Regularizing End-to-End Speech Translation with Triangular Decomposition Agreement

Yichao Du[1], Zhirui Zhang[2], Weizhi Wang[3], Boxing Chen[2], Jun Xie[2] and Tong Xu[1*]

[1]University of Science and Technology of China, Hefei, China
[2]Machine Intelligence Technology Lab, Alibaba DAMO Academy [3]Rutgers University, New Brunswick, USA
[1]duyichao@mail.ustc.edu.cn [1]tongxu@ustc.edu.cn
[2]{zhirui.zzr, boxing.cbx, qingjing.xj}@alibaba-inc.com [3]weizhi.wang@rutgers.edu

Abstract

End-to-end speech-to-text translation (E2E-ST) is becoming increasingly popular due to the potential of its less error propagation, lower latency, and fewer parameters. Given the triplet training corpus $\langle speech, transcription, translation\rangle$, the conventional high-quality E2E-ST system leverages the $\langle speech, transcription\rangle$ pair to pre-train the model and then utilizes the $\langle speech, translation\rangle$ pair to optimize it further. However, this process only involves two-tuple data at each stage, and this loose coupling fails to fully exploit the association between triplet data. In this paper, we attempt to model the joint probability of transcription and translation based on the speech input to directly leverage such triplet data. Based on that, we propose a novel regularization method for model training to improve the agreement of dual-path decomposition within triplet data, which should be equal in theory. To achieve this goal, we introduce two Kullback-Leibler divergence regularization terms into the model training objective to reduce the mismatch between output probabilities of dual-path. Then the well-trained model can be naturally transformed as the E2E-ST models by the pre-defined early stop tag. Experiments on the MuST-C benchmark demonstrate that our proposed approach significantly outperforms state-of-the-art E2E-ST baselines on all 8 language pairs, while achieving better performance in the automatic speech recognition task.

Introduction

Speech-to-text translation (ST) processes speech signals in a source language and generates the text in a target language. Traditional ST approaches cascade automatic speech recognition (ASR) and machine translation (MT) (Ney 1999; Sperber et al. 2017; Zhang et al. 2019a; Iranzo-Sánchez et al. 2020). With the rapid development of deep learning, the neural networks which are widely used in ASR and MT have been adapted to construct a new end-to-end speech-to-text translation (E2E-ST) paradigm (Liu et al. 2019; Wang et al. 2020b; Dong et al. 2021b). This approach aims to overcome known limitations of the cascade one and learns a single unified encoder-decoder model, which is easier to deploy, has lower latency and has less error propagation.

Despite the potential benefits, it is very challenging to develop a well-trained E2E-ST model that does not use intermediate transcriptions. Thus, various techniques have been proposed to ease the training process by using source transcriptions, including pre-training (Bansal et al. 2019; Wang et al. 2020b,c), multi-task learning (Anastasopoulos and Chiang 2018; Sperber et al. 2019), meta-learning (Indurthi et al. 2020), consecutive decoding (Dong et al. 2021a), and interactive decoding (Liu et al. 2020). Among them, the pre-training strategy is a simple yet effective way, which is widely used to build the high-quality E2E-ST system in practice. Specifically, given the triplet training dataset $\langle speech, transcription, translation\rangle$, the pre-training method first leverages the $\langle speech, transcription\rangle$ pair to pre-train the E2E-ST model via ASR and then further fine-tunes it with the $\langle speech, translation\rangle$ pair. However, these methods only utilize two-tuple data at each stage, failing to fully explore the association relationship in triplet data. Therefore, how to fully mine the associations between the triplet data still remains a crucial issue to improve the performance of the E2E-ST model.

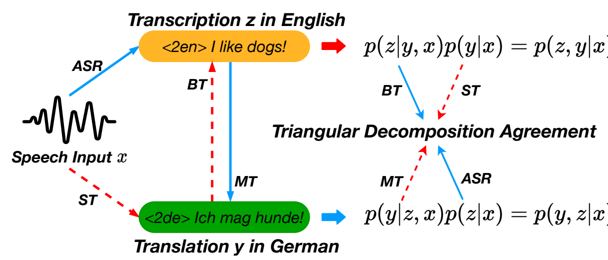

Figure 1: A training exemplar of English-to-German speech-to-text task based on triangular decomposition. We can follow ASR-MT (in blue line) and ST-BT (in red line) decoding paths respectively to model the joint probability of transcription and translation given the speech input. According to the chain decomposition rule, the probability distribution of these two ways is equivalent.

In this paper, we attempt to directly learn the joint probability of transcription and translation by a single unified encoder-decoder model and successive decoding to involve the whole triplet data. Actually, as shown in Figure 1, there are two different paths in the decoding process to fit this joint probability according to the triangular decomposition: (a) The speech signal x is first converted into source tran-

[*]Corresponding author.

scription z and then translated into target translation y. We name this decoding process as ASR-MT path, which can be formalized as $p(y, z|x) = p(z|x)p(y|z, x)$; (b) Meanwhile, we can directly perform speech-to-text translation on the speech signal and then adopt back-translation (BT) process to obtain the source transcription. This decoding process is formalized into $p(y, z|x) = p(y|x)p(z|y, x)$ and named as ST-BT path. We adopt the language tag to distinguish these two different decoding paths. Theoretically, the conditional distributions of such dual-path should be consistent due to the chain decomposition rule. However, there is no guarantee that the above relationship will hold, if these two lines are learned separately.

Based on this observation, we propose a novel model regularization method called **T**riangular **D**ecomposition **A**greement (**TDA**) to better exploit the association between triplet data. This goal is achieved by introducing two Kullback-Leibler divergence regularization terms into the original training objective to enhance the agreement between output probabilities of dual-path. In this way, we not only maximize the probability distribution of the triplet training data, but also minimize the mismatch between ASR-MT and ST-BT decoding paths to promote the training process in the correct direction. In the inference stage, the well-trained model can be naturally transformed as the ASR and E2E-ST models by choosing ASR-MT and ST-BT decoding paths respectively, thus keeping the same inference delay as the previous ASR and E2E-ST models.

Our experiments are conducted on the MuST-C benchmark with all 8 language pairs, and demonstrate that our proposed approach not only gains up to 1.8 BLEU score improvements over the E2E-ST baseline on average but also achieves better performance in the ASR task. Our code is open-sourced at https://github.com/duyichao/E2E-ST-TDA.

Background: End-to-End Speech Translation

In this section, we first give a formal definition of the E2E-ST task, then briefly introduce the backbone model we use.

Problem Formulation. The ST training corpus consists of a set of triplet data $\mathcal{D}_{ST} = \left\{(\mathbf{x}^{(n)}, \mathbf{z}^{(n)}, \mathbf{y}^{(n)})\right\}_{n=1}^{N}$. Here $\mathbf{x}^{(n)} = (x_1^{(n)}, x_2^{(n)}, ..., x_{|\mathbf{x}^{(n)}|}^{(n)})$ denotes the input sequence of the speech wave (in most cases, acoustic features are used), $\mathbf{z}^{(n)} = (z_1^{(n)}, z_2^{(n)}, ..., z_{|\mathbf{z}^{(n)}|}^{(n)})$ is the transcription sequence from the source language and the $\mathbf{y^{(n)}} = (y_1^{(n)}, y_2^{(n)}, ..., y_{|\mathbf{y}^{(n)}|}^{(n)})$ represents the translation sequence of target language. The goal of E2E-ST is to directly seek an optimal translation sequence $\mathbf{y}$ without generating an intermediate transcription $\mathbf{z}$, and the standard training objective is to optimize the maximum likelihood estimation (MLE) loss of the training data:

$$\mathcal{L}_{MLE}(\theta) = \sum_{n=1}^{N} \log P\left(\mathbf{y^{(n)}} \mid \mathbf{x^{(n)}}; \theta\right), \tag{1}$$

where we use a single encoder-decoder structure to learn the conditional distribution $P(\mathbf{y^{(n)}}|\mathbf{x^{(n)}})$ and θ is the model parameter. In order to obtain the high-quality E2E-ST system, previous methods usually leverage ASR and MT tasks ($\{\mathbf{x}^{(n)}, \mathbf{z}^{(n)}\}$ and $\{\mathbf{z}^{(n)}, \mathbf{y}^{(n)}\}$) to pre-train the encoder and decoder respectively (Bansal et al. 2019; Wang et al. 2020b,c). Since this process only adopts two-tuple data at each stage, this loose coupling fails to fully utilize the association between triplet data.

Backbone E2E-ST Model. In this work, we adopt the transformer-based (Vaswani et al. 2017) structure as the backbone, which has become increasingly common in the speech processing field. Concretely, the entire encoder consists of a multi-layer convolutional down-sampling module and a transformer-based encoder. The multi-layer convolutional module takes the acoustic features as input to generate local representation, which is then fed to the transformer-based encoder to output the contextual representation. The transformer-based decoder conducts token classification for the next word prediction by considering the output of the encoder and predictions of previous tokens. It is worth noting that our method can be easily applied to any other encoder-decoder architecture.

Dual-Path Decoding with Triangular Decomposition Agreement

In order to make full use of the triplet data $(\mathbf{x}, \mathbf{z}, \mathbf{y}) \in \mathcal{D}_{ST}$ within a single unified encoder-decoder model, in this work, we directly learn the joint probability $P(\mathbf{y}, \mathbf{z}|\mathbf{x})$ of transcription and translation given the speech input. To this end, we propose a novel model regularization method called **T**riangular **D**ecomposition **A**greement (**TDA**) to fully exploit the association between triplet data, as illustrated in Figure 2. In this way, the whole training objectives are decomposed into two parts: the standard maximum likelihood of training data, and the regularization terms that indicate the divergence of dual-path based on the current model parameter. In this section, we start with the dual-path decoding based on the joint probability. Furthermore, we introduce additional training objectives in accordance with TDA to improve the agreement of the dual-path decoding. In the last part, we show the flexibility of our method during inference.

Dual-Path Decoding

We jointly model the generation of transcription and translation in a single decoder. In this case, the optimization objective can be calculated by:

$$\mathcal{L}_{MLE}(\theta) = \sum_{n=1}^{N} \log P\left(\mathbf{y^{(n)}}, \mathbf{z^{(n)}} \mid \mathbf{x^{(n)}}; \theta\right), \tag{2}$$

where $P\left(\mathbf{y^{(n)}}, \mathbf{z^{(n)}} \mid \mathbf{x^{(n)}}; \theta\right)$ is the ST model that adopts successive decoding. Actually, as shown in Figure 2, there are two different decomposition paths for such conditional probability $P\left(\mathbf{y}, \mathbf{z} \mid \mathbf{x}; \theta\right)$:

- **ASR-MT Decoding**. The source transcription $\mathbf{z}$ is first produced by ASR, followed by generating target translation $\mathbf{y}$ through MT:

$$P\left([\mathbf{z}, \mathbf{y}] \mid \mathbf{x}; \theta\right) = P\left(\mathbf{z} \mid \mathbf{x}; \theta\right) P\left(\mathbf{y} \mid \mathbf{z}, \mathbf{x}; \theta\right), \tag{3}$$

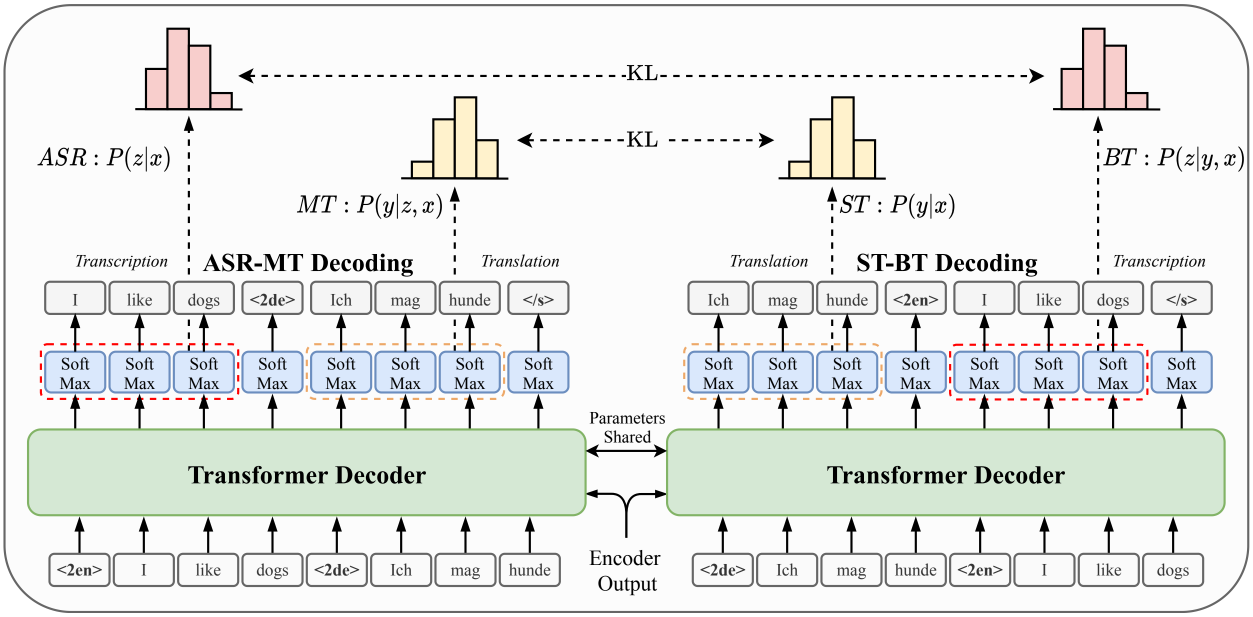

Figure 2: The overview of dual-path decoding with triangular decomposition agreement. The pink histogram is the probability distribution of an example token "dogs" in the ASR and BT sequences, and similarly, the orange histogram represents the probability distribution of the token "hunde" in the ST and MT sequences.

where $[\mathbf{z}, \mathbf{y}]$ means the concatenation of $\mathbf{z}$ and $\mathbf{y}$.

- **ST-BT Decoding**. The decoder output $[\mathbf{y}, \mathbf{z}]$ is the concatenation of the translation $\mathbf{y}$ generated by ST and transcription $\mathbf{z}$ obtained by BT process:

$$P([\mathbf{y}, \mathbf{z}] \mid \mathbf{x}; \theta) = P(\mathbf{y} \mid \mathbf{x}; \theta) P(\mathbf{z} \mid \mathbf{y}, \mathbf{x}; \theta). \quad (4)$$

We perform this dual-path decoding in a parameter shared transformer-based decoder and leverage the language tag to distinguish different paths. Specifically, taking English-to-German ST in Figure 2 as an example, ASR-MT decoding utilizes the language tag `<2en>` as begin-of-sentence (BOS) and generates transcription sequence. Unlike the standard decoding, we take `<2de>` as the end of the transcription sequence. When the decoder recognizes `<2de>`, it will continue to produce the translation sequence and take end-of-sentence (EOS) as the end mark. In this way, we series the transcription-translation sequence through the `<2de>` identifier. Similarly, we adopt `<2de>` as the BOS to select the ST-BT decoding, while the translation-transcription sequence is concatenated by the `<2en>`. Therefore, the original training objective in Equation 2 can be rewritten as:

$$\begin{aligned}\mathcal{L}_{MLE}(\theta) = &\sum_{n=1}^{N} \log P\left([\mathbf{y}^{(\mathbf{n})}, \mathbf{z}^{(\mathbf{n})}] \mid \mathbf{x}^{(\mathbf{n})}; \theta\right) \\ &+ \sum_{n=1}^{N} \log P\left([\mathbf{z}^{(\mathbf{n})}, \mathbf{y}^{(\mathbf{n})}] \mid \mathbf{x}^{(\mathbf{n})}; \theta\right).\end{aligned} \quad (5)$$

Model Regularization

According to the chain decomposition rule, output probabilities of these dual paths should be identical if the learned model is perfect (we drop θ for concise):

$$\begin{aligned}P([\mathbf{y}, \mathbf{z}] \mid \mathbf{x}) &= P(\mathbf{y} \mid \mathbf{x}) P(\mathbf{z} \mid \mathbf{y}, \mathbf{x}) \\ &= P(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) P(\mathbf{z} \mid \mathbf{x}) = P([\mathbf{z}, \mathbf{y}] \mid \mathbf{x}).\end{aligned} \quad (6)$$

However, if these two paths are optimized independently by MLE (like Equation 5), there is no guarantee that the above equation will hold. To handle this problem, we introduce two word-level Kullback-Leibler (KL) divergence based on the output probability as the regularization terms, aiming to enhance the agreement between ASR-MT and ST-BT decoding paths. Since KL divergence is asymmetric, it involves two directions:

$$\begin{aligned}\overrightarrow{\mathrm{KL}} &= \mathrm{KL}\left(P(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) \,||\, P(\mathbf{y} \mid \mathbf{x})\right) \\ &\quad + \mathrm{KL}\left(P(\mathbf{z} \mid \mathbf{x}) \,||\, P(\mathbf{z} \mid \mathbf{y}, \mathbf{x})\right) \\ &= \sum_{t=1}^{|\mathbf{y}|} P\left(y_t \mid y_{<t}, z_{\leq|\mathbf{z}|}, \mathbf{x}\right) \log \frac{P\left(y_t \mid y_{<t}, z_{\leq|\mathbf{z}|}, \mathbf{x}\right)}{P\left(y_t \mid y_{<t}, \mathbf{x}\right)} \\ &\quad + \sum_{t'=1}^{|\mathbf{z}|} P\left(z_{t'} \mid z_{<t'}, \mathbf{x}\right) \log \frac{P\left(z_{t'} \mid z_{<t'}, \mathbf{x}\right)}{P\left(z_{t'} \mid z_{<t'}, y_{\leq|\mathbf{y}|}, \mathbf{x}\right)}, \\ \overleftarrow{\mathrm{KL}} &= \mathrm{KL}\left(P(\mathbf{y} \mid \mathbf{x}) \,||\, P(\mathbf{y} \mid \mathbf{z}, \mathbf{x})\right) \\ &\quad + \mathrm{KL}\left(P(\mathbf{z} \mid \mathbf{y}, \mathbf{x}) \,||\, P(\mathbf{z} \mid \mathbf{x})\right) \\ &= \sum_{t'=1}^{|\mathbf{y}|} P\left(y_{t'} \mid y_{<t'}, \mathbf{x}\right) \log \frac{P\left(y_t \mid y_{<t'}, \mathbf{x}\right)}{P\left(y_{t'} \mid y_{<t'}, z_{\leq|\mathbf{z}|}, \mathbf{x}\right)} \\ &\quad + \sum_{t=1}^{|\mathbf{z}|} P\left(z_t \mid z_{<t}, y_{\leq|\mathbf{y}|}, \mathbf{x}\right) \log \frac{P\left(z_t \mid z_{<t}, y_{\leq|\mathbf{y}|}, \mathbf{x}\right)}{P\left(z_t \mid z_{<t}, \mathbf{x}\right)}\end{aligned} \quad (7)$$

where $|\mathbf{y}|$ and $|\mathbf{z}|$ represent the length of translation and transcription respectively. The entire regularization term is summarized as:

$$\mathcal{L}_{TDA}(\theta) = \overrightarrow{\mathrm{KL}} + \overleftarrow{\mathrm{KL}}. \tag{8}$$

The Equation 6 holds when these regularization terms are 0, otherwise regularization terms will guide the training process to reduce the disagreement of output probabilities of dual-path decoding. Besides, we assign a weighting term to this loss and combine it with the MLE loss to obtain the entire model training objective, as described by:

$$\mathcal{L}(\theta) = \mathcal{L}_{MLE}(\theta) - \lambda\mathcal{L}_{TDA}(\theta), \tag{9}$$

where λ is a hyper-parameter to balance the preference between the ground truth and agreement distribution.

Inference

Since we can adapt language tags to switch the ASR-MT and ST-BT decoding paths, it gives the flexibility of the inference strategy. The well-trained model is naturally transformed as the ASR and E2E-ST models by choosing ASR-MT and ST-BT decoding paths, respectively. Specifically, we directly select the ST-BT path to conduct the English-to-German ST task, and terminate the inference when generating language identifier `<2en>`. In this way, our proposed approach maintains the same decoding speed as the traditional E2E-ST model. Similarly, we can gain the ASR result by selecting the ASR-MT decoding path and terminating the decoding when `<2de>` is recognized. In addition, our approach can simultaneously leverage two ways to inference and adopt the way of premature termination for different scenarios, that is, only the corresponding text is displayed, or both transcription and translation are displayed to the user at the same time.

Experiments

Setup

We consider restricted and extended settings on the benchmark MuST-C to evaluate the effectiveness of our proposed approach. For the restricted setting, we run experiments on the MuST-C dataset with all 8 languages. For comparison in practical scenarios, we extend the above setting to verify the gain of our method on English-to-German and English-to-French translation directions with available external ASR and MT data.

MuST-C Dataset. MuST-C (Gangi et al. 2019) is a publicly large-scale multilingual speech-to-text translation corpus, consisting of triplet data sources: source speech, source transcription, and target translation. The speech sources of MuST-C are from English TED Talks, which are aligned at the sentence level with their manual transcriptions and translations. MuST-C contains translations from English (EN) to 8 languages: Dutch (NL), French (FR), German (DE), Italian (IT), Portuguese (PT), Romanian (RO), Russian (RU), and Spanish (ES). The statistics of different language pairs are illustrated in Table 1.

Language	Sentence Pair	Speech Duration	Source Words	Target Words
German (DE)	234 K	408 hrs	4.3 M	4.0 M
French (FR)	280 K	492 hrs	5.2 M	5.4 M
Spanish (ES)	270 K	504 hrs	5.3 M	5.1 M
Italian (IT)	258 K	465 hrs	4.9 M	4.6 M
Dutch (NL)	253 K	442 hrs	4.7 M	4.3 M
Portuguese (PT)	211 K	385 hrs	4.0 M	3.8 M
Romanian (RO)	240 K	432 hrs	4.6 M	4.3 M
Russian (RU)	270 K	489 hrs	5.1 M	4.3 M

Table 1: The statistics of 8 translation directions in the MuST-C dataset.

External ASR and MT Datasets. We introduce the LibriSpeech dataset (Panayotov et al. 2015) as the external ASR data. The LibriSpeech ASR dataset is derived from audiobooks that are part of the LibriVox project. This dataset contains 960 hours of speech samples in English and approximately 290K speech-transcription pair samples, in which the transcription texts are not punctuated and capitalized. We adopt English-to-German and English-to-French WMT14 (Bojar et al. 2014) training data as the external MT parallel corpus in the extended setting, which consists of 4M and 30M bilingual sentence pairs, respectively.

Pre-processing of Data. We follow FAIRSEQ S2T (Wang et al. 2020a) recipes to perform data pre-processing. For speech data, both in LibriSpeech and MuST-C, acoustic features are 80-dimensional log-mel filter banks extracted with a stepsize of 10ms and a window size of 25ms. The acoustic features are normalized by global channel mean and variance. In addition, the SpecAugment method (Park et al. 2019) is applied for all experiments, and the samples of more than 3000 frames are removed. As for text data in MuST-C and WMT14, we reserve punctuation, as well as the original word splitting and normalization. We lowercase all transcription sentences in the Librispeech ASR dataset, capitalize the first letter of all sentences and put a full stop at the end of the sentence to be consistent with MuST-C and WMT14 datasets. For sub-wording, we employ the unigram sentencepiece[1] model to build a sub-word vocabulary with a size of 10000. On each translation direction, the sentencepiece model is learned on text data from the training set, and the dictionary is shared across source and target languages.

Methods. We compare our proposed approach (E2E-ST-TDA) with several baseline methods in the experiment:

- E2E-ST-Base: we optimize the E2E-ST model with the training process proposed in Wang et al. (2020a). The model is first pre-trained with speech-transcription pairs and then directly fine-tuned by speech-translation pairs.
- E2E-ST-JT: we train the E2E-ST model with multi-task learning, including ASR and ST tasks.
- E2E-ST-TDA: we extend the E2E-ST-Base method with the proposed model regularization method TDA.

[1]https://github.com/google/sentencepiece

Model	Params.	Extra.	EN-DE	EN-FR	EN-RU	EN-ES	EN-IT	EN-RO	EN-PT	EN-NL	Avg.
ESPnet ST	31M	×	22.9	32.8	15.8	28.0	23.8	21.9	28.0	27.4	25.1
ESPnet Cascaded	84M	×	23.6	33.8	16.4	28.7	24.0	22.7	29.0	27.9	25.8
Fairseq ST	31M	×	22.7	32.9	15.3	27.2	22.7	21.9	28.1	27.3	24.8
Fairseq Multi-ST	76M	×	24.5	34.9	16.0	28.2	24.6	23.8	**31.1**	28.6	26.5
AFS	-	×	22.4	31.6	14.7	26.9	23.0	21.0	26.3	24.9	23.9
Dual-Decoder	48M	×	23.6	33.5	15.2	28.1	24.2	22.9	30.0	27.6	25.7
W2V2-Transformer	-	✓	22.3	34.3	15.8	28.7	24.2	22.4	29.3	28.2	25.6
LNA-E,D	76M	✓	24.3	34.6	15.9	28.4	24.4	23.3	30.5	28.3	26.2
Adapter Tuning	76M	×	24.6	34.7	**16.4**	28.7	25.0	23.7	31.0	28.8	26.6
E2E-ST-Base^s	31M	×	22.8	33.0	15.2	27.2	22.9	21.6	28.0	27.3	24.8
E2E-ST-JT^s	32M	×	23.1	32.8	14.9	27.5	23.6	22.1	28.7	27.8	25.1
E2E-ST-TDA^s	32M	×	24.3	34.6	15.9	28.3	24.2	23.4	30.3	28.7	26.2
E2E-ST-Base^m	74M	×	23.5	33.8	15.5	27.8	23.4	22.8	28.6	27.5	25.4
E2E-ST-JT^m	76M	×	23.2	34.1	14.9	28.2	23.1	22.4	28.4	27.9	25.3
E2E-ST-TDA^m	76M	×	**25.4**	**36.1**	**16.4**	**29.6**	**25.1**	**23.9**	**31.1**	**29.6**	**27.2**

Table 2: BLEU scores of different methods on MuST-C tst-COMMON set. "Extra." indicates whether the method uses additional data. "Params." represents the parameter scale of the model. The superscripts s and m represent the small model and medium model, respectively.

Model	EN-DE	EN-FR	EN-RU	EN-ES	EN-IT	EN-RO	EN-PT	EN-NL	Avg.
E2E-ST-Base^s	18.2	17.2	17.7	17.7	17.9	18.1	19.1	17.6	17.9
E2E-ST-JT^s	17.2	16.7	16.9	16.4	16.8	16.8	17.8	16.6	16.9
E2E-ST-TDA^s	16.4	15.6	16.6	16.4	16.2	16.6	16.9	16.2	16.4
E2E-ST-Base^m	16.8	16.9	16.9	16.9	17.0	17.0	17.4	16.7	17.0
E2E-ST-JT^m	16.3	15.2	16.5	15.1	16.3	16.9	16.8	15.6	16.1
E2E-ST-TDA^m	**14.9**	**14.1**	**15.7**	**14.4**	**15.2**	**15.4**	**16.5**	**14.9**	**15.1**

Table 3: WER scores of different methods on Must-C tst-COMMON set.

We implement these methods with small and medium model sizes respectively, in which we adopt superscripts s and m to represent the correspondent model size. Besides, we also compare E2E-ST-TDA with other E2E-ST baselines, which include using only MuST-C data and using external data: ESPnet ST and Cascaded (Inaguma et al. 2020), Fairseq ST and Multi-ST (Wang et al. 2020a), AFS (Zhang et al. 2020), Dual-Decoder (Le et al. 2020), W2V2-Transformer (Han et al. 2021), LNA-E,D (Li et al. 2021), Adapter Tuning (Le et al. 2021) and Chimera (Han et al. 2021).

Training Details and Evaluation. All experiments are implemented based on the FAIRSEQ[2] (Ott et al. 2019) toolkit. We adopt the transformer-based backbone for all models, consisting of 2 layers of one-dimensional convolutional layers with a down-sampling factor of 4, 12 Transformer encoder layers, and 6 Transformer decoder layers. More specifically, for the small model, we set the size of the self-attention layer, the feed-forward network, and the head to 256, 2048, and 4, respectively; for the medium model, the above parameters are set to 512, 2048, and 8, respectively. All models are initialized using the pre-trained ASR speech encoder to speed up the model convergence. During training, we use the adam optimizer (Kingma and Ba 2015) with a learning rate set to 0.002 to update model parameters with 10K warm-up updates. The label smoothing and dropout ratios are set to 0.1 and 0.3, respectively. In practice, we train all models with 2 Nvidia Tesla-V100 GPUs and it takes 1-2 days to finish the whole training. The batch size in each GPU is set to 10000, and we accumulate the gradient for every 4 batches. During inference, we average the model parameters on the 10 best checkpoints based on the performance of the MuST-C dev set, and adopt beam search strategy with beam size of 5. In our experiments, we report the WER score for ASR task and the case-sensitive BLEU score (Papineni et al. 2002) for ST task using sacreBLEU[3].

Main Results

E2E-ST Performance on MuST-C. We evaluate the E2E-ST performance of our proposed method on the MuST-C dataset with 8 languages. As illustrated in Table 2, we can observe that our approach E2E-ST-TDA significantly outperforms two baselines E2E-ST-Base and E2E-ST-JT, in all languages. More specifically, E2E-ST-TDA obtains an average BLEU score improvement of 1.4/1.8 respectively compared to E2E-ST-Base with different model sizes. These results demonstrate that our approach leverages triangular

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

[3]https://github.com/mjpost/sacrebleu, with a configuration of 13a tokenizer, case-sensitiveness, and full punctuation

Model	Params.	EN-DE	EN-FR	Avg.
MT	-	32.2	46.1	39.2
Cascaded ST	-	27.0	**38.6**	**32.8**
Chimera Mem-16	165M	25.6	35.0	30.3
Chimera	165M	26.3	35.6	31.0
E2E-ST-Base^m	74M	25.8	35.9	30.9
E2E-ST-TDA^m	76M	**27.1**	37.4	32.3

Table 4: BLEU scores on MuST-C tst-COMMON set in an extended setting. The external data includes 960 hours of LibriSpeech ASR data and WMT14 EN-DE/FR MT data.

decomposition agreement to fully exploit the triplet training data, leading to better translation performance. In addition, we include the results from previous work, such as ESPnet ST and Cascaded, Fairseq ST and Multi-ST, AFS, Dual-Decoder, W2V2-Transformer, LNA-E,D and Adapter Tuning. We can find that our implemented baseline E2E-ST-Base^s achieves similar performance as ESPnet ST and Fairseq ST, while our proposed method outperforms the cascaded system - ESPnet Cascaded trained on the same data. Compared with Dual-Decoder, our proposed method E2E-ST-TDA^m gains more remarkable improvement in all languages. Different from the Dual-Decoder that considers the interaction between loosely coupled ASR decoder and ST decoder, our method leverages the agreement of dual-paths with a shared decoder and saves the inference time. Besides, our approach outperforms some methods that use external data and multilingual versions. Instead of log-mel filter banks, W2V2-Transformer adopts pre-trained wav2vec 2.0 (Baevski et al. 2020) to improve the E2E-ST performance. Multilingual ST models, including Fairseq Multi-ST, LNA-E,D, and Adapting Tuning, beat most baselines since the target languages are mostly Indo-European languages with similar grammatical structures, which better help learn shared model parameters. As we can see, our proposed approach achieves state-of-the-art performance on all translation directions among all cascade and end-to-end systems in Table 2, which proves the effectiveness of our method.

ASR Performance on MuST-C. Since our method involves the ASR model during training, we also compare correspondent performance with two baselines: E2E-ST-Base and E2E-ST-JT. E2E-ST-Base denotes the performance of the pre-trained ASR model used to initialize all E2E-ST models. As illustrated in Table 3, our approach can significantly improve the performance on ASR tasks, reducing 1.5/1.9 WER scores over E2E-ST-Base with the small and medium model ,respectively. The performance improvement indicates that our proposed method can make full use of the entire training data by achieving better agreement in the training process to jointly improve the performance on E2E-ST and ASR tasks.

E2E-ST Performance on Extended Setting. We further verify the effectiveness of our proposed method with external ASR and MT data. Table 4 shows the results of all methods, including Cascaded ST, E2E-ST-Base, E2E-ST-TDA, and the recent SOTA method Chimera. For E2E-ST-Base and E2E-ST-TDA, we first train two MT models with the mixed data of WMT14 and MuST-C on EN-DE/EN-FR translation directions and then translate the transcriptions in Librispeech to build the additional triplet corpus. We also pre-train the ASR model on the mixed data of Librispeech and MuST-C to initialize all E2E-ST models. These ASR and MT models are used to perform Cascaded ST. The first two rows in the Table 4 show that the translation quality drops sharply when the output of the ASR model is fed as the input of the MT model compared with the clean transcription input. Our approach E2E-ST-TDA^m significantly surpasses E2E-ST-Base and Chimera under this setting, achieving a smaller parameter scale than Chimera at the same time. These experimental results prove that our proposed method can stably improve the translation quality of the E2E-ST system even with external data.

Model	BLEU			WER		
	EN-DE	EN-FR	Avg.	EN-DE	EN-FR	Avg.
E2E-ST-Base^m	23.5	33.8	28.7	16.8	16.9	16.9
+ WordKD	23.9	34.5	29.2	-	-	-
+ SeqKD	24.5	35.2	29.9	-	-	-
E2E-ST-TDA^m	**25.4**	**36.1**	**30.8**	**14.9**	**14.1**	**14.5**
w/o KL	23.8	34.5	29.2	16.3	16.0	16.2

Table 5: BLEU and WER scores of ablation study on MuST-C tst-COMMON set. “w/o” means without.

Analysis

Ablation Study. In order to analyze the effectiveness of different modules in our method, we carry out an ablation study on EN-DE and EN-FR translation directions in the MuST-C dataset. As shown in Table 5, besides E2E-ST-Base and E2E-ST-TDA, we evaluate the performance of three models: E2E-ST-Base with word-level or sequence-level knowledge distillation (WordKD/SeqKD) (Kim and Rush 2016) and E2E-ST-TDA without KL regularization terms. Actually, E2E-ST-TDA can be considered as the expansion of WordKD and SeqKD methods, while these approaches merely reduce the mismatch between MT and E2E-ST models. E2E-ST-TDA yields better translation results than E2E-ST-Base + WordKD/SeqKD, since our method can fully leverage the triangular relationship in training data. On the other hand, compared with E2E-ST-Base, the performance improvement of E2E-ST-TDA without KL regularization terms seems marginal. It indicates that optimizing the model with only MLE loss fails to fully utilize the association between triplet data.

Effect of Model Size. As illustrated in Table 2, the bigger model seems to obtain better improvement when using our method. To further verify the performance of our method with different model sizes, we conduct experiments on the MuST-C EN-DE dataset. We adopt dimensions ranging from $(256, 512, 768, 1024)$ for quick experiments. The detailed results are shown in Figure 3. From the figure, we can see that with the increase of the embedding dimension, the performance gain increases first and then remains stable, while

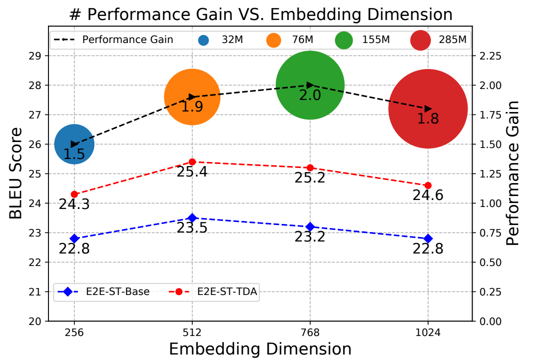

Figure 3: The impact of model size on performance. The area of the circle represents the parameter size of the model.

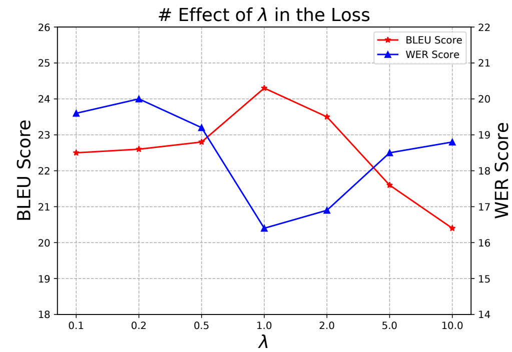

Figure 4: The impact of the hyper-parameter λ in the loss function on performance.

the model performance of both E2E-ST-Base and E2E-ST-TDA increase first and then decrease. It is because that a larger model (with a higher embedding dimension) typically requires more data for training, suffering from the overfitting risk and decreased efficiency.

Effect of Hyper-parameter λ. In our experiments, we attempt different settings ($\lambda = 0.1, 0.5, 1.0, 2.0, 5.0, 10.0$), and find that $\lambda = 1.0$ achieves the best BLEU and WER scores on MuST-C EN-DE development set. These results are shown in Figure 4. A larger λ will make the model pay more attention to triangular decomposition agreement, while a smaller λ will make the model pay more attention to optimizing the dual-path. In the early stage of model training, since the learned parameters are not perfect, a larger λ will cause the parameters of the model to be constrained in the wrong position by agreement earlier, making it challenging to further optimize the model. However, a smaller λ will make the dual-path too independent, and it is not easy to narrow the output representation.

Related Work

Speech Translation. Early ST methods (Ney 1999; Matusov, Kanthak, and Ney 2005; Sperber, Niehues, and Waibel 2017; Cheng et al. 2018) cascade the ASR subsystem and the MT subsystem. With the rapid development of deep learning, the neural networks widely used in ASR and MT have been adapted to construct a new end-to-end speech-to-text translation paradigm. However, due to the scarcity of triplet training data, developing an E2E-ST model that does not use intermediate transcription is still very challenging. Thus, various techniques have been proposed to ease the training process by using source transcriptions, including pre-training (Bansal et al. 2019; Wang et al. 2020c,b), multi-task learning (Weiss et al. 2017; Anastasopoulos and Chiang 2018; Sperber et al. 2019), meta-learning (Indurthi et al. 2020), interactive decoding (Liu et al. 2020), consecutive decoding (Dong et al. 2021a), and adapter tuning (Le et al. 2021). Among them, the pre-training strategy is a simple and effective way that pre-train different components of the ST system and merges them into one. However, the training process of these methods only loosely couples the two type two-tuple data. It does not fully explore the potential connections between the triplet data. Therefore, how to fully mining the associations between the scarce triplet data still remains a crucial issue to improve the performance of E2E-ST. Following this research line, we introduce TDA to fully exploit the association relationship in triplet data.

Agreement Regularization. One line of agreement regularization attempts to regularize model predictions to be invariant with minute perturbations on input data, which focused on semi-supervised learning areas. The minute perturbations can be random noise (Zheng et al. 2016), adversarial noise (Miyato et al. 2019; Carmon et al. 2019; Zhu et al. 2020; Jiang et al. 2020), gaussian noise (Aghajanyan et al. 2021) and various data augmentation methods (Ye et al. 2019; Xie et al. 2020). Another line tries to take into consideration the agreement between the different models, especially in sequence modeling. For instance, there are some attempts in speech recognition (Mimura, Sakai, and Kawahara 2018), neural machine translation (Liu et al. 2016; Zhang et al. 2019b), and speech synthesis (Zheng et al. 2019), which try to improve the performance by integrating the predicted probability from forward and backward decoding sequences. Our method is most similar to the latter, but we aim to constrain the agreement between the probability distributions of two directions sequences in a single unified model.

Conclusion

In this paper, we propose a flexible and effective regularization method for the ST task, namely Triangular Decomposition Agreement (TDA), which relies on the agreement between inherent and unexplored dual decomposition paths ASR-MT and ST-BT of ST. In our method, two Kullback-Leibler divergences are added to the standard training objective as regularization terms to resolve the mismatch between the joint probability distributions of dual-path ASR-MT and ST-BT. In addition, our approach can use two paths for inference and adopt early termination methods for different scenarios to ensure efficient inference speed. Empirical evaluations of the eight translation directions of MuST-C demonstrate that our proposed approach leads to significant improvements compared with strong baseline systems.

Acknowledgements

This research is supported by grants from the National Natural Science Foundation of China (Grant No.62072423) and National Major Scientific Instruments and Equipments Development Project (Grant No.61727809). This work is also supported by Alibaba Innovative Research Program. We appreciate Dongqi Wang, Linan Yue, Dexin Wang and Linlin Zhang for the fruitful discussions. We thank the anonymous reviewers for helpful feedback on early versions of this work. This work is done during the first author's internship at Alibaba DAMO Academy.

References

Aghajanyan, A.; Shrivastava, A.; Gupta, A.; Goyal, N.; Zettlemoyer, L.; and Gupta, S. 2021. Better Fine-Tuning by Reducing Representational Collapse. In *ICLR*.

Anastasopoulos, A.; and Chiang, D. 2018. Tied Multitask Learning for Neural Speech Translation. In *NAACL*.

Baevski, A.; Zhou, H.; rahman Mohamed, A.; and Auli, M. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *Advances in Neural Information Processing Systems*.

Bansal, S.; Kamper, H.; Livescu, K.; Lopez, A.; and Goldwater, S. 2019. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *NAACL*.

Bojar, O.; Buck, C.; Federmann, C.; Haddow, B.; Koehn, P.; Leveling, J.; Monz, C.; Pecina, P.; Post, M.; Saint-Amand, H.; Soricut, R.; Specia, L.; and Tamchyna, A. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *WMT@ACL*.

Carmon, Y.; Raghunathan, A.; Schmidt, L.; Liang, P.; and Duchi, J. C. 2019. Unlabeled Data Improves Adversarial Robustness. In *NeurIPS*.

Cheng, Y.; Tu, Z.; Meng, F.; Zhai, J.; and Liu, Y. 2018. Towards Robust Neural Machine Translation. In *ACL*.

Dong, Q.; Wang, M.; Zhou, H.; Xu, S.; Xu, B.; and Li, L. 2021a. Consecutive Decoding for Speech-to-text Translation. In *AAAI*.

Dong, Q.; Ye, R.; Wang, M.; Zhou, H.; Xu, S.; Xu, B.; and Li, L. 2021b. Listen, Understand and Translate: Triple Supervision Decouples End-to-end Speech-to-text Translation. In *AAAI*.

Gangi, M. A. D.; Cattoni, R.; Bentivogli, L.; Negri, M.; and Turchi, M. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *NAACL*.

Han, C.; Wang, M.; Ji, H.; and Li, L. 2021. Learning Shared Semantic Space for Speech-to-Text Translation. In *Findings of ACL*.

Inaguma, H.; Kiyono, S.; Duh, K.; Karita, S.; Yalta, N.; Hayashi, T.; and Watanabe, S. 2020. ESPnet-ST: All-in-One Speech Translation Toolkit. In *ACL*.

Indurthi, S.; Han, H.; Lakumarapu, N. K.; Lee, B.; Chung, I.; Kim, S.; and Kim, C. 2020. End-end Speech-to-Text Translation with Modality Agnostic Meta-Learning. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7904–7908.

Iranzo-Sánchez, J.; Giménez-Pastor, A.; Silvestre-Cerdà, J.; Baquero-Arnal, P.; Saiz, J. C.; and Juan-Císcar, A. 2020. Direct Segmentation Models for Streaming Speech Translation. In *EMNLP*.

Jiang, H.; He, P.; Chen, W.; Liu, X.; Gao, J.; and Zhao, T. 2020. SMART: Robust and Efficient Fine-Tuning for Pretrained Natural Language Models through Principled Regularized Optimization. In *ACL*.

Kim, Y.; and Rush, A. M. 2016. Sequence-Level Knowledge Distillation. In *EMNLP*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Le, H.; Pino, J.; Wang, C.; Gu, J.; Schwab, D.; and Besacier, L. 2020. Dual-decoder Transformer for Joint Automatic Speech Recognition and Multilingual Speech Translation. In *COLING*.

Le, H.; Pino, J.; Wang, C.; Gu, J.; Schwab, D.; and Besacier, L. 2021. Lightweight Adapter Tuning for Multilingual Speech Translation. In *ACL/IJCNLP*.

Li, X.; Wang, C.; Tang, Y.; Tran, C.; Tang, Y.; Pino, J.; Baevski, A.; Conneau, A.; and Auli, M. 2021. Multilingual Speech Translation from Efficient Finetuning of Pretrained Models. In *ACL/IJCNLP*.

Liu, L.; Utiyama, M.; Finch, A.; and Sumita, E. 2016. Agreement on Target-bidirectional Neural Machine Translation. In *NAACL*.

Liu, Y.; Xiong, H.; He, Z.; Zhang, J.; Wu, H.; Wang, H.; and Zong, C. 2019. End-to-End Speech Translation with Knowledge Distillation. In *INTERSPEECH*.

Liu, Y.; Zhang, J.; Xiong, H.; Zhou, L.; He, Z.; Wu, H.; Wang, H.; and Zong, C. 2020. Synchronous Speech Recognition and Speech-to-Text Translation with Interactive Decoding. In *AAAI*.

Matusov, E.; Kanthak, S.; and Ney, H. 2005. On the integration of speech recognition and statistical machine translation. In *INTERSPEECH*.

Mimura, M.; Sakai, S.; and Kawahara, T. 2018. Forward-Backward Attention Decoder. In *INTERSPEECH*.

Miyato, T.; Maeda, S.; Koyama, M.; and Ishii, S. 2019. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41: 1979–1993.

Ney, H. 1999. Speech translation: coupling of recognition and translation. *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, 1: 517–520 vol.1.

Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *NAACL*.

Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: An ASR corpus based on public domain audio books. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5206–5210.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL*.

Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *INTERSPEECH*.

Sperber, M.; Neubig, G.; Niehues, J.; and Waibel, A. 2017. Neural Lattice-to-Sequence Models for Uncertain Inputs. In *EMNLP*.

Sperber, M.; Neubig, G.; Niehues, J.; and Waibel, A. 2019. Attention-Passing Models for Robust and Data-Efficient End-to-End Speech Translation. *Transactions of the Association for Computational Linguistics*, 7: 313–325.

Sperber, M.; Niehues, J.; and Waibel, A. 2017. Toward robust neural machine translation for noisy input sequences. In *International Workshop on Spoken Language Translation (IWSLT)*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 5998–6008.

Wang, C.; Tang, Y.; Ma, X.; Wu, A.; Okhonko, D.; and Pino, J. 2020a. Fairseq S2T: Fast Speech-to-Text Modeling with Fairseq. In *AACL*.

Wang, C.; Wu, Y.; Liu, S.; Yang, Z.; and Zhou, M. 2020b. Bridging the Gap between Pre-Training and Fine-Tuning for End-to-End Speech Translation. In *AAAI*.

Wang, C.; Wu, Y.; Liu, S.; Zhou, M.; and Yang, Z. 2020c. Curriculum Pre-training for End-to-End Speech Translation. In *ACL*.

Weiss, R. J.; Chorowski, J.; Jaitly, N.; Wu, Y.; and Chen, Z. 2017. Sequence-to-Sequence Models Can Directly Translate Foreign Speech. In *INTERSPEECH*.

Xie, Q.; Dai, Z.; Hovy, E.; Luong, T.; and Le, Q. 2020. Unsupervised Data Augmentation for Consistency Training. *NeurIPS*, 33.

Ye, M.; Zhang, X.; Yuen, P.; and Chang, S.-F. 2019. Unsupervised Embedding Learning via Invariant and Spreading Instance Feature. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6203–6212.

Zhang, B.; Titov, I.; Haddow, B.; and Sennrich, R. 2020. Adaptive Feature Selection for End-to-End Speech Translation. In *Findings of EMNLP*.

Zhang, P.; Chen, B.; Ge, N.; and Fan, K. 2019a. Lattice Transformer for Speech Translation. In *ACL*.

Zhang, Z.; Wu, S.; Liu, S.; Li, M.; Zhou, M.; and Chen, E. 2019b. Regularizing Neural Machine Translation by Target-bidirectional Agreement. In *AAAI*.

Zheng, S.; Song, Y.; Leung, T.; and Goodfellow, I. 2016. Improving the Robustness of Deep Neural Networks via Stability Training. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4480–4488.

Zheng, Y.; Wang, X.; He, L.; Pan, S.; Soong, F.; Wen, Z.; and Tao, J. 2019. Forward-Backward Decoding for Regularizing End-to-End TTS. In *INTERSPEECH*.

Zhu, C.; Cheng, Y.; Gan, Z.; Sun, S.; Goldstein, T.; and Liu, J. 2020. FreeLB: Enhanced Adversarial Training for Natural Language Understanding. In *ICLR*.