Improving End-to-End Speech Translation by Leveraging Auxiliary Speech and Text Data

Yuhao Zhang[1], Chen Xu[1], Bojie Hu[3], Chunliang Zhang[1,2], Tong Xiao[1,2*], Jingbo Zhu [1,2]
[1] School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2] NiuTrans Research, Shenyang, China
[3] Tencent Minority-Mandarin Translation, Beijing, China
yoohao.zhang@gmail.com, xuchenneu@outlook.com, bojiehu@tencent.com,
{zhangchunliang, xiaotong, zhujingbo}@mail.neu.edu.cn

Abstract

We present a method for introducing a text encoder into pre-trained end-to-end speech translation systems. It enhances the ability of adapting one modality (i.e., source-language speech) to another (i.e., source-language text). Thus, the speech translation model can learn from both unlabeled and labeled data, especially when the source-language text data is abundant. Beyond this, we present a denoising method to build a robust text encoder that can deal with both normal and noisy text data. Our system sets new state-of-the-arts on the MuST-C En-De, En-Fr, and LibriSpeech En-Fr tasks.

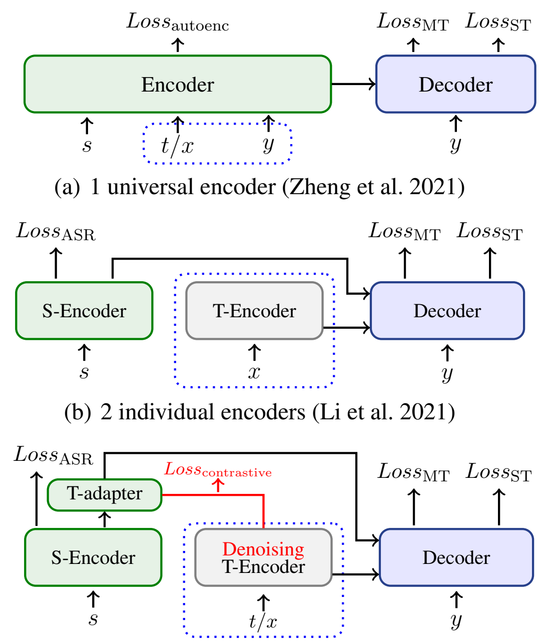
Introduction

In Speech Translation (ST), End-to-End (E2E) neural approaches have gained attraction as a promising line of research towards systems with lower latency and less error propagation. However, developing models of this type can be challenging because the aligned speech-to-translation data is scarce (Wang et al. 2020b; Dong et al. 2021b; Zheng et al. 2021; Tang et al. 2021a). This leads researchers to explore methods that resort to large-scale unlabeled data. A simple one is to use pre-trained models to encode acoustic and/or textual input (Pino et al. 2020; Ye, Wang, and Li 2021), whereas others train ST models using additional data of either Automatic Speech Recognition (ASR) or Machine Translation (MT), or both (Wang et al. 2020c; Xu et al. 2021; Indurthi et al. 2021).

Such a paradigm provides an opportunity to make use of both labeled and unlabeled data, say, the speech, text, speech-to-transcription, text-to-text, and speech-to-text data. For example, one can feed all available data into an autoencoder to train ST models (Zheng et al. 2021). More recently, it has been found that stronger results can be achieved by using an explicit text encoder to ease the training on the MT data (Li et al. 2021).

Here, we take a further step towards more effective use of both labeled and unlabeled data in ST. We claim that the source-language text encoder plays an important role in leveraging ASR and MT although it is not involved in standard end-to-end ST. We then develop a method (named as Multi-Step Pre-training for Speech Translation, or MSP-ST for short) to expose the text encoder to both ASR and MT learning processes, and force them to assist each other. Having the text encoder as the bridge between ASR and MT is perhaps helpful: the result ST system can learn acoustic and textual encoding simultaneously (see Figure 1). Note that such a design also addresses the role mismatch problem wherein the pre-trained ASR encoder does not behave like what the target-language decoder expects (Wang et al. 2020b; Xu et al. 2021). To our knowledge, this is the first to discuss the problem in large-scale pre-training with all ASR, MT and ST data on end-to-end ST tasks.

Another improvement is that we denoise the text encoder so that it is robust to the noisy transcription-like input. In this



(a) 1 universal encoder (Zheng et al. 2021)

(b) 2 individual encoders (Li et al. 2021)

(c) 2 correlated encoders + denoising (this work)

Figure 1: Model architectures. Dotted boxes mean that the items are dropped in ST tuning and inference. s = acoustic signal sequence, t = transcription, x = source-language word sequence, and y = target-language word sequence.

*Corresponding author.

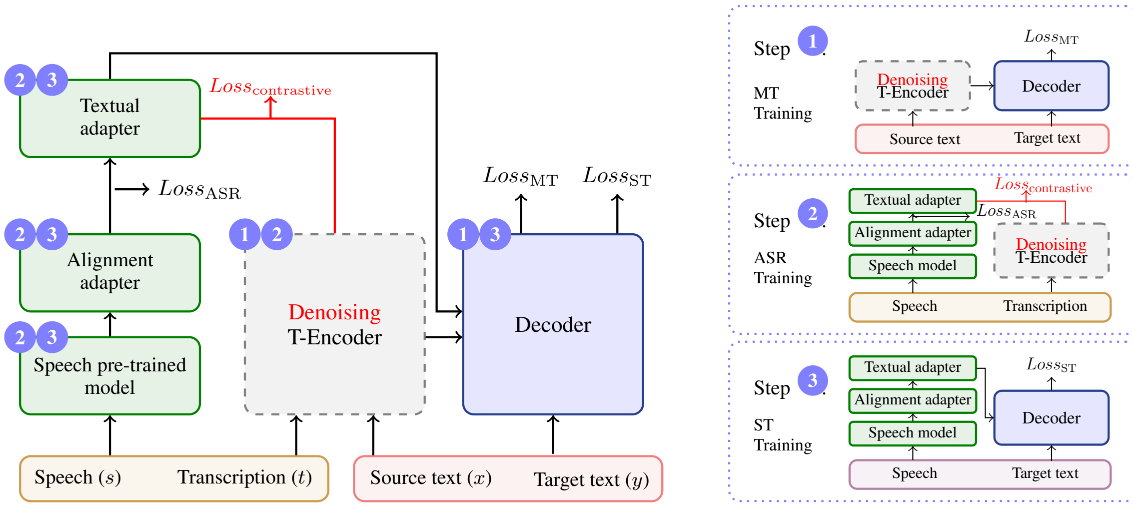


Figure 2: The end-to-end speech translation architecture with a text encoder. Circled numbers indicate training steps. The denoising text-encoder (T-encoder) will be dropped during Step 3.

way, the text encoder can deal with both the normal text and the transcription. This is beneficial when the text encoder is used to supervise the learning of the ST encoder, where the speech-to-transcription data is the input.

We implement our method in a Transformer-based ST system. On the MuST-C and LibriSpeech tasks, it outperforms very strong baselines significantly. It achieves BLEU scores of 30.0 and 40.6 on the MuST-C En-De and En-Fr data and a BLEU score of 21.4 on the LibriSpeech En-Fr data. These results are new state-of-the-arts on these tasks. The performance is even comparable with that of the unrestricted MT system on the LibriSpeech task. We implement our method in the submitted system for IWSLT 2022 offline speech translation task (Zhang et al. 2022a).

Method

Our ST model is a standard encoder-decoder model, following the Transformer model (Vaswani et al. 2017). The encoder reads a sequence of source-language acoustic signals, and the decoder produces a sequence of target-language words. Broadly speaking, like any encoder-decoder model, one can train this architecture in a standard pre-training + fine-tuning fashion (Lewis et al. 2020). For example, the encoder is pre-trained by pure acoustic data (Baevski et al. 2020), and/or enhanced by training an ASR encoder on speech-to-transcription data. Likewise, the decoder is initialized by pre-trained models (for either the word embedding component or the whole decoding network). The final ST model is tuned on the labeled data, i.e., pairs of speech and translation.

But such a model does not accept source-language text as input, and it is non-trivial to learn the model on source-language text data. One way to use textual input is to have a sub-model, implicit or explicit, to introduce source-language text signals into the ST model. To this end, we develop a text encoder on the source-language side in addition to the ST encoder. In pre-training, it works with both the ST encoder

Order	Name	Data Type	Trained Model	Training
1	Init.	s	s-enc.	pre-train
		x	t-enc.	pre-train
		y	dec.	pre-train
2	MT	(x, y)	t-enc + dec.	pre-train
3	ASR	(s, t)	s-enc + t-enc.	pre-train
4	ST	(s, y)	s-enc. + dec.	fine-tune

Table 1: Data types used in training. s-enc. = ST encoder, t-enc = text encoder, and dec. = ST decoder.

and decoder. After that, the text encoder is absent, and the ST model is tuned and then used for inference, as usual.

Formally, let s be an acoustic signal sequence, t be a transcription of s, x be a source-language word sequence, and y be a target-language word sequence. There are many choices to build different types of training data. For example, (s, y) is the standard ST data, (x, y) is the MT data, x is the monolingual data. Table 1 shows the data types used here, ordered by the training pipeline of our method. Note that the term "pre-training" is used in many different ways. In this paper, the term refers to any training process other than the final tuning of the ST model on (s, y)[1].

Another note on notation. Not all these sequences are required to come in pairs. For example, x in the monolingual data might not appear in the MT data. Here we use this notation to emphasize what type of data is used in training, but not the actual data.

At the heart of our system is a design to guide the ST model via textual information. Two intuitions form the basis of this work:

[1]Training on (x, y) and (s, t) is actually a "tuning" process on the initialized/pre-trained model. Here we call them pre-training to avoid the misuse of "tuning" because it is typically used when tuning the model on the labeled target-task data.

- The text encoder can supervise the training of the ST encoder so that the behavior of the ST encoder is more consistent with that of a standard MT encoder.
- The text encoder can be robust to ASR noise, and can accept transcription as input.

To make use of these intuitions, we improve the ST encoder and develop a contrastive training method to incorporate the text encoder into the ASR-based training. Beyond this, we propose a denoising method to learn a text encoder that is robust to either normal text or transcription.

ASR Training with Text Encoder

ST encoders in general share a similar model structure with ASR encoders. An advantage of the ASR-based design for ST encoders is that it is better suited for processing acoustic signals and off-the-shelf pre-trained acoustic models are straightforwardly available to ST. However, the ASR-like encoder does not work with the target-language text decoder because the decoder wants text-friendly encoding instead of the acoustic encoding (Dong et al. 2021b; Xu et al. 2021). A way to address this modality-inconsistency issue is to stack adapters on top of the acoustic model. Thus, the system can learn to transform from one modality to another. However, it remains undesirable that the supervision of the encoder is only from the decoder and the vast number of source-language sentences are ignored.

We propose to use the text encoder to supervise the training of the ST encoder. See Figure 2 for the model architecture. The core design is the adapters for the ST encoder and the contrastive learning for the two encoders.

Adapters for ST Encoding For ST encoding, Connectionist Temporal Classification (CTC) based training is necessary for state-of-the-art performance (Graves et al. 2006). A common way is to add the CTC-based loss to the acoustic model. Then, an optional adapter can be used to map the acoustic model output to representations that the text decoder prefers (Xu et al. 2021), or shrinks the length of the acoustic model output (Li et al. 2021).

In our preliminary experiments, we found that it was not easy to do alignment in CTC-based training due to the big length difference between the acoustic model output and the word sequence. Thus, we propose an alignment adapter and place it between the acoustic model and the CTC-based loss. The adapter consists of n convolutional networks to shorten the sequence and a Conformer layer (Chen et al. 2021) to filter the down-sampling output. To make a stronger correlation with the text encoder, we share the same vocabulary and the output layer to predict each word in the representation space of the textual model when generating the CTC path. This way forces the acoustic representation space to align to that of the text encoder.

Another encoding network (call it textual adapter) is stacked upon the alignment adapter. It consists of a single self-attention layer. We add the position embedding before feeding the feature into this adapter to fuse location information. The textual adapter is intended to reduce the impact of blank noise and produces a more text encoder-like output, which is better suited for the input of the decoder.

Contrastive Training We train the ST encoder with the text encoder in addition to the supervision signal from the decoder side. This is a step before we fine-tune the ST model. Here we choose contrastive training as a way to connect the ST encoder and the text encoder. More formally, let $\mathcal{A}(s)$ be the output of the ST encoder given the speech s, and $\mathcal{M}(t)$ be the output of the pre-trained text encoder given the transcription t. Given a set of training samples $\{(s_i, t_i)\}$, the loss function of the contrastive training is defined to be:

$$\mathcal{L}_{\mathrm{CL}} = -\sum_{(s_i,t_i)} \log \frac{e^{\pi(\mathcal{A}(s_i),\mathcal{M}(t_i))/\tau}}{\sum_{t_j:j\neq i} e^{\pi\left(\mathcal{A}(s_i),\mathcal{M}\left(t_j\right)\right)/\tau}} \tag{1}$$

where $\pi(\cdot,\cdot)$ is a function that computes the similarity of the input vectors. Here we choose the cosine function for $\pi(\cdot,\cdot)$ and average pooling the two sequence representations. τ is a scaler to control the sharpness of the function output. For each s_i, we have its labeled transcription to form a positive sample (s_i, t_i). Also, we use transcriptions other than t_i (i.e., t_j for $j \neq i$) to form negative samples. Eq. 1 distinguishes the positive sample from the negative samples (i.e., $\{(s_i, t_j)|j \neq i\}$).Thus, $\mathcal{A}(s_i)$ would be close to $\mathcal{M}(t_i)$ and far way from other $\mathcal{M}(t_j)$.

For more diverse training samples, we decode a transcription t'_i by keeping blank labels in the output of the alignment adapter. For (s_i, t'_i), we compute a loss $\mathcal{L}'_{\mathrm{CL}}$ as in Eq. 1. The final loss function of ASR training is defined as:

$$\mathcal{L}_{\mathrm{ASR}} = \mathcal{L}_{\mathrm{CTC}} + \alpha(\beta\mathcal{L}_{\mathrm{CL}} + (1-\beta)\mathcal{L}^{'}_{\mathrm{CL}}) \tag{2}$$

where $\mathcal{L}_{\mathrm{CTC}}$ is the CTC loss (Watanabe et al. 2017) which is widely used in ST task (Wang et al. 2020b; Dong et al. 2021b; Xu et al. 2021), and α and β are coefficients for interpolation.

For guiding the textual adapter by contrastive training, the text encoder needs to suit the decoder and be well-trained. So we run the ASR training process after the MT training process and freeze the text encoder during ASR training.

Denoising Text Encoder

There are two jobs for the text encoder: 1) encoding real source-language sentences in MT training, 2) encoding transcriptions in ASR training.

As MT training is prior to ASR training, the text encoder is primarily trained to address the first point. This is potentially undesirable for a reason: in ASR training, the input of the text encoder is a transcription, which often contains symbols that never appear in MT data. The input will be more noisy if we use self-generated transcriptions in training (see Eq. 2).

We use denoising methods for a robust text encoder, of which the simplest one is to use a denoising autoencoder (DAE) to take noise into account (Lewis et al. 2020). Here we choose mBART as the initial model (Liu et al. 2020a) for its potential cross-lingual ability. It is complicated to update mBART for introducing ASR-related noise (such as blank symbols) into DAE training. We therefore further denoise the encoder in the MT training phase to make a Silence Insensitive DAE (SIDAE). Our method is inspired by Consistency Regularization (Zhang et al. 2020). In consistency

Models	Speech	Text	ASR	MT	MuST-C En-De	MuST-C En-Fr	LibriSpeech En-Fr
Unrestricted MT (Xu et al. 2021)	-	-	-	-	31.1	41.9*	21.3
Transformer (Wang et al. 2020a)	-	-	-	-	22.7	32.9	16.7
FAT-ST (Big) (Zheng et al. 2021)	✓	✓	✓	✓	25.5	-	-
VggTLarge (Pino et al. 2020)	✓	-	✓	-	25.6	-	-
LUT (Dong et al. 2021b)	-	✓	✓	-	-	-	18.3
Chimera (Han et al. 2021)	✓	-	-	✓	26.3	35.6	19.4
JT (Tang et al. 2021a)	-	-	-	✓	26.8	37.4	-
LNA-ED-Adapt (Gállego et al. 2021)	✓	✓	✓	✓	27.3	-	-
XSTNET (Ye, Wang, and Li 2021)	✓	-	-	✓	27.8	38.0	-
SATE (Xu et al. 2021)	-	-	✓	✓	28.1	-	20.8
TCN (Indurthi et al. 2021)	-	-	✓	✓	28.9	-	-
STPT (Tang et al. 2022)	✓	-	✓	✓	-	39.7	-
Baseline	✓	✓	✓	✓	27.5	38.6	20.8
MSP-ST	✓	✓	✓	✓	30.0	40.6	**21.4**
MSP-ST†	✓	✓	✓	✓	**30.2**	**40.8**	-

Table 2: Performance on different data set. The baseline is LNA (Li et al. 2021) and add an additional adapter (Gállego et al. 2021). * represents that we reproduce the result. † uses the HuBERT (Hsu et al. 2021) as the pre-trained acoustic encoder.

regularization, a "good" model should be less sensitive to perturbation on the input. We design a perturbation function $g(\cdot)$ that randomly adds blank symbols into source-language sentences. The size of adding blank is decided by the coefficient r multiply the length of sentence. For each sentence pair (x, y), we expect that the MT system can produce a correct prediction given both x and $g(x)$ as input. The loss function on a set of source-text and target-text pairs $\{x_i, y_i\}$ is given by:

$$\mathcal{L}_{\mathrm{MT}} = -\sum_{(x_i,y_i)} \log \mathrm{P}(y_i \mid x_i) + \log \mathrm{P}(y_i \mid g(x_i)) \tag{3}$$

where $\mathrm{P}(y \mid \cdot)$ is the MT system consisting of the text encoder and the text decoder.

Experiments

Data

We run our experiments on English to German (En-De) and English to French (En-Fr) translation tasks. For speech data, we use the Librilight (Kahn et al. 2020) which consists of about 60k hours of unlabelled speech. For text data, we follow Liu et al. (2020a)'s work which covers 25 languages.

We use LibriSpeech 960 hours (Panayotov et al. 2015) to train the pre-trained acoustic model on the English ASR task. To adapt the DAE model to the MT task, we use the Opensubtitle En-De and WMT14 En-Fr datasets respectively. The final data consists of 18M sentence pairs for the En-De translation. For En-Fr translation, we extract 10M sentence pairs from the WMT14 En-Fr data, following Xu et al. (2021)'s work. We use sentencepiece to segment the untokenized text into sub-words[2]. The sentencepiece model and the vocabulary are the same as in Liu et al. (2020a)'s work and we remove words which do not appear in all the corpora. The vocabulary size is set to 32K for the MuST-C tasks (Di Gangi et al. 2019) and 25K for the LibriSpeech En-Fr task (Kocabiyikoglu, Besacier, and Kraif 2018).

The MuST-C En-De and En-Fr tasks provide 400 hours and 484 hours speech data respectively. For the LibriSpeech En-Fr task, the size of the training set is 100 hours. Readers can refer to Appendix for more details about the information and processing of data.

Model Settings

We implement our systems by the Fairseq toolkit (Ott et al. 2019). For pre-training of unlabeled speech data, we use the open-source wav2vec2 model. For the DAE model, we also utilize the open-source mBART.CC25 model. For a stronger baseline, we re-implement the LNA method (Li et al. 2021). Following Gállego et al. (2021)'s work, we add an adapter (Bapna and Firat 2019) to mitigate the gap between the acoustic and textual model. We use speech as input for our pre-trained model.

For pre-training of SIDAE, we set the coefficient r to 0.3. We stop training until the perplexity converges on the validation set. For the alignment adapter, the size of the convolutional layer n is set to 3. For each Conformer layer, there are 1,024 hidden states, 16 attention heads and 4,096 FFN hidden states. We freeze the pre-trained acoustic model in the first 5,000 training steps to warm up the two adapters. The τ and α are set to 0.1 and 0.3. The initial value of β is 1. It then decreases by 0.1 per 5,000 steps until 0. For fine-tuning on the ST task, we use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We use dropout ($p = 0.1$) and label smoothing ($p = 0.1$) for robust training. We early stop the training if the last five checkpoints do not improve. For training settings of unrestricted MT, we follow Xu et al. (2021)'s work.

When evaluating the model, we average the weight of the last five checkpoints. For inference, the beam size is set to 4 and the length penalty is set to 1.0. We use SacreBLEU (Post 2018) to evaluate the performance. Following previous work, we report the case-sensitive score for the MuST-

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

Model	En-De	En-Fr
Baseline	27.5	38.6
+ Alignment adapter	28.1	38.8
+ Textual adapter, CL	29.1	39.8
+ KDCL	29.5	40.2
+ SIDAE	30.0	40.6

Table 3: Ablation study on the MuST-C tasks.

Model	ST data	Utterances	Test
Transformer	65h	39K	6.4
Transformer	400h	230K	22.7
MSP-ST	10h	5K	15.9
MSP-ST	65h	39K	24.3
MSP-ST	400h	230K	30.0

Table 4: Sample efficiency on the MuST-C En-De task.

C En-De and En-Fr tasks and case-insensitive score for the LibriSpeech En-Fr task.

Results

Table 2 shows our experimental results. We see, first of all, that our baselines which utilize all types of data are very strong and achieve the SOTA performance on LibriSpeech En-Fr tasks. While on the En-De task, the baseline fails to outperform the methods without using unlabeled data (Xu et al. 2021). This proves previous result that using unlabeled data has more room for improvement.

Our method gains remarkable improvements on two MuST-C tasks compared with the baseline and achieves the SOTA results without using any ST data-augmentation method. Though our method only gains a +0.6 BLEU improvement on the LibriSpeech En-Fr task, it is comparable with the MT baseline. Compared with Xu et al. (2021)'s work, our method shows a +1.9 higher BLEU score by using additional unlabeled data. In particular, we use much less labeled data compared with TCN (Indurthi et al. 2020) and still yields a 1.1 BLEU improvement. This also verifies the potential of unlabeled data in ST. Compared with the STPT (Tang et al. 2022), we utilize the additional monolingual text data to build the denoising model and less MT data. Our method gains +0.9 BLEU improvement and shows the importance of denoising ability to the ST model. The MSP-ST achieves a 2.5 BLEU improvement compared with the baseline (Li et al. 2021) due to the efficient use of unlabeled speech and text data.

Analysis

Ablation Study

We replace the adapters in the baseline system with our alignment adapter. Table 3 shows that the alignment adapter can achieve better performance. It indicates our alignment adapter is a more effective way to convert the representation space of the acoustic model to text model. Then, we introduce our textual adapter into the system and align it with the output of the DAE encoder by the contrastive loss. The results show that the textual adapter is the important for satisfactory performance. Also, this result confirms that the semantic conversion and denoising methods are important for ST. We introduce L'_{CL} (denoted as KDCL) into training (see Section). It shows that knowledge distillation can reduce the difficulty of semantic learning. The advances brought by the textual adapter and KDCL are the same apparently on the two tasks because the methods improve the acoustic model and use the similar speech data. We finally use SIDAE to replace DAE and mitigate the impact of blank noise for the textual adapter. As expected, it helps. The final results achieve new SOTA performance on the two MuST-C tasks.

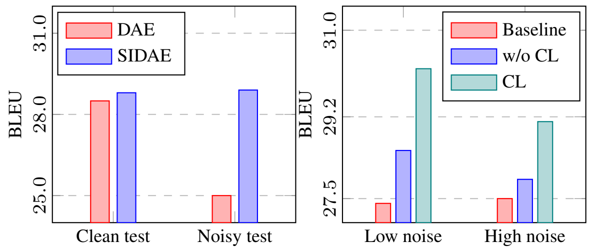
Effect of Denoising

Figure 3 (a) compares BLUE scores of DAE and SIDAE. We use the blank label to replace punctuations in the source text to get the noisy test set. The performance on the clean test is almost the same. The modest improvement of the SIDAE model may be due to the stronger generalization ability by noisy training. When the test text contains many blank labels, the vanilla DAE model performs worse while the SIDAE is robust to the noise. To explore the denosing influence on ST, we split the test set into two sets according to whether the blank label ratio is higher than 0.3. Figure 3 (b) shows the performance of different systems on the test sets. Here "w/o CL" and "CL" mean that the textual adapter is trained without and with contrastive loss. Its improvement is modest on the high noise test set, while our textual adapter achieves a bigger BLEU improvement.

We further explore why the SIDAE model is not impacted so much by blank symbols. As Figure 4 shows, the self-attention weight of blank label focus on all blank labels, which means that the output of this position is only with a blank message and it is easy to be recognized in the cross-attention module. The attention weights of cross-attention confirm our conjecture, the position of silent speech has a very low weight. Thus, the blank noise can not affect the interference process. In the rest of this paper, we use the SIDAE model to guide the textual adapter.

Impact of Alignment Adapter

Here we show the effectiveness of the alignment adapter. We select several high-frequency words from a random sentence and calculate the cosine similarities of words between the acoustic model and textual model in Figure 5. For cross-modal (cross-lingual) similarity, it's the similarity between the output of the alignment adapter and source (target) token embedding. The baseline model does not consider the alignment of the acoustic model and the text encoder. Both the cross-modal and cross-lingual similarities are almost around zero. The inconsistency of representation space aggravates the gap between acoustic and textual models. The alignment adapter boosts the alignment between the two models and can reduce the difficulty of contrastive learning because the adapter does not need to consider the transfer of the representation space. Because of the cross-lingual nature of multilingual DAE, the cross-language alignment can also better facilitate language transfer.



(a) Performance on clean and noisy test set

(b) Performance on different ratio of noise

Figure 3: Comparison of denoising or not on MuST-C En-De test.

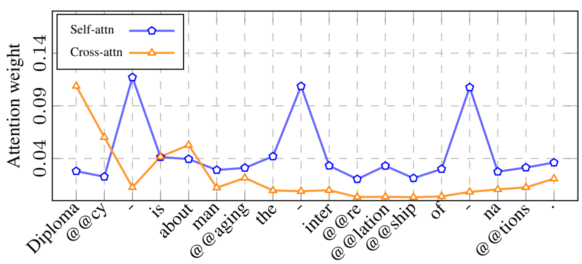
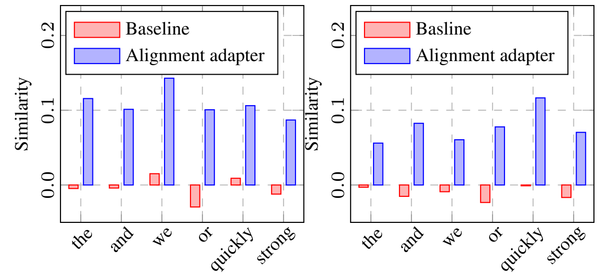


Figure 4: Blank self-attention and other word cross attention. "_" represents the blank token.



(a) Cross modal alignment

(b) Cross lingual alignment

Figure 5: Similarity of cross-modal and cross-lingual between acoustic and textual modal.

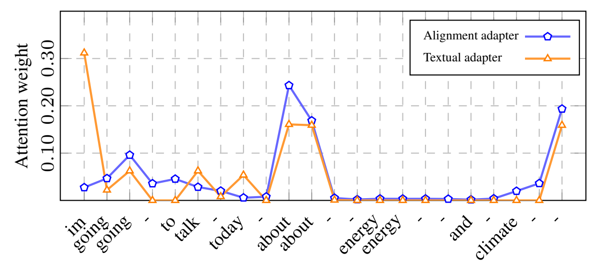


Figure 6: The self-attention weights of alignment adapter and textual adapter.

Impact of Textual Adapter

To study the impact of the textual adapter, we compare the attention wights between the alignment adapter and the textual adapter. Figure 6 shows that the textual adapter is helpful in adapting the ST encoder to a text-friendly encoder. The weight in the 1st position shows that the textual adapter learns information which may be unimportant for the cross-modal stage. This proves the difference between the acoustic model and textual model in encoding the input. Further, the adapter focuses more on the first position which is more important at the stage of translation. This indicates that the adapter learns something better suited to the MT model. Figure 6 also shows that in many blank positions, the weights are lower than those of the alignment adapter.

Figure 7 (a) shows our textual adapter can significantly mitigate the gap between the acoustic and textual model. Figures 7 (b) and (c) show the average information entropy (IE) of attention weights. Note the IE also has noise information. The IE of the textual adapter is much lower due to the inattention of noise. The IE of w/o CL is the highest, because its adapter learns more semantic information but fails to remove the noise. Figure 7 (c) shows the usage of adapter can boost the decoder to extract more information. It also proves that our gains are mainly from the improvement of encoder since the IEs of the two adapters are similar.

Sample Efficiency

We study how different systems behave under different sized speech translation data. To do this, we scale the training data by about 6 times each time. Table 4 shows that our model obtains a good speech translation result by only 10-hour labeled data, which is better than vanilla Transformer (Wang et al. 2020a) learned on 65-hour labeled data. The improvement is still significant when more data is used. This phenomenon shows that the MSP-ST method can build a decent system under the condition of insufficient ST data.

Parameter Efficiency

Using pre-training models in general leads to a significant increase of model parameters. To evaluate the efficiency of model size, we compare the performance and parameter number of different methods in Figure 8. The upper left of the figure means a higher efficiency. We see that our method is efficient: it achieves the best BLEU score with a slight increase of the number of the parameters. On the performance side, the models which stack the pre-trained acoustic and the whole SIDAE model also achieve comparable performance with our MSP-ST system. But our model is more parameter efficient for dropping the pretrained textual encoder at the ST tuning stage. The w/o CL model with the same parameters performs much worse compared with MSP-ST due to the lack of the denoising ability. We implement the SATE method by using the same data and model size. The SATE does not outperform the *Stack* and MSP-ST methods, we conjecture that the alignment adapter has bridged the gap between two pretrained models.

Impact of Data

We remove the additional unlabeled data and implement the MSP-ST method in Table 5. To make a more fair comparison with other work, we use a small-scale parameter

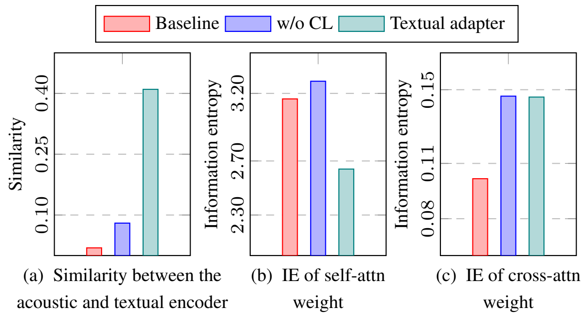


Figure 7: Effects of the textual adapter.

	MuST-C En-De	#Params
MSP-ST-S	27.6	138M
SATE	27.3	-

Table 5: Comparison of two methods without using unlabeled data.

named MSP-ST-S. Note both the two methods do not use the knowledge distillation and SpecAugment(Park et al. 2019). It shows that our method is effective without unlabeled data, and achieves comparable performance with our baseline which trains by massive unlabeled data. This shows previous work fails to use the unlabeled data effectively.

Related Work

One aspect of ST where there has already been substantial success is the cascaded model of ASR and MT (Ney 1999; Matusov, Kanthak, and Ney 2005; Mathias and Byrne 2006). An obvious next step is towards end-to-end ST but initial work attempting to develop fully end-to-end systems on limited labeled data has met with much less success in competing the cascaded counterpart (Bérard et al. 2016). This motivates an active line of research on introducing unlabeled data into ST. A straightforward method is to train ST models by additional ASR and/or MT supervision signals, as in multi-task learning (Anastasopoulos and Chiang 2018; Le et al. 2020; Vydana et al. 2021; Tang et al. 2021b; Han et al. 2021). Similar ideas can be found in other related work, including pseudo data generation (Pino et al. 2019, 2020), meta-learning (Indurthi et al. 2020), knowledge distillation (Liu et al. 2019; Jia et al. 2019) and curriculum learning (Wang et al. 2020c).

For stronger results, a number of recent studies focus on pre-training components of ST systems and fine-tuning them on labeled ST data (Weiss et al. 2017; Bérard et al. 2018; Zheng et al. 2021; Li et al. 2021). Although these systems are of different model designs, researchers are aware that simply incorporating pre-trained ASR and MT models into ST does not work (Wang et al. 2020b; Xu et al. 2021), because there is a great length difference between acoustic sequence and word sequence, and the two models have different scopes of encoding, i.e., the ASR model is locally attentive, while the MT model, which represents sentence semantics, is more globally attentive.

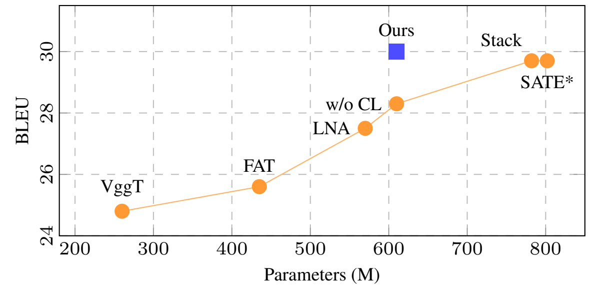


Figure 8: Efficiency of parameters. The *Stack* means stacking the acoustic model and the SIDAE. “SATE*” represents that we reproduce the SATE method. Note the acoustic model contains the alignment adapter.

Several research groups address this by using an additional encoding network to adapt acoustic encoding to text-like encoding (Dong et al. 2021b; Tang et al. 2021a; Li et al. 2021; Xu et al. 2021). Zhang et al. (2022b) propose the hidden-unit pre-training and Ye, Wang, and Li (2022) use the contrastive learning to mitigate the inconsistent representation between the speech and text. Here we explicitly design a trainable text encoder to link ASR and MT pre-training. Perhaps the most related work to what is doing here is Li et al. (2021)’s work. Their system benefits from encoder-decoder pre-training by a text-based BART-like method, but the text encoder is discarded when they train the ST encoder. In this work we find that the involvement of the text encoder in the entire pre-training pipeline is critical to achieve the state-of-the-art performance. We thus share the text encoder in both ASR-based and MT-based pre-training.

Also, it is well-known that silent moments often appear in the acoustic model output but not in MT data. This is in general addressed by either down-sampling the output sequence of the acoustic model (Dong et al. 2021a; Liu et al. 2020b) or converting the source text to the imitation of the acoustic output by CTC paths (Wang et al. 2020b). Here we instead develop a simple denoising method to enhance the ability of the text encoder in dealing with normal and noisy sentences.

The adapter has been introduced to bridge the modality gap (Xu et al. 2021) or length gap between acoustic and textual model (Li et al. 2021). We further consider the robust modeling and the inconsistency of local and global preferences between the two models. And our adapters avoid the use of the whole textual encoder.

Conclusions

We explore methods to pre-train all the components of an ST model on labeled and unlabeled speech and text data. To improve the ST encoder, we develop an alignment adapter and textual adapter. Then, we use a text-based pre-trained encoder to bridge the acoustic encoding and text encoding. In addition, we use contrastive training and denoising training to resist the influence of silent moments in speech. Experiments show our system achieves SOTA results on the MuST-C En-De, En-Fr and LibriSpeech En-Fr tasks.

Acknowledgments

This work was supported in part by the National Science Foundation of China (Nos. 62276056, 61732005 and 61876035), the China HTRD Center Project (No. 2020AAA0107904), Yunnan Provincial Major Science and Technology Special Plan Projects (Nos. 202002AD080001 and 202103AA080015), National Frontiers Science Center for Industrial Intelligence and Systems Optimization (Northeastern University, China. No. B16009) and the Fundamental Research Funds for the Central Universities. The authors would like to thank anonymous reviewers for their comments.

References

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

Baevski, A.; Zhou, Y.; Mohamed, A.; and Auli, M. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Proc. of NeurIPS*.

Bapna, A.; and Firat, O. 2019. Simple, Scalable Adaptation for Neural Machine Translation. In *Proc. of EMNLP*.

Bérard, A.; Pietquin, O.; Servan, C.; and Besacier, L. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *arXiv preprint arXiv:1612.01744*.

Bérard, A.; Besacier, L.; Kocabiyikoglu, A. C.; and Pietquin, O. 2018. End-to-End Automatic Speech Translation of Audiobooks. In *Proc. of ICASSP*.

Chen, S.; Wu, Y.; Chen, Z.; Wu, J.; Li, J.; Yoshioka, T.; Wang, C.; Liu, S.; and Zhou, M. 2021. Continuous Speech Separation with Conformer. In *Proc. of ICASSP*.

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

Dong, Q.; Wang, M.; Zhou, H.; Xu, S.; Xu, B.; and Li, L. 2021a. Consecutive decoding for speech-to-text translation. In *Proc. of 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 *Proc. of AAAI*.

Gállego, G. I.; Tsiamas, I.; Escolano, C.; Fonollosa, J. A.; and Costa-jussà, M. R. 2021. End-to-End Speech Translation with Pre-trained Models and Adapters: UPC at IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proc. of ICML*.

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

Hsu, W.-N.; Bolte, B.; Tsai, Y.-H. H.; Lakhotia, K.; Salakhutdinov, R.; and Mohamed, A. 2021. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

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. In *Proc. of ICASSP*.

Indurthi, S.; Zaidi, M. A.; Kumar Lakumarapu, N.; Lee, B.; Han, H.; Ahn, S.; Kim, S.; Kim, C.; and Hwang, I. 2021. Task Aware Multi-Task Learning for Speech to Text Tasks. In *Proc. of ICASSP*.

Jia, Y.; Johnson, M.; Macherey, W.; Weiss, R. J.; Cao, Y.; Chiu, C.-C.; Ari, N.; Laurenzo, S.; and Wu, Y. 2019. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *Proc. of ICASSP*.

Kahn, J.; Riviere, M.; Zheng, W.; Kharitonov, E.; Xu, Q.; Mazaré, P.-E.; Karadayi, J.; Liptchinsky, V.; Collobert, R.; Fuegen, C.; et al. 2020. Libri-light: A benchmark for asr with limited or no supervision. In *Proc. of ICASSP*.

Kocabiyikoglu, A. C.; Besacier, L.; and Kraif, O. 2018. Augmenting Librispeech with French Translations: A Multimodal Corpus for Direct Speech Translation Evaluation. In *Proc. of LREC*.

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 *Proc. of COLING*.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proc. of ACL*.

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 *Proc. of ACL*.

Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; and Zettlemoyer, L. 2020a. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*.

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

Liu, Y.; Zhu, J.; Zhang, J.; and Zong, C. 2020b. Bridging the modality gap for speech-to-text translation. *arXiv preprint arXiv:2010.14920*.

Mathias, L.; and Byrne, W. 2006. Statistical phrase-based speech translation. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*.

Matusov, E.; Kanthak, S.; and Ney, H. 2005. On the integration of speech recognition and statistical machine translation. In *Ninth European Conference on Speech Communication and Technology*.

Ney, H. 1999. Speech translation: coupling of recognition and translation. In *Proc. of ICASSP*.

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 *Proc. of AACL*.

Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: An ASR corpus based on public domain audio books. In *Proc. of ICASSP*.

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

Pino, J.; Puzon, L.; Gu, J.; Ma, X.; McCarthy, A. D.; and Gopinath, D. 2019. Harnessing indirect training data for end-to-end automatic speech translation: Tricks of the trade. *arXiv preprint arXiv:1909.06515*.

Pino, J.; Xu, Q.; Ma, X.; Dousti, M. J.; and Tang, Y. 2020. Self-Training for End-to-End Speech Translation. In *Proc. Interspeech 2020*, 1476–1480.

Post, M. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Tang, Y.; Gong, H.; Dong, N.; Wang, C.; Hsu, W.-N.; Gu, J.; Baevski, A.; Li, X.; Mohamed, A.; Auli, M.; et al. 2022. Unified Speech-Text Pre-training for Speech Translation and Recognition. *arXiv preprint arXiv:2204.05409*.

Tang, Y.; Pino, J.; Li, X.; Wang, C.; and Genzel, D. 2021a. Improving Speech Translation by Understanding and Learning from the Auxiliary Text Translation Task. In *Proc. of ACL*.

Tang, Y.; Pino, J.; Wang, C.; Ma, X.; and Genzel, D. 2021b. A General Multi-Task Learning Framework to Leverage Text Data for Speech to Text Tasks. In *Proc. of ICASSP*.

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 *Proc. of NeurIPS*.

Vydana, H. K.; Karafiát, M.; Zmolikova, K.; Burget, L.; and Černocký, H. 2021. Jointly Trained Transformers Models for Spoken Language Translation. In *Proc. of ICASSP*.

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

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

Watanabe, S.; Hori, T.; Kim, S.; Hershey, J. R.; and Hayashi, T. 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*.

Weiss, R. J.; Chorowski, J.; Jaitly, N.; Wu, Y.; and Chen, Z. 2017. Sequence-to-sequence models can directly translate foreign speech. *arXiv preprint arXiv:1703.08581*.

Xu, C.; Hu, B.; Li, Y.; Zhang, Y.; Huang, S.; Ju, Q.; Xiao, T.; and Zhu, J. 2021. Stacked Acoustic-and-Textual Encoding: Integrating the Pre-trained Models into Speech Translation Encoders. In *Proc. of ACL*.

Ye, R.; Wang, M.; and Li, L. 2021. End-to-end Speech Translation via Cross-modal Progressive Training. *arXiv preprint arXiv:2104.10380*.

Ye, R.; Wang, M.; and Li, L. 2022. Cross-modal Contrastive Learning for Speech Translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5099–5113. Seattle, United States: Association for Computational Linguistics.

Zhang, H.; Zhang, Z.; Odena, A.; and Lee, H. 2020. Consistency Regularization for Generative Adversarial Networks. In *Proc. of ICLR*.

Zhang, Y.; Huang, C.; Xu, C.; Liu, X.; Li, B.; Ma, A.; Xiao, T.; and Zhu, J. 2022a. The NiuTrans's Submission to the IWSLT22 English-to-Chinese Offline Speech Translation Task. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*.

Zhang, Z.-H.; Zhou, L.; Ao, J.; Liu, S.; Dai, L.; Li, J.; and Wei, F. 2022b. SpeechUT: Bridging Speech and Text with Hidden-Unit for Encoder-Decoder Based Speech-Text Pre-training. *ArXiv*, abs/2210.03730.

Zheng, R.; Chen, J.; Ma, M.; and Huang, L. 2021. Fused Acoustic and Text Encoding for Multimodal Bilingual Pre-training and Speech Translation. In *Proc. of ICML*.