Transformer-Based End-to-End Speech Translation With Rotary Position Embedding

Xueqing Li, Shengqiang Li, Xiao-Lei Zhang, *Senior Member, IEEE*, and Susanto Rahardja, *Fellow, IEEE*

***Abstract*—Recently, many Transformer-based models have been applied to end-to-end speech translation because of their capability to model global dependencies. Position embedding is crucial in Transformer models as it facilitates the modeling of dependencies between elements at various positions within the input sequence. Most position embedding methods employed in speech translation such as the absolute and relative position embedding, often encounter challenges in leveraging relative positional information or adding computational burden to the model. In this letter, we introduce a novel approach by incorporating rotary position embedding into Transformer-based speech translation (RoPE-ST). RoPE-ST first adds absolute position information by multiplying the input vector with rotation matrices, and then implements relative position embedding through the dot-product of the self-attention mechanism. The main advantage of the proposed method over the original method is that rotary position embedding combines the benefits of absolute and relative position embedding, which is suited for position embedding in speech translation tasks. We conduct experiments on a multilingual speech translation corpus MuST-C. Results show that RoPE-ST achieves an average improvement of 2.91 BLEU over the method without rotary position embedding in eight translation directions.**

***Index Terms*—End-to-end speech translation, rotary position embedding, Transformer.**

I. Introduction

SPEECH translation (ST) is a task of converting or translating spoken phrases to a second language, which is the target language. It has wide applications in overcoming language barriers. Conventional ST techniques are cascaded architectures, which essentially cascade automatic speech recognition (ASR) and machine translation (MT) models [1]. Recently, end-to-end speech translation (E2E-ST), which translates speech into target text directly using a single model [2], [3], has been proposed and attracted considerable attention for its ability in avoiding error propagation and reducing high latency in the cascaded systems [4].

As deep learning technology has developed quickly, many deep architectures were used for E2E-ST, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and attention-based mechanisms [5]. Berard et al. [6] and Weiss et al. [7] used the bi-LSTM architecture, a special form of RNNs [8], [9], [10], when they first achieved E2E-ST. CNNs can reduce the time scale of speech signal to a reasonable range and they are often combined with RNNs or Transformer for E2E-ST [11]. In [12], Vaswani et al. proposed a Transformer-based system based on the attention mechanism. Di et al. [11] applied the Transformer model to E2E-ST and down-sampled the input with CNNs. This approach achieved superior translation quality over the RNNs-based baseline. Besides, many strategies incorporate auxiliary models or training methods to improve the Transformer-based ST models, including multi-task learning [13], pre-training [14], and knowledge distillation [15].

Although the Transformer-based ST models have demonstrated remarkable advantages, a critical aspect that is often overlooked is how to effectively handle sequential ordering. This limitation is caused by the inherent incapacity of the self-attention mechanism in modeling the position information within the input sequence. To address this issue, positional information has been incorporated into the Transformer-based ST models through absolute [11] and relative position embedding methods [16]. Absolute position embedding adds learnable [17] or pre-defined [12] embeddings to the input sequence before self-attention. However, the absolute position embedding is unable to capture relative information, and its suitability for acoustic modeling in ST is questionable because of the significantly different sequence lengths between speech and text [16], [18]. Relative position embedding is effective in exploiting the relative position between self-attention states, by directly modifying the attention calculation method for the inputs of ST [16]. However, it introduces new challenges, such as an increase of the model parameters and extended training times, compared to the absolute position embedding. In addition, some efforts have been spent on other aspects. For example, Islam et al. [19] proposed to combine sinusoidal position embedding in the Transformer model with the recurrent inductive bias of recurrent neural networks. Su et al. [20] proposed to rotate the input embedding according to the position information.

In this letter, we propose to apply the Rotary Position Embedding (RoPE) [20] for speech translation (RoPE-ST). Specifically, in contrast to existing position embedding methods which adds absolute positional information to the input embedding or

Manuscript received 27 July 2023; revised 26 November 2023; accepted 31 December 2023. Date of publication 11 January 2024; date of current version 24 January 2024. This work was supported in part by the National Science Foundation of China (NSFC) under Grant 62176211 and in part by the Project of the Science, Technology, and Innovation Commission of Shenzhen Municipality, China, under Grants JCYJ20210324143006016 and JSGG20210802152546026. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Konrad Kowalczyk. *(Corresponding authors: Xiao-Lei Zhang; Susanto Rahardja.)*

Xueqing Li and Shengqiang Li are with the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: lixueqing@mail.nwpu.edu.cn; shengqiangli@mail.nwpu.edu.cn).

Xiao-Lei Zhang is with the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China, and also with Research and Development Institute, Northwestern Polytechnical University, Shenzhen 710072, China (e-mail: xiaolei.zhang@nwpu.edu.cn).

Susanto Rahardja is with the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China, and also with Engineering Cluster, Singapore Institute of Technology, Singapore 138683 (e-mail: susantorahardja@ieee.org).

Digital Object Identifier 10.1109/LSP.2024.3353039

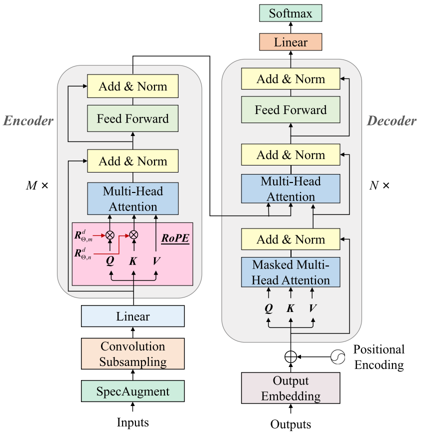


Fig. 1. Architecture of the proposed speech translation model. RoPE means rotary position embedding.

incorporates relative positional information by modifying the attention calculation, RoPE-ST first adds absolute position information by multiplying the rotation matrix with the input vector, and then implements relative position embedding through the dot-product mechanism in the self-attention layer. The results obtained from experiments on the MuST-C speech translation dataset show that our proposed model outperforms the original Transformer model by a BLEU score of 2.91 on average in eight translation directions.

II. Transformer for ST and the Model Architecture

A. The Framework of End-to-End Speech Translation

E2E-ST learns a sequence-to-sequence model that maps features extracted from speech signals in the source language to the respective text sequence in the target language [21]. We denote a speech translation corpus as $\mathcal{C} = \{(\mathbf{X}, \mathbf{U}, \mathbf{Y})\}$, where $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L]^T$ is the sequence of the input audio spectrogram frames, $\mathbf{U}$ is the transcription of $\mathbf{X}$, and $\mathbf{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K]^T$ is the translation of $\mathbf{U}$ to the target language. Therefore, E2E-ST aims to generate translation $\mathbf{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_J]^T$ from the input $\mathbf{X}$. The network is trained by minimizing the following cross-entropy:

$$L_{ST}(\beta) = -\sum_{\boldsymbol{x}\in\mathbf{X}, \boldsymbol{h}\in\mathbf{H}} \log P(\boldsymbol{h}\,|\boldsymbol{x}; \beta), \tag{1}$$

where β is trainable parameters of the model.

B. Model Architecture

In this letter, we adopt Transformer [12] as the basic speech translation model. The proposed architecture of the model is given in Fig. 1. In order to shrink the length of speech representations, we first use a convolution sub-sampling module to down-sample the acoustic input feature. The encoder first encodes the input sequence with position information by a rotary position embedding module, then feeds it into Transformer encoder blocks which encode speech input as an intermediate representation. Each block is comprised of two major modules: a multi-head self-attention layer and a feed forward layer, which are both wrapped by residual connections and layer normalization. Then, the decoder blocks decode the intermediate representation to a probability distribution over the target text feature space in an auto-regressive manner.

III. Rotary Position Embedding

A. Preliminaries

The self-attention mechanism in the Transformer model [12] can be expressed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{2}$$

where d is the dimension of the hidden representation, and Q, K and V denote query, key and value vectors respectively. Multi-head attention (MHA) allows the model to jointly attend to information from different representation sub-spaces, which is defined as:

$$\text{MHA}(Q, K, V) = \text{Concat}\left(\text{head}^1, \ldots, \text{head}^H\right)\mathbf{W}_o,$$
$$\text{head}^h = \text{Attention}\left(Q\mathbf{W}_q^h, K\mathbf{W}_k^h, V\mathbf{W}_v^h\right), \tag{3}$$

where $\mathbf{W}_q^h, \mathbf{W}_k^h, \mathbf{W}_v^h \in \mathbb{R}^{d\times d_m}$, $\mathbf{W}_o \in \mathbb{R}^{Hd_m\times d}$ are learnable parameter matrices, H is the number of heads, and d_m is the hidden size of the attention module.

Absolute position embedding is usually implemented by adding position embedding to the input. Specifically, each element in the sequence is assigned an absolute position-dependent embedding. The implementation of absolute position embedding can be formulated as:

$$\boldsymbol{q}_i = (\boldsymbol{x}_i + \boldsymbol{p}_i)\,\boldsymbol{W}_q,$$
$$\boldsymbol{k}_i = (\boldsymbol{x}_i + \boldsymbol{p}_i)\,\boldsymbol{W}_k, \tag{4}$$

where $\boldsymbol{q}_i$ and $\boldsymbol{k}_i$ represent query and key vector after position encoding respectively, $\boldsymbol{p}_i \in \mathbb{R}^d$ is a learnable [17] or pre-defined [12] vector assigned to position i with the same dimension as $\boldsymbol{x}_i$. Subsequently, the Transformer can allocate position-dependent attention score $a_{m,n}$ to each pair of vectors at positions m and n, denoted as $a_{m,n} = \frac{1}{\sqrt{d}}\boldsymbol{q}_m\boldsymbol{k}_n^T$.

In contrast, relative position embedding considers pairs of elements and their relative distance between each other. It often modifies the attention calculation [22] as:

$$a_{m,n} = \frac{1}{\sqrt{d}}\left(\boldsymbol{x}_m\boldsymbol{W}_q\right)\left(\boldsymbol{x}_n\boldsymbol{W}_k\right)^T + s_{m-n}, \tag{5}$$

where s_{m-n} is a learnable bias, $m - n$ represents the relative position distance of $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$.

B. Rotary Position Embedding

A weakness of the Transformer-based ST is that its self-attention mechanism (denoted as the **PLAIN** model in [23]) is invariant with respect to the reordering of the input sequence. Since speech translation is a content-dependent task, position information needs to be encoded into word embeddings. In order to utilize the relative position information between self-attention states and avoid computational complexity, it is necessary to encode the relative positional information into the input sequence of absolute position embedding. To achieve this, we incorporate absolute position information into the word embeddings $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$ using the functions $F_q(\cdot)$ and $F_k(\cdot)$:

$$\boldsymbol{q}_m = F_q\left(\boldsymbol{x}_m, m\right),$$
$$\boldsymbol{k}_n = F_k\left(\boldsymbol{x}_n, n\right), \tag{6}$$

where $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$ are assumed to be row vectors. Furthermore, to ensure that the positional information is only encoded in a relative form, we define a function $G(\cdot)$ as the inner product of query $\boldsymbol{q}_m$ and key $\boldsymbol{k}_n$, which only takes $\boldsymbol{x}_m$, $\boldsymbol{x}_n$, and $m - n$ as

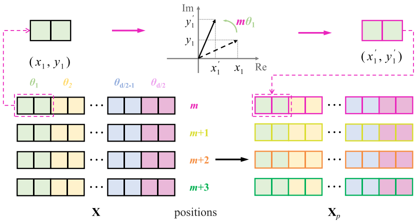


Fig. 2. Illustration of RoPE. $\mathbf{X}$ is the query/key of the input sequence. $\mathbf{X}_p$ is the query/key encoded with the position information.

its input variables:

$$\begin{aligned} \boldsymbol{q}_m \boldsymbol{k}_n^T &= \langle F_q(\boldsymbol{x}_m, m), F_k(\boldsymbol{x}_n, n)\rangle \\ &= G(\boldsymbol{x}_m, \boldsymbol{x}_n, m-n). \end{aligned} \tag{7}$$

Then, the key to achieving this position encoding mechanism is to design the functions $F_q(\cdot)$ and $F_k(\cdot)$.

In the following, we first introduce the RoPE with the dimension $d = 2$, and then generalize this situation to the RoPE when d is even. When the dimension $d = 2$, RoPE encodes the relative positional information into the input via the absolute position embedding [20]:

$$\begin{aligned} \boldsymbol{q}_m &= F_q(\boldsymbol{x}_m, m) = (\boldsymbol{x}_m \boldsymbol{W}_q) e^{jm\theta}, \\ \boldsymbol{k}_n &= F_k(\boldsymbol{x}_n, n) = (\boldsymbol{x}_n \boldsymbol{W}_k) e^{jn\theta}, \end{aligned} \tag{8}$$

where $\theta \in \mathbb{R}$ is a non-zero constant, and j is the mathematical symbol for the imaginary part of a complex number. Substituting (8) into (7) derives the function $G(\cdot)$ for RoPE:

$$\begin{aligned} \boldsymbol{q}_m \boldsymbol{k}_n^T &= G(\boldsymbol{x}_m, \boldsymbol{x}_n, m-n) \\ &= \mathrm{Re}\left[(\boldsymbol{x}_m \boldsymbol{W}_q)(\boldsymbol{x}_n \boldsymbol{W}_k)^* e^{j(m-n)\theta}\right]. \end{aligned} \tag{9}$$

As a result, $\boldsymbol{q}_m$ and $\boldsymbol{k}_n$ depend explicitly on m and n, respectively, while their inner product depends on $m - n$ only.

Note that, $F_q(\boldsymbol{x}_m, m)$ can be decomposed into the following multiplication of matrices:

$$F_q(\boldsymbol{x}_m, m) = \left[x_m^{(1)}, x_m^{(2)}\right] \begin{bmatrix} W_q^{(11)} & W_q^{(12)} \\ W_q^{(21)} & W_q^{(22)} \end{bmatrix} \begin{bmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{bmatrix}, \tag{10}$$

where $[x_m^{(1)}, x_m^{(2)}]$ represents the 2-dimensional (2D) coordinate of $\boldsymbol{x}_m$. The matrix $\left[\begin{smallmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{smallmatrix}\right]$ describes a rotation of the word embedding vector.

To generalize the case of RoPE with $d = 2$ to a general case that d can be any even, as shown in Fig. 2, we divide the d-dimensional vector into multiple 2D vectors, and combine them by the following linear superposition of inner product:

$$\begin{aligned} \boldsymbol{q}_m &= F_q(\boldsymbol{x}_m, m) = \boldsymbol{x}_m \boldsymbol{W}_q \boldsymbol{R}_{\Theta,m}^d, \\ \boldsymbol{k}_n &= F_k(\boldsymbol{x}_n, n) = \boldsymbol{x}_n \boldsymbol{W}_k \boldsymbol{R}_{\Theta,n}^d, \end{aligned} \tag{11}$$

where

$$\boldsymbol{R}_{\Theta,m}^d = \begin{bmatrix} \boldsymbol{M}_1 & & & \\ & \boldsymbol{M}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{M}_{d/2} \end{bmatrix}, \tag{12}$$

$$\boldsymbol{M}_r = \begin{bmatrix} \cos m\theta_r & -\sin m\theta_r \\ \sin m\theta_r & \cos m\theta_r \end{bmatrix}, \tag{13}$$

$$\Theta = \left\{\theta_r = 10000^{-2(r-1)/d}, r \in [1, 2, \ldots, d/2]\right\}. \tag{14}$$

In contrast to the additive sinusoidal position embedding employed in [12], we apply multiplicative rotary position embedding in the encoder, as shown in the RoPE module in Fig. 1. Instead of adding the position embedding to the input sequence, we simply multiply the query and key vectors with the rotation matrices at the self-attention layer.

At last, another possible application of the RoPE to ST is to apply it to the decoder as well. However, RoPE, as relative position embedding, performs better when dealing with long sequences. The input to the decoder is the text translated by the model in the past, and its sequence length is generally shorter, making it more suitable for the original sinusoidal position embedding. This finding is further validated in Section IV and in light of the outcome of the validation, a sinusoidal position embedding is therefore utilized in the decoder.

IV. Experiments

A. Experimental Settings

Experiments are conducted on a multilingual speech translation corpus MuST-C [24].[1] We select the best model on the dev set and report case-sensitive detokenized sacreBLEU [25] scores on the tst-COMMON set. Following the preprocessing recipes of FAIRSEQ S2T [26], we extract log-Mel filter-bank coefficients computed every 10 ms with a 25 ms window.

Our experiments are implemented based on the FAIRSEQ S2T toolkit.[2] RoPE-ST first pre-trains an ASR model with the English speech and its transcription, and use the ASR encoder as the encoder of ST model in Fig. 1. The decoder in Fig. 1 is then trained with the speech translation data from MuST-C. To reduce the computation cost, two 1D convolution layers are used in front of the encoder to down-sample the speech signal (along the temporal dimension) with a kernel size of 5 and a channel size of 1024, followed by non-linear activations of gated linear units. We set the learning rate to $2e - 3$. The dropout rate and label smoothing are both set to 0.1. During the inference, the last ten checkpoints are used for model averaging with a beam size of 5.

We compare the proposed system with a cascaded system [27], and four state-of-the-art E2E-ST systems, which are the Fairseq ST [26], AFS [28], Speechformer [29], and Simple and Effective ST [30] respectively. Note that the four end-to-end baselines and our methods do not use any additional training data. We use BLEU as the evaluation metric for ST.

B. Results

Table I summarizes the experimental results of the ST task where the competing methods listed in Table I did achieve efficient extraction of speech features [28], improvement of attention mechanisms [29], and optimization of training methods [30], but in contrast: our work focuses on improving the approach of position embedding. From the table, we see that RoPE-ST achieves an average of 25.88 BLEU over eight directions, performing better than the five competing methods on average. In addition, the performance of ST is enhanced by jointly pre-training the ASR model with the speech data from the eight directions. It outperforms the runner-up baseline, i.e.

[1] [Online]. Available: https://live.European-language-grid.eu/catalogue/corpus/4945

[2] [Online]. Available: https://github.com/pytorch/fairseq

TABLE I
BLEU↑ [%] SCORES ON THE MUST-C CORPUS

Models	En-De	En-Nl	En-Es	En-Fr	En-It	En-Pt	En-Ro	En-Ru	Avg.
Cascaded [27]	23.65	27.91	**28.68**	33.84	**24.04**	29.04	22.68	16.39	25.78
Fairseq ST [26]	22.70	27.30	27.20	32.90	22.70	28.10	21.90	15.30	24.76
AFS [28]	22.38	25.05	27.04	33.43	23.35	26.55	21.87	15.10	24.35
Speechformer [29]	23.60	27.70	28.50	-	-	-	-	-	-
Simple and Effective ST [30]	-	-	27.20	-	-	-	-	15.30	-
RoPE-ST (encoder-decoder)	22.66	26.91	26.84	33.30	22.60	28.20	21.65	14.93	23.91
RoPE-ST (Proposed)	23.85	27.87	28.15	33.41	23.90	29.13	22.87	16.17	25.67
RoPE-ST+joint (Proposed)	**23.86***	**28.36***	28.16	**34.23***	23.89	**29.47***	**22.90***	**16.19**	**25.88***

A higher BLEU score indicates better performance. "*" indicates that the score is superior to the cascaded method. "encoder-decoder" means that the position embedding in the encoder and decoder has been replaced with RoPE. "joint" means that the pretrained ASR model is trained on all speech data in eight directions.

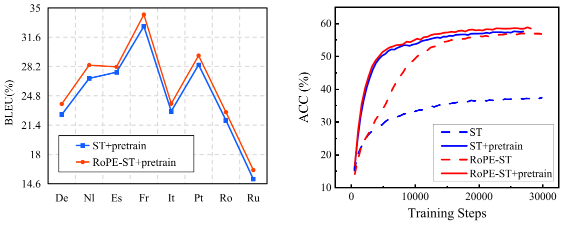
TABLE II
BLEU↑ [%] SCORES COMPARED WITH THE TRANSFORMER-BASED ST WITHOUT ROPE ON THE MUST-C CORPUS

Models	En-De	En-Nl	En-Es	En-Fr	En-It	En-Pt	En-Ro	En-Ru	Avg.
Transformer-based ST	20.15	25.33	25.55	31.10	21.91	26.10	20.09	13.53	22.97
+ pretrain	22.62	26.82	27.54	32.87	23.01	28.40	21.96	15.13	24.79
RoPE-ST	21.74	26.67	26.99	31.99	22.36	27.28	21.41	14.33	24.10
+ pretrain	23.86	28.36	28.16	34.23	23.89	29.47	22.90	16.19	**25.88**

TABLE III
BLEU ↑ SCORES OF THE ROPE-ST WITH DIFFERENT LENGTHS OF THE INPUT SEQUENCES

max_tokens	5,000	10,000	20,000	30,000	40,000
BLEU (En-De)	22.86	23.41	23.56	23.64	23.86
BLEU (En-Fr)	33.36	33.87	34.02	34.11	34.23

The "max tokens" means the maximum number of tokens in a batch.



(a) BLEU

(b) ACC

Fig. 3. (a) BLEU↑ on MuST-C. (b) Training accuracy on the MuST-C En-De dev set. The models in the legend correspond to rows 1, 2, 3, and 4 in Table II, respectively.

TABLE IV
COMPARISON OF THE PARAMETERS AND TRAINING TIME OF THE ST MODELS ON THE MUST-C EN-DE TASK

Models	Number of parameters	Training time
Transformer-based ST	39.0M	21h
RoPE-ST	29.5M	16h

cascaded method, on En-De, Nl, Fr, Pt and Ro. On the E2E track, although RoPE-ST does not achieve the best result on En-Es, it outperforms other methods by 1.1%∼13.2% in the other seven directions.

Besides, from Table I, we see that when RoPE is applied to both the encoder and decoder, the translation quality is noticeably inferior to the case when it is only used in the encoder. To investigate the relationship between the length of the input sequence and the performance of RoPE-ST, we conduct additional experiments on En-De and Fr. The results in Table III indicate that, when the length of the input sequence increases, the BLEU scores are improved, suggesting that RoPE-ST handles long inputs well. It is also seen that the performance decreases when max _ tokens is set below 10,000.

To study the effects of RoPE on the ST performance, we compare the proposed method with the Transformer-based ST without RoPE in Table II and Fig. 3(a). From the table and figure, we see that, if pre-training is not used, RoPE-ST outperforms the Transformer-based ST by an average BLEU score of 1.13 over the eight translation directions. After pretraining, RoPE-ST achieves an average improvement of 2.91 BLEU over the baseline model.

We also record the parameters and training time of RoPE-ST and the Transformer-based ST on En-De ST task. From Table IV, we see that RoPE-ST reduces the amount of model parameters, and shortens the training time over the Transformer-based ST. Fig. 3(b) lists the accuracy (BLEU score for unigram) of the comparison methods on the En-De Dev dataset during training. From the figure, we see that, RoPE-ST performs better than the Transformer-based ST, in terms of both accuracy and convergence speed, particularly when the pretraining is not employed.

V. CONCLUSION

In this letter, we propose RoPE-ST: adopting rotary position embedding to Transformer-based speech translation. RoPE-ST implements position embedding through encoding relative position information through a rotation matrix in the the self-attention layer. The experimental results show that our proposed method outperforms the Transformer-based ST and several other competitive methods.

It should be noted that, the translation performance of our model in two translation subtasks did not reach the optimum performance in the current comparison. In the future, we will further investigate whether this is caused by the grammatical differences between English used by the pre-trained model and the target languages. To further enhance the performance of speech translation, we can also leverage pretraining with large-scale data from speech recognition or machine translation tasks.

REFERENCES

[1] Q. Dong, F. Wang, Z. Yang, W. Chen, S. Xu, and B. Xu, “Adapting translation models for transcript disfluency detection,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 6351–6358.

[2] Q. Fang, R. Ye, L. Li, Y. Feng, and M. Wang, “STEMM: Self-learning with speech-text manifold mixup for speech translation,” 2022, *arXiv:2203.10426*.

[3] Y. Du, Z. Zhang, W. Wang, B. Chen, J. Xie, and T. Xu, “Regularizing end-to-end speech translation with triangular decomposition agreement,” in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 10590–10598.

[4] M. Sperber and M. Paulik, “Speech translation and the end-to-end promise: Taking stock of where we are,” in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7409–7421.

[5] U. Sulubacak et al., “Multimodal machine translation through visuals and speech,” *Mach. Transl.*, vol. 34, no. 2, pp. 97–147, 2020.

[6] A. Bérard, O. Pietquin, L. Besacier, and C. Servan, “Listen and translate: A proof of concept for end-to-end speech-to-text translation,” in *Proc. NIPS Workshop End-to-End Learn. Speech Audio Process.*, 2016.

[7] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, “Sequence-to-sequence models can directly translate foreign speech,” in *Proc. Interspeech*, 2017, pp. 2625–2629.

[8] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, “End-to-end automatic speech translation of audiobooks,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 6224–6228.

[9] Y. Jia et al., “Leveraging weakly supervised data to improve end-to-end speech-to-text translation,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 7180–7184.

[10] P. Bahar, A. Zeyer, R. Schlueter, and H. Ney, “On using specaugment for end-to-end speech translation,” in *Proc. 16th Int. Conf. Spoken Lang. Transl.*, 2019.

[11] M. A. Di Gangi, M. Negri, and M. Turchi, “Adapting transformer to end-to-end spoken language translation,” in *Proc. Int. Speech Commun. Assoc.*, 2019, pp. 1133–1137.

[12] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.

[13] R. Imaizumi et al., “End-to-end Japanese multi-dialect speech recognition and dialect identification with multi-task learning,” *APSIPA Trans. Signal Inf. Process.*, vol. 11, no. 1, 2022.

[14] N.-Q. Pham, A. Waibel, and J. Niehues, “Adaptive multilingual speech recognition with pretrained models,” 2022, *arXiv:2205.12304*.

[15] Y. Lee, K. JANG, J. Goo, Y. Jung, and H.-R. Kim, “FitHuBERT: Going thinner and deeper for knowledge distillation of speech self-supervised learning,” in *Proc. 23rd Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 3588–3592.

[16] N.-Q. Pham et al., “Relative positional encoding for speech recognition and direct translation,” in *Proc. Conf. Interspeech*, 2020, pp. 31–35.

[17] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, “ELECTRA: Pre-training text encoders as discriminators rather than generators,” 2020, *arXiv:2003.10555*.

[18] G. Ke, D. He, and T.-Y. Liu, “Rethinking positional encoding in language pre-training,” in *Proc. Int. Conf. Learn. Representations*, 2020.

[19] M. A. Islam, S. Jia, and N. D. Bruce, “How much position information do convolutional neural networks encode,” in *Proc. Int. Conf. Learn. Representations*, 2019.

[20] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu, “RoFormer: Enhanced transformer with rotary position embedding,” 2021, *arXiv:2104.09864*.

[21] J. Zhao, W. Luo, B. Chen, and A. Gilman, “Mutual-learning improves end-to-end speech translation,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 3989–3994.

[22] C. Raffel et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.

[23] P. Dufter, M. Schmitt, and H. Schütze, “Position information in transformers: An overview,” *Comput. Linguistics*, vol. 48, pp. 733–763, 2022.

[24] M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, “MuST-C: A multilingual speech translation corpus,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2019, pp. 2012–2017.

[25] M. Post, “A call for clarity in reporting BLEU scores,” in *Proc. 3rd Conf. Mach. Transl. Res. Papers*, 2018, pp. 186–191.

[26] C. Wang, Y. Tang, X. Ma, A. Wu, D. Okhonko, and J. Pino, “FAIRSEQ S2T: Fast speech-to-text modeling with FAIRSEQ,” in *Proc. 1st Conf. Asia-Pacific Chapter Assoc. Comput. Linguistics 10th Int. Joint Conf. Natural Lang. Process. Syst. Demonstrations*, 2020, pp. 33–39.

[27] H. Inaguma et al., “ESPnet-ST: All-in-one speech translation toolkit,” in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics Syst. Demonstrations*, 2020, pp. 302–311.

[28] B. Zhang, I. Titov, B. Haddow, and R. Sennrich, “Adaptive feature selection for end-to-end speech translation,” in *Proc. Findings Assoc. Comput. Linguistics*, 2020, pp. 2533–2544.

[29] S. Papi, M. Gaido, M. Negri, and M. Turchi, “SpeechFormer: Reducing information loss in direct speech translation,” 2021, *arXiv:2109.04574*.

[30] C. Wang et al., “Simple and effective unsupervised speech translation,” 2022, *arXiv:2210.10191*.

[31] E. A. P. Habets, “Room impulse response (RIR) generator,” 2016. [Online]. Available: https://github.com/ehabets/RIR-Generator34

[32] G. O. Young, “Synthetic structure of industrial plastics,” in *Plastics*, 2nd ed., vol. 3, J. Peters, Ed. New York, NY, USA: McGraw-Hill, 1964, pp. 15–64.

[33] W.-K. Chen, *Linear Networks and Systems*, Belmont, CA, USA: Wadsworth, 1993, pp. 123–135.