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

Anonymous ACL submission

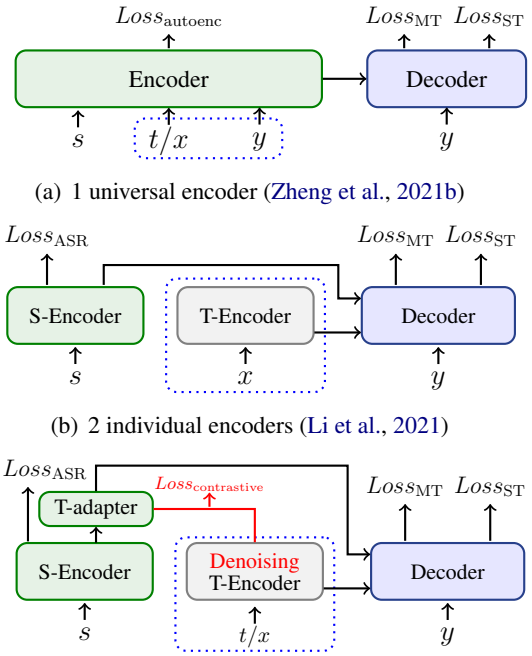
Abstract

We present a method for introducing a text encoder into pre-training 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 for a robust text encoder that can deal with both normal and noisy text data. Our system sets new state-of-the-art on the MuST-C En-De, En-Fr, and LibriSpeech En-Fr tasks.

1 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., 2021a; 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 et al., 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., 2021b). More recently, stronger results have been reported in the work that explicitly designs a text encoder to ease the training on the MT data (Li et al., 2021; Gállego et al., 2021).



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

(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.

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 on all ASR, MT and ST data.

Another improvement is that we denoise the text encoder so that it is robust to the noisy transcription-like input. In this 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-art on these tasks. The performance is even comparable with that of the unrestricted ST system on the LibriSpeech task.

2 Related Work

One aspect of ST where there has already been substantial success is the cascaded model of ASR and MT (Ney, 1999; Schultz et al., 2004; Matusov et al., 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., 2021a; 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.

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). 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). 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 Connectionist Temporal Classification (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.

3 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-

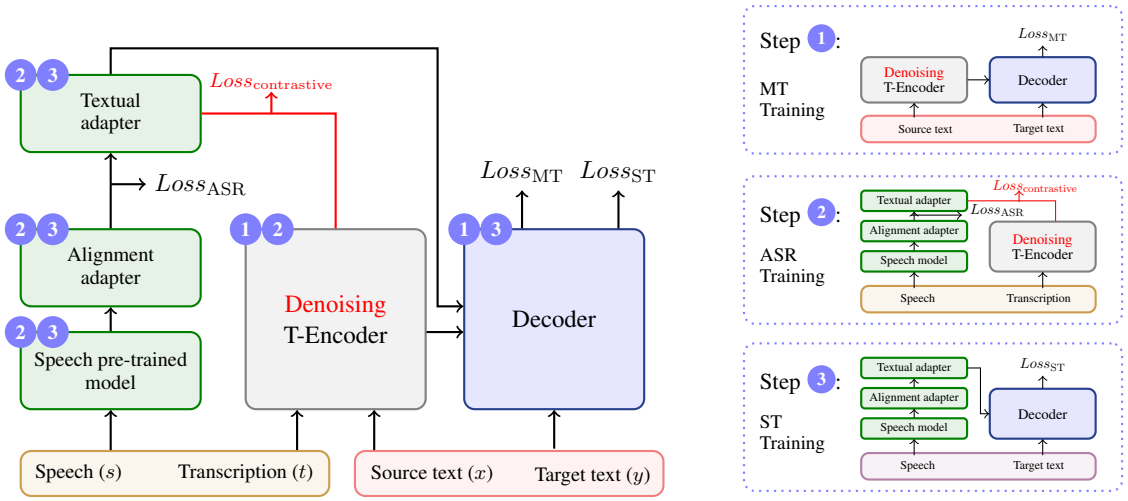


Figure 2: The end-to-end speech translation architecture with a text encoder. Circled numbers indicate training steps.

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.

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 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 these notations 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 textal information. Two intuitions form the basis of this work:

- 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.

3.1 ASR Training with the Text Encoder

An ST encoder in general shares a similar model structure with ASR encoders. An advantage of the ASR-based design for ST encoders is that it is

[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.

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.

3.1.1 Adapters for ST Encoding

For ST encoding, 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).

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 convolution 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 produce a more text encoder-like output, which is better suited for the input of the decoder.

3.1.2 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. The loss function of the contrastive training is defined to be:

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

where $\pi(\cdot,\cdot)$ is a function that computes the similarity of the input vectors. Here we choose the cosine function for $\pi(\cdot,\cdot)$. 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 (x_i, t'_i), we compute a loss $\mathcal{L}'_{\text{CL}}$ as in Eq. 1. The final loss function of ASR training is defined as:

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

where $\mathcal{L}_{\text{CTC}}$ is the CTC loss 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.

3.2 Denoising the Text Encoder

There are two jobs for the text encoder:

- Encode real source-language sentences in MT training
- Encode 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

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
VggT (Pino et al., 2020)	✓	-	✓	-	24.8	34.5	-
FAT-ST (Big) (Zheng et al., 2021b)	✓	✓	✓	✓	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 et al., 2021)	✓	-	-	✓	27.8	38.0	-
SATE (Xu et al., 2021)	-	-	✓	✓	28.1	-	20.8
TCN (Indurthi et al., 2021)	-	-	✓	✓	28.9	-	-
Baseline	✓	✓	✓	✓	27.5	38.6	20.8
MSP-ST	✓	✓	✓	✓	**30.0**	**40.6**	**21.4**

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.

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. Ho 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 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 is described as:

$$\mathcal{L}_{\text{MT}} = -\sum_{(x,y)} \log \text{P}(y \mid x) + \log \text{P}(y \mid g(x)) \quad (3)$$

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

4 Experiments

4.1 Experiment Data

We run our experiments on English to German (En→De) and English to French (En→Fr) translation tasks.

Unlabeled Data. For speech data, we use the LibriVox (Baevski et al., 2020) to pre-train the acoustic model. It consists of about 60k hours of unlabelled speech. For text data, we followed Liu et al. (2020a)'s work which covers 25 languages.

ASR and MT Data. 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 MT tasks, we use Opensubtitle En-De and WMT14 En-Fr datasets respectively. We filter the parallel data by a max length ratio 1.5 and a max length of 200. The final data size is 18M for 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 sentence model and the vocabulary are the same as in (Liu et al., 2020a) and we remove words which do not appear in all the corpora. The vocabulary size is set to 32K for the MuST-C tasks and 25K for the LibriSpeech En-Fr task.

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

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

Table 3: Ablation study on the MuST-ST En-De task.

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.

ST Data. The MuST-C corpus is a multilingual speech translation corpus extracted from TED talks. The size of speech translation data is 400 hours (230K utterances) for the En-De task and 484 hours (270K utterances) for the En-Fr task. For the LibriSpeech En-Fr task, the size of the training set is 100 hours (44K utterances). We remove the utterances of more than 3,000 frames in all the experiments.

4.2 Model settings

We implement our systems by the Fairseq toolkit(Ott et al., 2019; Wang et al., 2020a). 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 comparison, we re-implement the LNA method (Li et al., 2021). For a stronger baseline, we follow Gállego et al. (2021)'s to 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 Transformer without pre-training, the input speech is represented as 80D log mel-filterbank coefficients that are computed every 10ms with a 25ms window.

For pre-training of SIDAE, we set the coefficient r to 0.3. We stop training until the perplexity converges on the valid set. For the alignment adapter, the size of the convolution layer n is set to 3, i.e., we use three 1D convolution layers with a stride of 2. It results in 8 times length compression. For each Conformer layer, there are 1,024 hidden states, 16 attention heads and 4,096 FFN hidden states. For the textual adapter, the configurations of the Conformer layer are the same as the alignment adapter. 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 $\beta_2 = 0.98$. Also, 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. We pre-train our model on the ASR and MT tasks on 8 Nvidia Tesla-V100 GPUs. We fine-tune on the ST task using 4 GPUS with a max token number of 10,000. For unrestricted MT, we first use the MT data and then the ST data to fine-tune the DAE and training settings following 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 to evaluate the performance (Post, 2018). Following previous work, we report case-sensitive SacreBLEU for the MuST-C tasks and case-insensitive SacreBLEU for the LibriSpeech En-Fr task.

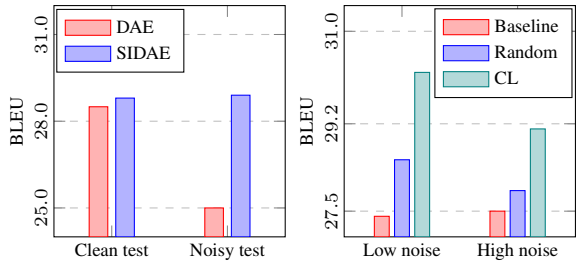
4.3 Results

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 two En-Fr tasks. While on the En-De task, the baseline fails to outperform the methods without using unlabeled data. 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 1.1 BLEU improvement. This also verifies the potential of unlabeled data in ST.

5 Analysis

5.1 Ablation Study

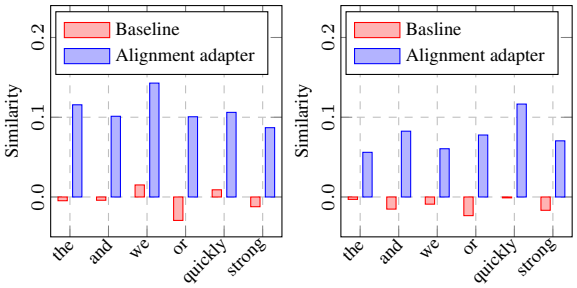
We replace the adapters in the baseline system with our alignment adapter. Table 3 shows that the align-



(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.



(a) Cross modal alignment

(b) Cross lingual alignment

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

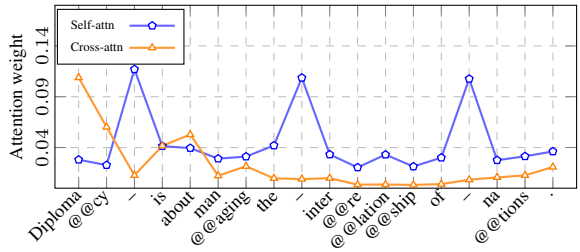


Figure 4: Blank self-attention and other word cross attention. "_" represents the blank token.

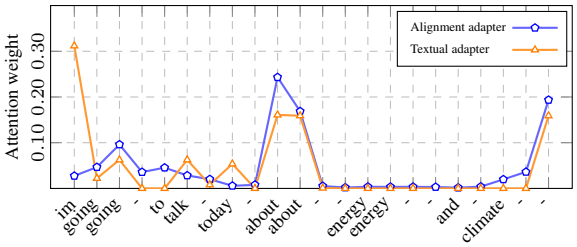


Figure 6: The self-attention weights of alignment adapter and textual adapter.

ment 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. The results show that the textual adapter is the important for satisfactory performance. Also, this results confirms that the semantic conversion and denoising methods are important for ST. Also, we introduce $Loss'_{\text{CL}}$ (denoted as KDCL) into training. 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 side and use the similar speech data. We finally use a Silence Insensitive DAE to mitigate the impact of blank noise for textual adapter. As expected, it helps. Our final MSP-ST method achieves new SOTA results on the MuST-C En-De, En-Fr and LibriSpeech En-Fr tasks.

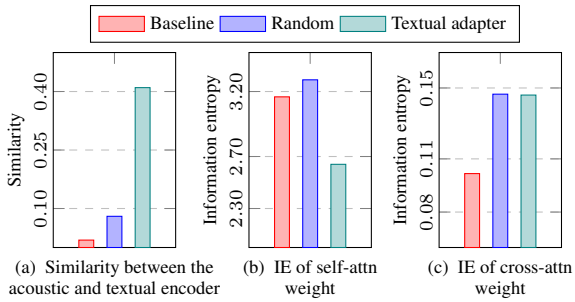
5.2 Effect of Denoising

Figure 3 (a) compares the performance of DAE and SIDAE. 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 perturbation training. When the test text contains many blank labels, the vanilla DAE model is degraded while the SIDAE is robust to the noise. To explore the denosing influence on ST, we split the test set into 2 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 "Random" means that the textual adapter is initialized in a random manner. Its improvement is modest on the high noise test set, while our textual adapter achieves a bigger improvement on the BLEU score.

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.

5.3 Effects of the Alignment Adapter

Here we show the effectiveness of the alignment adapter. We calculate the cosine similarity of word representation between the acoustic model and textual model. The baseline model does not consider the alignment of the acoustic model and the text



(a) Similarity between the acoustic and textual encoder (b) IE of self-attn weight (c) IE of cross-attn weight

Figure 7: Effects of the textual adapter.

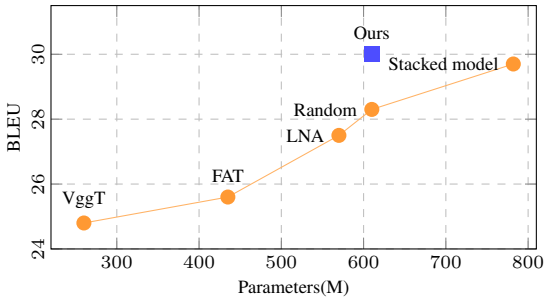


Figure 8: Efficient of parameters. The stacked method means stack the acoustic model and DAE.

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 boost 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.

5.4 Effects of the 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 during the process of information extraction. 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. Figure 7 (b) and (c) show the average information entropy (IE) of attention weights. Note the IE also consists of the noise information. The IE of textual adapter is much lower due to the inattention of noise. The random adapter learns more semantic information but fails to drop the noise so its IE is the highest. Figure 7 (c) shows the usage of adapter can boost the decoder to extract more information. It also proves our gains mainly can from improvement of encoder since the IEs of the two adapters are similar.

5.5 Sample Efficiency

We study how different systems behave under different sized speech translation data. To do this, we scale the training data by about 6.3 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 large when more data is used.

5.6 Parameter Efficiency

Using the pre-training model 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 figure means a higher efficiency. We see that our method is efficient: it achieves the best BLEU score with a slight increase of the parameters. From an aspect of performance, the model which directly stacks the pre-trained acoustic and the whole SIDAE model also achieves comparable performance with our MSP-ST. But our model is more parameter efficient.

6 Conclusions

We explore methods to pre-train all the components of an ST model by 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 model and text model. In addition, we use contrastive training and denoising training to mitigate the influence of silent moments in speech. Our system achieves SOTA results on the MuST-C En-De, En-Fr and LibriSpeech En-Fr tasks.

References

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 82–91, New Orleans, Louisiana. Association for Computational Linguistics.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *arXiv preprint arXiv:1612.01744*.

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6224–6228.

Sanyuan Chen, Yu Wu, Zhuo Chen, Jian Wu, Jinyu Li, Takuya Yoshioka, Chengyi Wang, Shujie Liu, and Ming Zhou. 2021. Continuous speech separation with conformer. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5749–5753.

Qianqian Dong, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. 2021a. Consecutive decoding for speech-to-text translation. In *The Thirty-fifth AAAI Conference on Artificial Intelligence, AAAI*.

Qianqian Dong, Rong Ye, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. 2021b. "listen, understand and translate": Triple supervision decouples end-to-end speech-to-text translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12749–12759.

Gerard I Gállego, Ioannis Tsiamas, Carlos Escolano, José AR Fonollosa, and Marta R Costa-jussà. 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)*, pages 110–119.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Chi Han, Mingxuan Wang, Heng Ji, and Lei Li. 2021. Learning shared semantic space for speech-to-text translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2214–2225, Online. Association for Computational Linguistics.

Sathish Indurthi, Houjeung Han, Nikhil Kumar Lakumarapu, Beomseok Lee, Insoo Chung, Sangha Kim, and Chanwoo Kim. 2020. End-end speech-to-text translation with modality agnostic meta-learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7904–7908. IEEE.

Sathish Indurthi, Mohd Abbas Zaidi, Nikhil Kumar Lakumarapu, Beomseok Lee, Hyojung Han, Seokchan Ahn, Sangha Kim, Chanwoo Kim, and Inchul Hwang. 2021. Task aware multi-task learning for speech to text tasks. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7723–7727.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7180–7184. IEEE.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2020. Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3520–3533, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Multilingual speech translation from efficient finetuning of pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 827–838, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020a. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-End Speech Translation with Knowledge Distillation. In *Proc. Interspeech 2019*, pages 1128–1132.

Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020b. Bridging the modality gap for speech-to-text translation. *arXiv preprint arXiv:2010.14920*.

Lambert Mathias and William Byrne. 2006. Statistical phrase-based speech translation. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE.

Evgeny Matusov, Stephan Kanthak, and Hermann Ney. 2005. On the integration of speech recognition and statistical machine translation. In *Ninth European Conference on Speech Communication and Technology*.

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

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Juan Pino, Liezl Puzon, Jiatao Gu, Xutai Ma, Arya D McCarthy, and Deepak Gopinath. 2019. Harnessing indirect training data for end-to-end automatic speech translation: Tricks of the trade. *arXiv preprint arXiv:1909.06515*.

Juan Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. 2020. Self-Training for End-to-End Speech Translation. In *Proc. Interspeech 2020*, pages 1476–1480.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Tanja Schultz, Szu-Chen Jou, Stephan Vogel, and Shirin Saleem. 2004. Using word latice information for a tighter coupling in speech translation systems. In *Proc. Interspeech 2004*, pages 41–44.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021a. Improving speech translation by understanding and learning from the auxiliary text translation task. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4252–4261, Online. Association for Computational Linguistics.

Yun Tang, Juan Pino, Changhan Wang, Xutai Ma, and Dmitriy Genzel. 2021b. A general multi-task learning framework to leverage text data for speech to text tasks. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6209–6213.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Hari Krishna Vydana, Martin Karafiát, Katerina Zmolikova, Lukáš Burget, and Honza Černocký. 2021. Jointly trained transformers models for spoken language translation. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7513–7517.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020a. Fairseq S2T: Fast speech-to-text modeling with fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 33–39, Suzhou, China. Association for Computational Linguistics.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. 2020b. Bridging the gap between pretraining and fine-tuning for end-to-end speech translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9161–9168.

Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. 2020c. Curriculum pre-training for end-to-end speech translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3728–3738, Online. Association for Computational Linguistics.

Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. *arXiv preprint arXiv:1703.08581*.

Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2021. Stacked acoustic-and-textual encoding: Integrating the pre-trained models into speech translation encoders. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*

and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 2619–2630, Online. Association for Computational Linguistics.

Rong Ye, Mingxuan Wang, and Lei Li. 2021. End-to-end speech translation via cross-modal progressive training. *arXiv preprint arXiv:2104.10380*.

Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. 2020. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations*.

Renjie Zheng, Junkun Chen, Mingbo Ma, and Liang Huang. 2021a. Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation. *arXiv preprint arXiv:2102.05766*.

Renjie Zheng, Junkun Chen, Mingbo Ma, and Liang Huang. 2021b. Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12736–12746. PMLR.