASR and T2TT.

The data scarcity problem in end-to-end S2TT has been addressed in previous studies through approaches such as data augmentation for speech [1], [2], multi-task learning with ASR or T2TT [3], [4], pre-training using unlabeled speech or text data [5], [6], knowledge distillation from T2TT [7], [8], multilingual speech translation [9], [10], and back-translation from target language text to discrete speech units [11]. This paper focuses on pseudo-labeling (PL) [12], [13] which enables the use of unlabeled speech dataset, trained ASR and T2TT models.

PL, often called self-training, is a method used to address the data scarcity problem by taking advantage of large amounts of unlabeled data. PL is a straightforward process. An initial model is first trained using a limited amount of labeled data. The trained model is then used to predict labels for the unlabeled data, with the predicted labels referred to as pseudo-labels. Finally, the pseudo-labels are combined with the initial labeled data to train a new model.

PL has been shown to be effective in improving performance for several speech and natural language processing tasks, such as T2TT [14]–[16], end-to-end ASR [17]–[20], end-to-end S2TT [12], [13], and speech-to-speech translation [21]. In S2TT, accuracy improvement has been confirmed by reconstructing the S2TT model by assigning pseudo-labels to unlabeled speech data using either S2TT or T2TT in a cascade system, but not both of them [12], [13]. However, a S2TT model overfits to a small amount of labeled data, which degrades the performance for tokens that occur infrequently even within a target domain. It has been observed that the accuracy of a S2TT model decreases for a set of sentences with low-frequency tokens, as shown in Table II.

To address this problem, we propose a semi-supervised end-to-end S2TT method with a powerful PL that uses both end-to-end S2TT and T2TT in cascade system with ASR. Our proposed method enables highly accurate PL even when the amount of labeled data is small. To combine end-to-end S2TT and T2TT, we use shallow fusion [22], which integrates the posterior probabilities during decoding. It is expected that T2TT, which is robust to a wider domain, can help address low-frequency tokens in training data. We consider the case in which ASR and T2TT models are available, because training data for these models is easier to collect than for S2TT, as mentioned above. We also assume that a small amount of labeled speech data and a large amount of unlabeled speech data are available as training data.

We verified the effectiveness of our proposed method through an English to Japanese S2TT task. We show the effectiveness of joint decoding with end-to-end S2TT and T2TT models. Furthermore, the results show that PL with joint decoding outperforms conventional methods that use either the end-to-end S2TT or T2TT model for PL.

II. Related Work

Our study is related to semi-supervised learning for end-to-end ASR [17]–[20]. In semi-supervised end-to-end ASR,

979-8-3315-7206-8/25/$31.00

powerful language models are often incorporated via shallow fusion to improve the quality of pseudo-labels. In contrast, our work utilizes a strong machine translation model to generate high-quality pseudo-labels, aiming to improve the performance of end-to-end S2TT.

Our proposed method is related to label smoothing [23] with which a slightly reduced probability of the correct class is assigned to prevent models from making over-confident predictions. Training on pseudo-labeled data is the same as training with noise on the prediction target. Therefore, learning with PL is also expected to be effective in avoiding over-confident predictions. Our proposed method is also related to scheduled sampling [24]. Scheduled sampling uses the model's predictions as decoder inputs during training to maintain consistency, while semi-supervised learning with PL also uses the model's predictions as target labels for training. Therefore, PL is expected to be able to maintain consistency as well as scheduled sampling.

III. Semi-supervised End-to-End S2TT

A. Strategy and Settings

We adopt a semi-supervised learning approach to effectively utilize a small labeled dataset and a large unlabeled dataset. Figure 1 illustrates the overall flow of the proposed method. In our setup, pre-trained ASR and T2TT models are assumed to be available. We aim to build an S2TT model using pseudo-labeling (PL), given labeled speech data $\mathcal{D}_{\mathrm{l}}$ and unlabeled speech data $\mathcal{D}_{\mathrm{u}}$. Our method converts the unlabeled dataset into a pseudo-labeled dataset. The final model is trained on both the labeled and pseudo-labeled datasets. For PL, we introduce joint decoding with T2TT and S2TT models using shallow fusion. This enables the generation of higher-quality pseudo labels compared to using either model individually.

B. Modeling

1) Cascade system of ASR and T2TT: We construct a cascade system with trained transformer-based auto-regressive ASR and the T2TT models.

ASR model: The encoder converts the input acoustic features $\boldsymbol{X}$ into hidden representations $\boldsymbol{f}$ by using a transformer encoder. The output of the transformer encoder $\boldsymbol{f}$ is obtaind as

$$\boldsymbol{f} = \texttt{TransformerEncoder}(\boldsymbol{X}; \boldsymbol{\theta}_f), \tag{1}$$

where $\texttt{TransformerEncoder}(\cdot)$ is a transformer encoder including a scaled dot-product multi-head self-attention layer and a position-wise feed-forward network and $\boldsymbol{\theta}_f$ is the parameter. The hidden representations $\boldsymbol{f}$ from the encoder are fed into the transformer decoder. The decoder also receive the embeddings of predicted tokens in decoder. When the decoder output is $\boldsymbol{Y} = \{y_1, \cdots, y_t, \cdots y_T\}$ and the first input of the decoder is the start of sentence as $y_0 = <\texttt{s}>$, the token embeddings are calculated as

$$\boldsymbol{e}_t = \texttt{Embedding}(y_t; \boldsymbol{\theta}_e), \tag{2}$$

where $\texttt{Embedding}(\cdot)$ is a function that converts a token into a continuous representation and $\boldsymbol{\theta}_e$ is the parameter. When the output of the t-th time step for the transformer in the decoder is $\boldsymbol{g}_t$, the transformer decoder constructs a hidden representation from the token embeddings in previous time steps $\boldsymbol{e}_{0:t-1}$ and encoder output $\boldsymbol{f}$ as

$$\boldsymbol{g}_t = \texttt{TransformerDecoder}(\boldsymbol{e}_{0:t-1}, \boldsymbol{f}; \boldsymbol{\theta}_g), \tag{3}$$

where $\texttt{TransformerDecoder}(\cdot)$ 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 $\boldsymbol{\theta}_g$ is the parameter. Finally, the network estimates the probabilities of a distribution of the output tokens P_{ASR} as

$$P_{\mathrm{ASR}}(y_t | y_{1:t-1}, \boldsymbol{X}; \boldsymbol{\Theta}_{\mathrm{ASR}}) = \texttt{Softmax}(\boldsymbol{g}_t; \boldsymbol{\theta}_o), \tag{4}$$

where $\texttt{Softmax}(\cdot)$ represents the softmax function with linear Transformation and $\boldsymbol{\theta}_o$ is the parameter. The ASR result $\hat{\boldsymbol{Y}} = \{\hat{y}_1, \cdots, \hat{y}_t, \cdots\}$ is obtained by beam search decoding as

$$\hat{\boldsymbol{Y}} = \arg\max_{\boldsymbol{Y}} P_{\mathrm{ASR}}(\boldsymbol{Y} | \boldsymbol{X}, \Theta_{\mathrm{ASR}}). \tag{5}$$

T2TT model: The ASR result is input into the T2TT model. The token embeddings of the ASR result $\overline{\boldsymbol{Y}} = \{\overline{y}_1, \cdots \overline{y}_t, \cdots\}$ are embedded as

$$\overline{y}_t = \texttt{Embedding}(\hat{y}_t; \boldsymbol{\theta}_e), \tag{6}$$

where $\boldsymbol{\theta}_e$ is the parameter. Similar to the ASR model, the T2TT model calculates the output probability of tokens. When the output of the decoder is $\boldsymbol{Z} = \{z_1, \cdots, z_t, \cdots, z_T\}$ and the first input of the decoder is the start of sentence as $z_0 = <\texttt{s}>$, the probabilities of a distribution of the output tokens P_{T2TT} are calculated as

$$\boldsymbol{f}' = \texttt{TransformerEncoder}(\overline{\boldsymbol{Y}}; \boldsymbol{\theta}'_g), \tag{7}$$

$$\boldsymbol{e}'_t = \texttt{Embedding}(z_t; \boldsymbol{\theta}'_e), \tag{8}$$

$$\boldsymbol{g}'_t = \texttt{TransformerDecoder}(\boldsymbol{e}'_{0:t-1}; \boldsymbol{f}', \boldsymbol{\theta}'_g), \tag{9}$$

$$P_{\mathrm{T2TT}}(z_t | z_{1:t-1}, \overline{\boldsymbol{Y}}; \boldsymbol{\Theta}_{\mathrm{T2TT}}) = \texttt{Softmax}(\boldsymbol{g}'_t; \boldsymbol{\theta}'_o). \tag{10}$$

The translation result $\hat{\boldsymbol{Z}}$ is obtained by beam search decoding as

$$\hat{\boldsymbol{Z}} = \arg\max_{\boldsymbol{Z}} P_{\mathrm{T2TT}}(\boldsymbol{Z} | \overline{\boldsymbol{Y}}; \boldsymbol{\Theta}_{\mathrm{T2TT}}). \tag{11}$$

2) S2TT model: The S2TT model consists of three components: a speech encoder, connector, and text decoder. **Speech encoder:** The speech encoder converts the input acoustic features $\boldsymbol{X}$ into hidden representations $\boldsymbol{s}$ by using the transformer encoder. The output of the transformer encoder $\boldsymbol{s}$ is obtained as

$$\boldsymbol{s} = \texttt{TransformerEncoder}(\boldsymbol{X}; \boldsymbol{\theta}_s), \tag{12}$$

where $\boldsymbol{\theta}_s$ is the parameter.

Connector: We employed a connector consisting of 1D convolutions and a transformer encoder [25]. The outputs

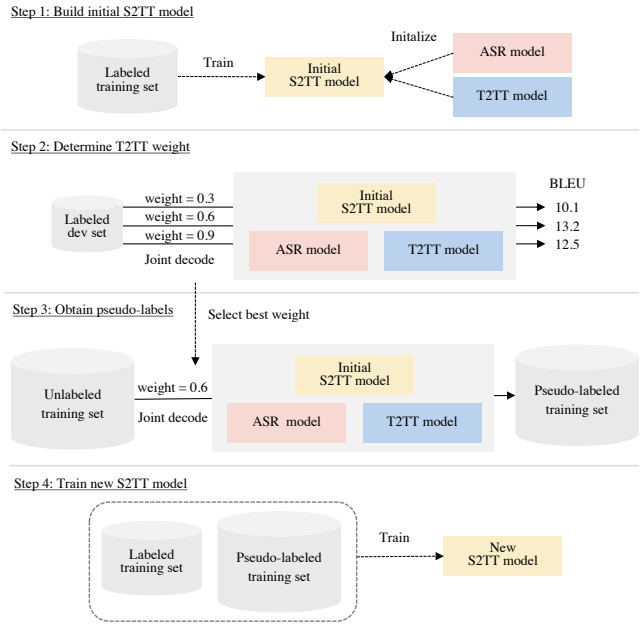

Fig. 1. Flow of our semi-supervised speech-to-text translation.

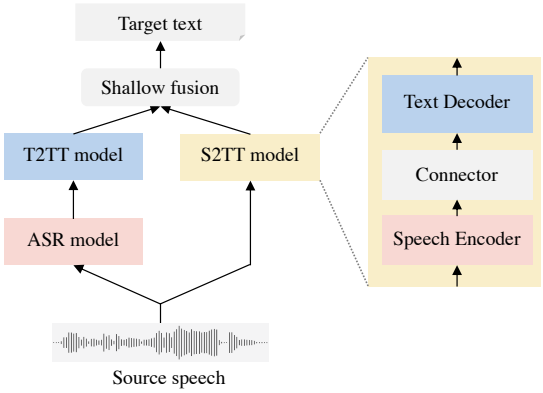

Fig. 2. Joint T2TT and S2TT decoding with shallow fusion.

of the speech encoder are input to multi-layer stack of 1D convolutions to reduce the input speech length. The output is subsequently connected to a multi-layer transformer as

$$\boldsymbol{d} = \texttt{Convolution1D}(\boldsymbol{s}; \boldsymbol{\theta}_d), \tag{13}$$

$$\boldsymbol{h} = \texttt{TransformerEncoder}(\boldsymbol{d}; \boldsymbol{\theta}_h), \tag{14}$$

where $\texttt{Convolution1D}(\cdot)$ is a multi-layer stack of 1D convolutions with max pooling, and $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_h$ is the parameter.

Text decoder: When the output of the text decoder is $\boldsymbol{Z} = \{z_1, \cdots, z_t, \cdots\}$ and the first input of the decoder is the start of sentence as $z_0 = <\texttt{s}>$, the token embeddings are calculated as

$$\boldsymbol{u}_t = \texttt{Embedding}(z_t; \boldsymbol{\theta}_u), \tag{15}$$

where $\boldsymbol{\theta}_u$ is the parameter. The output of the connector and token embeddings are input to the transformer decoder as

$$\boldsymbol{v}_t = \texttt{TransformerDecoder}(\boldsymbol{u}_{0:t-1}, \boldsymbol{h}; \boldsymbol{\theta}_v), \tag{16}$$

where $\boldsymbol{\theta}_d$ is the parameter. The probabilities of a distribution of the output tokens P_{T2TT} are calculated as

$$P_{\text{S2TT}}(z_t|z_{1:t-1}, \boldsymbol{X}; \boldsymbol{\Theta}_{\text{S2TT}}) = \texttt{Softmax}(\boldsymbol{v}_t; \boldsymbol{\theta}_o), \tag{17}$$

where $\boldsymbol{\theta}_o$ is the parameter. The translation result $\hat{\boldsymbol{Z}}$ is obtained by beam search decoding as

$$\hat{\boldsymbol{Z}} = \arg\max_{\boldsymbol{Z}} P_{\text{S2TT}}(\boldsymbol{Z}|\boldsymbol{X}; \boldsymbol{\Theta}_{\text{S2TT}}). \tag{18}$$

3) Joint T2TT and S2TT decoding: In our proposed method, we integrates the outputs of both T2TT in cascade system and S2TT by shallow fusion. In our proposed method, we integrate the outputs of both the T2TT model in the cascade system and S2TT model by shallow fusion as shown in Figure 2. We carry out beam-search decoding while integrating the probabilities of a distribution predicted from the S2TT and T2TT models. For each time step, the output distribution of each model is integrated as

$$score_t = \log P_{\text{S2TT}}(z_t|z_{1:t-1}, \boldsymbol{X}; \boldsymbol{\Theta}_{\text{S2TT}}) + w \log P_{\text{T2TT}}(z_t|z_{1:t-1}, \boldsymbol{Y}; \boldsymbol{\Theta}_{\text{T2TT}})\}, \tag{19}$$

where $score_t$ is the integrated score from both T2TT and S2TT models in the time step t and w is the T2TT weight for shallow fusion. The translation result $\boldsymbol{Z}$ is obtained by beam search decoding as

$$\hat{\boldsymbol{Z}} = \arg\max_{\boldsymbol{Z}} \sum_t score_t. \tag{20}$$

C. Semi-Supervised Learning

In our settings, the labeled training set is $\mathcal{D}_{\text{l}} = \{(\boldsymbol{X}_1, \boldsymbol{Z}_1), \cdots, (\boldsymbol{X}_{n_l}, \boldsymbol{Z}_{n_l})\}$ and the unlabeled data is $\mathcal{D}_{\text{u}} = \{\overline{\boldsymbol{X}}_1, \cdots, \overline{\boldsymbol{X}}_{n_u}\}$, where n_l and n_u are the number of samples in the labeled training and unlabeled training sets respectively. The following explanation is based on the steps in Figure 1.

Step 1: We train an initial S2TT model from the labeled training set. The parameters in the speech encoder and text decoder are initialized with the ASR encoder and T2TT decoder respectively. The model parameters $\boldsymbol{\Theta}_{\text{S2TT}}$ are optimized so as to maximize the generative probability in the decoder when given an input speech. Thus, the model parameters are optimized by minimizing the cross entropy loss as

$$\hat{\boldsymbol{\Theta}}_{\text{S2TT}} = \arg\min_{\boldsymbol{\Theta}_{\text{S2TT}}} - \sum_{(\boldsymbol{Z}', \boldsymbol{X}') \in \mathcal{D}_{\text{l}}} \log P(\boldsymbol{Z}'|\boldsymbol{X}'; \boldsymbol{\Theta}_{\text{S2TT}}). \tag{21}$$

Step 2: We determine a T2TT weight for joint T2TT and S2TT decoding in PL. We carry out joint decoding of the labeled development set with the cascade system and initial S2TT model from **step 1**. The T2TT weight is determined from the BLEU score on the development set and use the best weight for the next step.

TABLE I
DETAILS OF DATASETS.

Split	En→Ja		Ja→En	
	Size (hours)	# sentence	Size (hours)	# sentence
Labeled train	104.87	64,387	77.19	38,799
Unlabeled train	436.08	264,252	303.83	155,597
Dev	26.10	1,369	7.76	4,000
Test	24.65	2,841	8.17	4,000

Step 3: We convert the unlabeled training set into a pseudo-labeled training set. The unlabeled training set is decoded using the cascade system and initial S2TT model from **step 1** with the T2TT weight from **step 2**. The generated pseudo-labeled training set is written as $\mathcal{D}_{\rm pl} = \{(\overline{\boldsymbol{X}}_1, \overline{\boldsymbol{Z}}_1), \cdots, (\overline{\boldsymbol{X}}_{n_u}, \overline{\boldsymbol{Z}}_{n_u})\}$.

Step 4: At this point, we can use two labeled datasets: labeled training set $\mathcal{D}_{\rm l}$ and pseudo-labeled training set $\mathcal{D}_{\rm pl}$. The model parameters $\boldsymbol{\Theta}_{S2TT}$ are optimized by minimizing the cross entropy loss function with the two datasets as

$$\tilde{\boldsymbol{\Theta}}_{\rm S2TT} = \arg\min_{\boldsymbol{\Theta}_{\rm S2TT}} - \sum_{\mathcal{D}\in\{\mathcal{D}_{\rm l},\mathcal{D}_{\rm pl}\}} \sum_{(\boldsymbol{Z}',\boldsymbol{X}')\in\mathcal{D}} \log P(\boldsymbol{Z}'|\boldsymbol{X}';\boldsymbol{\Theta}_{\rm S2TT}). \quad (22)$$

IV. EXPERIMENTS

To evaluate the presented methods, we report BLEU results computed by sacrebleu [26] on the English→Japanese (En→Ja) and Japanese→English (En→Ja) S2TT task.

A. Setups

1) Datasets: Table I shows the details of the training, development, and test sets. We used the MuST-C v2.0 dataset [27]. The training set was split into labeled and unlabeled sets so that the talk was not split. The development and test sets are official splits of *dev* and *tst-common*, respectively.

2) Models and PL: **ASR and T2TT models:** The ASR model is a general domain model trained from various domain Japanese and English datasets of about 20K hours. The encoder and decoder in the ASR model have a 12-layer transformer and 2-layer transformer, respectively. The same ASR model was used for English ASR and Japanese ASR. The token embedding dimension, hidden-state dimension, non-linear layer dimension, and number of heads are 512, 512, 1024, and 4, respectively. The word error rate on the development set with this ASR model is 16.01 %. The T2TT model is trained from JParaCrawl v3 [28] of about 26M text pairs. The encoder and decoder in the T2TT model have an 8- and 6-layer transformer, respectively. The token embedding dimension, hidden-state dimension, non-linear layer dimension, and number of heads are 512, 512, 2048, and 8, respectively. Two separate T2TT models were trained by reversing the translation directions: one for En→Ja and the other for Ja→En. Note that these models were not fine-tuned on any target domain datasets in all experiments. Beam sizes for both models were set to 4. We deleted the sentences including repetitions from pseudo-labels to clean the training data.

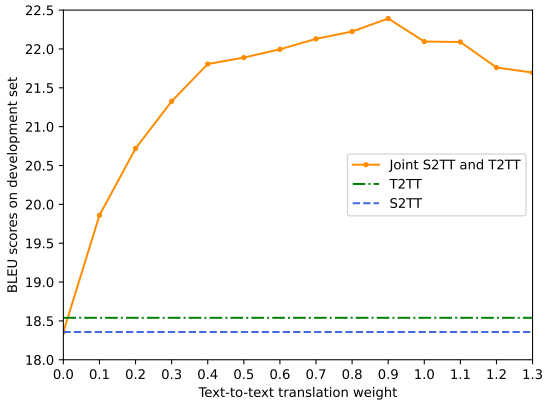

Fig. 3. Development set BLEU scores in different text-to-text translation weight on Ja→En.

S2TT models: The speech encoder has an 12-layer transformer initialized by the encoder in the ASR model. The connector has a 2-layer 1D convolution layer and 8-layer transformer. The text decoder has a 6-layer transformer initialized by the decoder in the T2TT model. In the connector, the token-embedding dimension, hidden-state dimension, non-linear layer dimension, and number of heads are 256, 256, 2048, and 4, respectively. We used 80-dimensional log Mel-filterbank as the acoustic features and applied SpecAugment [29] during the training. The vocabulary sizes were 10,411 characters for Japanese and 10,000 sub-words for English [30]. During the training, the decoder parameters were frozen except for layer normalization and cross-attention [31]. We used the RAdam optimizer [32] with an initial learning rate of 5e-5. Early stopping was applied if no best model was found in the development set for ten epochs. When decoding by beam search, the beam size was set to 4. For regularization, we used label smoothing [23] with a smoothing parameter of 0.1. We set the mini-batch size to 32.

PL: For evaluation, we carried out semi-supervised learning with PL using conventional methods using only T2TT (ASR-T2TT-PL) and only S2TT (S2TT-PL) models, and the proposed method using both. We also prepared a fully supervised model for when unlabeled data were labeled.

B. Results

1) Joint T2TT and S2TT Decoding: Figure 3 and 4 show the BLEU scores of the initial S2TT and T2TT models and joint T2TT and S2TT models with different T2TT weights on the Ja→En and En→Ja respectively. The combination of T2TT and S2TT models confirmed the improvement in BLEU score. The best weights were 0.9 for Ja→En and 0.7 for En→Ja, and these values were used in the PL mentioned in Section IV-B2.

We examined the relationship between the BLEU score on the development set and the frequency of tokens appearing in the training set. When the scored sentence is $\boldsymbol{w} = \{w_1, \cdots, w_l\}$ in the development set, $\boldsymbol{w}$ was scored

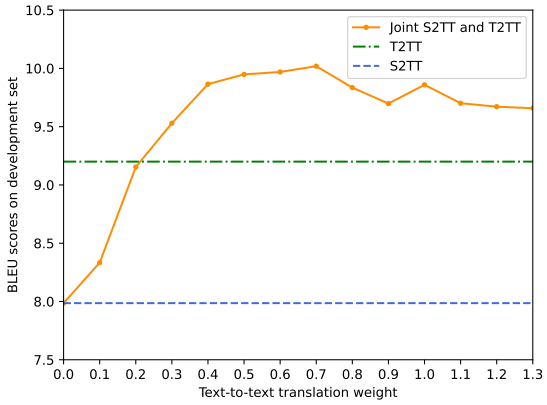

Fig. 4. Development set BLEU scores in different text-to-text translation weight on En→Ja.

TABLE II
DEVELOPMENT SET BLEU SCORES IN DIFFERENT GROUPS BASED ON FREQUENCY OF TOKEN OCCURRENCE ON EN→JA.

Group	T2TT	S2TT	Joint T2TT & S2TT
High frequency	9.33	8.85	11.41
Middle frequency	9.05	8.15	9.55
Low frequency	9.24	7.27	9.48

by $(1/l)\sum_{n=1}^{l} 1/f(w_n)$, where l is the number of tokens in the scored sentence and $f(w)$ is the frequency of w in the training set. Then, the development set was divided into three groups on the basis of the scores so that the groups have the same number of utterances. Table II showed the BLEU scores in the different groups based on the scores. It confirmed that the accuracy of the S2TT decreases as the number of low-frequency tokens increases. In addition, Joint T2TT and S2TT decoding outperformed T2TT and S2TT.

2) Semi-Supervised S2TT: Table III shows the BLEU scores for semi-supervised learning with each PL. We conducted inferences with and without joint decoding during inference. We confirmed that semi-supervised learning with any PL improved the accuracy from the initial S2TT. From comparing the proposed method with conventional methods, the proposed method had higher accuracy on the En→Ja and Ja→En tasks. This confirms the effectiveness of joint decoding, which enables highly accurate PL.

V. CONCLUSIONS

In this paper, we proposed a semi-supervised S2TT method for building end-to-end S2TT from a small amount of labeled dataset and a large amount of unlabeled dataset. In our method, we generate pseudo-labels with using both T2TT and S2TT models. We introduced a shallow fusion to integrate T2TT and S2TT efficiently. In our experiments, we confirmed the effectiveness of the joint T2TT and S2TT decoding. The results of the semi-supervised S2TT showed that our proposed semi-supervised learning outperformed conventional methods with using either T2TT or S2TT model for PL.

TABLE III
BLEU SCORES ON DEVELOPMENT AND TEST SETS IN DIFFERENT METHODS ON EN→JA AND JA→EN.

Method	En→Ja		Ja→En	
	dev	test	dev	test
ASR-T2TT	9.20	11.73	18.54	18.96
w/o joint decoding in inference				
Initial S2TT	7.99	9.32	18.38	19.32
S2TT-PL	8.11	10.40	21.54	22.15
ASR-T2TT-PL	8.79	11.27	21.98	22.31
Proposed PL	9.47	11.51	22.42	22.76
Fully supervised	11.16	14.00	27.67	28.28
w/ joint decoding in inference				
Initial S2TT	10.02	12.00	22.39	22.36
S2TT-PL	9.46	11.86	22.62	23.26
ASR-T2TT-PL	9.03	11.96	23.08	23.43
Proposed PL	10.13	12.20	23.33	23.71
Fully supervised	11.51	14.43	30.38	30.60

REFERENCES

[1] J. Pino, L. Puzon, J. Gu, X. Ma, A. D. McCarthy, and D. Gopinath, "Harnessing indirect training data for end-to-end automatic speech translation: Tricks of the trade," *In Proc. of International Conference on Spoken Language Translation (IWSLT)*, 2019.

[2] Y. Jia, M. Johnson, W. Macherey, *et al.*, "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," *In Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7180–7184, 2019.

[3] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," *In Proc. of Conference of the International Speech Communication Association (INTERSPEECH)*, 2017.

[4] A. Berard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," *In Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6224–6228, 2018.

[5] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," *In Proc. of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 58–68, 2019.

[6] M. C. Stoian, S. Bansal, and S. Goldwater, "Analyzing ASR pretraining for low-resource speech-to-text translation," *In Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7909–7913, 2020.

[7] Y. Liu, H. Xiong, J. Zhang, *et al.*, "End-to-end speech translation with knowledge distillation," *In Proc. of Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1128–1132, 2019.

[8] H. Wang, Z. Xue, Y. Lei, and D. Xiong, "End-to-end speech translation with mutual knowledge distilla-

tion," *In Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11 306–11 310, 2024.

[9] M. A. D. Gangi, M. Negri, and M. Turchi, "One-to-many multilingual end-to-end speech translation," *In Proc. of Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 585–592, 2019.

[10] H. Inaguma, K. Duh, T. Kawahara, and S. Watanabe, "Multilingual end-to-end speech translation," *In Proc. of Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 570–577, 2019.

[11] D. Zhang, R. Ye, T. Ko, M. Wang, and Y. Zhou, "DUB: discrete unit back-translation for speech translation," *In Proc. of Association for Computational Linguistics (ACL)*, pp. 7147–7164, 2023.

[12] J. M. Pino, Q. Xu, X. Ma, M. J. Dousti, and Y. Tang, "Self-training for end-to-end speech translation," *In Proc. of Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1476–1480, 2020.

[13] M. Gheini, T. Likhomanenko, M. Sperber, and H. Setiawan, "Joint speech transcription and translation: Pseudo-labeling with out-of-distribution data," *In Proc. of Meeting of the Association for Computational Linguistics (ACL)*, pp. 7637–7650, 2023.

[14] Z. Zhang, S. Liu, M. Li, M. Zhou, and E. Chen, "Joint training for neural machine translation models with monolingual data," *In Proc. of AAAI*, 2018.

[15] J. He, J. Gu, J. Shen, and M. Ranzato, "Revisiting self-training for neural sequence generation," *In Proc. of International Conference on Learning Representations (ICLR)*, 2020.

[16] W. Jiao, X. Wang, Z. Tu, S. Shi, M. R. Lyu, and I. King, "Self-training sampling with monolingual data uncertainty for neural machine translation," *In Proc. of Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pp. 2840–2850, 2021.

[17] Q. Xu, T. Likhomanenko, J. Kahn, A. Y. Hannun, G. Synnaeve, and R. Collobert, "Iterative pseudo-labeling for speech recognition," *In Proc. of Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1006–1010, 2020.

[18] D. S. Park, Y. Zhang, Y. Jia, *et al.*, "Improved noisy student training for automatic speech recognition," *In Proc. of Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2817–2821, 2020.

[19] J. Kahn, A. Lee, and A. Y. Hannun, "Self-training for end-to-end speech recognition," *In Proc. of Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 7084–7088, 2020.

[20] T. Likhomanenko, Q. Xu, J. Kahn, G. Synnaeve, and R. Collobert, "SlimIPL: Language-model-free iterative pseudo-labeling," *In Proc. of Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 741–745, 2021.

[21] Q. Dong, F. Yue, T. Ko, M. Wang, Q. Bai, and Y. Zhang, "Leveraging pseudo-labeled data to improve direct speech-to-speech translation," *In Proc. of Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1781–1785, 2022.

[22] Ç. Gülçehre, O. Firat, K. Xu, *et al.*, "On using monolingual corpora in neural machine translation," *CoRR*, vol. abs/1503.03535, 2015.

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

[25] S. Sedlácek, S. Kesiraju, A. Polok, and J. Cernocký, "Aligning pre-trained models for spoken language translation," *CoRR*, vol. abs/2411.18294, 2024.

[26] M. Post, "A call for clarity in reporting BLEU scores," *In Proc. of Conference on Machine Translation: Research Papers*, pp. 186–191, 2018.

[27] R. Cattoni, M. A. D. Gangi, L. Bentivogli, M. Negri, and M. Turchi, "MuST-C: A multilingual corpus for end-to-end speech translation," *Comput. Speech Lang.*, vol. 66, p. 101 155, 2021.

[28] M. Morishita, K. Chousa, J. Suzuki, and M. Nagata, "JParaCrawl v3.0: A large-scale English-Japanese parallel corpus," *In Proc. of Language Resources and Evaluation Conference (LREC)*, pp. 6704–6710, 2022.

[29] D. S. Park, W. Chan, Y. Zhang, *et al.*, "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.

[30] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *In Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 66–71, 2018.

[31] X. Li, C. Wang, Y. Tang, *et al.*, "Multilingual speech translation from efficient finetuning of pretrained models," *In Proc. of Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 2021.

[32] L. Liu, H. Jiang, P. He, *et al.*, "On the variance of the adaptive learning rate and beyond," *In Proc. of International Conference on Learning Representations (ICLR)*, 2020.