Non-Parametric Domain Adaptation for End-to-End Speech Translation

Yichao Du[‡♯]*, **Weizhi Wang**[§*], **Zhirui Zhang**[♮†], **Boxing Chen**[♭], **Tong Xu**[‡♯]
Jun Xie[♭] **and Enhong Chen**[‡♯†]
[‡]University of Science and Technology of China [♯]State Key Laboratory of Cognitive Intelligence
[§]University of California, Santa Barbara [♮]Tencent AI Lab
[♭]Machine Intelligence Technology Lab, Alibaba DAMO Academy
[‡♯]duyichao@mail.ustc.edu.cn [‡♯]{tongxu, cheneh}@ustc.edu.cn [§]weizhiwang@ucsb.edu
[♮]zrustc11@gmail.com [♭]{boxing.cbx, qingjing.xj}@alibaba-inc.com

Abstract

The end-to-end speech translation (E2E-ST) has received increasing attention due to the potential of its less error propagation, lower latency and fewer parameters. However, the effectiveness of neural-based approaches to this task is severely limited by the available training corpus, especially for domain adaptation where in-domain triplet data is scarce or nonexistent. In this paper, we propose a novel non-parametric method that leverages in-domain text translation corpus to achieve domain adaptation for E2E-ST systems. To this end, we first incorporate an additional encoder into the pre-trained E2E-ST model to realize text translation modeling, based on which the decoder's output representations for text and speech translation tasks are unified by reducing the correspondent representation mismatch in available triplet training data. During domain adaptation, a k-nearest-neighbor (kNN) classifier is introduced to produce the final translation distribution using the external datastore built by the domain-specific text translation corpus, while the universal output representation is adopted to perform a similarity search. Experiments on the Europarl-ST benchmark demonstrate that when in-domain text translation data is involved only, our proposed approach significantly improves baseline by 12.82 BLEU on average in all translation directions, even outperforming the strong in-domain fine-tuning strategy.

1 Introduction

Speech translation (ST), the task of automatically translating speech signals in a given language into text in another language, becomes a widely studied topic with the increasing demand for international communications. Traditional ST systems cascade automatic speech recognition (ASR) and machine translation (MT) (Ney, 1999; Sperber et al., 2017; Zhang et al., 2019; Iranzo-Sánchez et al., 2020a; Machácek et al., 2021). So far, various large-scale speech-translation datasets have been proposed, e.g., Libri-Trans (Kocabiyikoglu et al., 2018), MuST-C (Gangi et al., 2019) and CoVoST (Wang et al., 2020a). With these large-scale annotations, building an end-to-end speech translation (E2E-ST) system (Vila et al., 2018; Liu et al., 2019; Li et al., 2021; Han et al., 2021; Dong et al., 2021) has become popular, since it has lower latency and less error propagation compared with previous ST methods. Recent studies have also shown that there is no significant difference between end-to-end models and cascaded systems in translation performance (Bentivogli et al., 2021).

In many practical application scenarios, such as political negotiations, business meetings, etc., there is no available in-domain speech-translation dataset to conduct the end-to-end training, which essentially limits the promotion of E2E-ST systems. The most common practice is that the E2E-ST model learns knowledge well enough in the general domain, and then it is directly used to translate speech input in the target domain. Unfortunately, due to the domain shift issue (Gretton et al., 2006; Ramponi and Plank, 2020), the generalization capabilities of current end-to-end models are somehow weak across different scenarios. Instead of speech-translation annotations, parallel text corpus in the target domain is usually abundant and easy to collect. Thus, it is essential to explore and extend the capability of E2E-ST systems in this scenario, in which a large amount of in-domain bilingual text is utilized.

In this paper, we focus on this domain adaptation setting and attempt to replace the domain-specific parameter updating in neural-based E2E-ST models with a non-parametric search to make it adaptable and achieve domain adaptation without any speech-translation annotations. Actually, the non-parametric approach kNN-MT, recently proposed by Khandelwal et al. (2020), is a promising alter-

*Equal Contribution
†Corresponding author

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 306–320
December 7-11, 2022

native to reach this goal. The kNN-MT equips the pre-trained neural machine translation (NMT) model with a k-nearest-neighbor (kNN) classifier over an external datastore to improve translation accuracy without retraining. However, it requires the in-domain speech-translation corpus to construct an effective datastore when we apply this method in the speech translation setting. To tackle this problem, we propose a novel **N**on-**P**arametric **D**omain **A**daptation framework based on kNN-MT for E2E-ST, named as NPDA-kNN-ST. Its key core is to directly leverage the in-domain text translation corpus to generate the corresponding datastore and encourage it to play a similar role as the real in-domain speech-translation data, through the carefully designed architecture and loss function.

Specifically, we first incorporate an additional trainable encoder for text modeling into the pre-trained E2E-ST model. Based on that, we make the decoder's output representations for text and speech translation tasks close, through reducing the representation inconsistency of these two tasks in triplet training data and keeping the parameters of the original pre-trained E2E-ST model fixed. In this way, the additional encoder module learns the semantic mapping in feature space from the source language text to the speech signal, which enables the construction of an effective in-domain datastore when text translation data is involved only. Then we introduce a kNN classifier to produce the final translation distribution based on the domain-specific datastore built by the correspondent text translation data. Meanwhile, the universal output representation is adopted to perform a similarity search and guides the translation process.

We evaluate our approach on the Europarl-ST benchmark and demonstrate that our method significantly outperforms the strong in-domain fine-tuning strategy by 3.85 BLEU scores on average in all translation directions when only using large-scale in-domain text translation data. Additional experiments on Europarl-ST and MuST-C datasets verify that the in-domain text translation datastore generated by our method could play a similar role with the real in-domain speech-translation data, thanks to the universal output representation.

2 Background

2.1 End-to-End Speech Translation

E2E-ST systems receive speech signals in a source language and directly generate the text in a target language without an intermediate transcription process. Concretely, the E2E-ST corpus consists of a set of triplet data $\mathcal{D}_{ST} = \left\{(\mathbf{x}^{(n)}, \mathbf{z}^{(n)}, \mathbf{y}^{(n)})\right\}_{n=1}^{N}$, where $\mathbf{x}^{(n)} = (x_1^{(n)}, x_2^{(n)}, ..., x_{|\mathbf{x}^{(n)}|}^{(n)})$ is the input sequence of the speech wave (in most cases, acoustic features are used), $\mathbf{z}^{(n)} = (z_1^{(n)}, z_2^{(n)}, ..., z_{|\mathbf{z}^{(n)}|}^{(n)})$ represents the transcription sequence from the source language and $\mathbf{y^{(n)}} = (y_1^{(n)}, y_2^{(n)}, ..., y_{|\mathbf{y}^{(n)}|}^{(n)})$ denotes the translation sequence of target language. The goal of E2E-ST model is to 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) of the training data $\mathcal{D}_{ST}$:

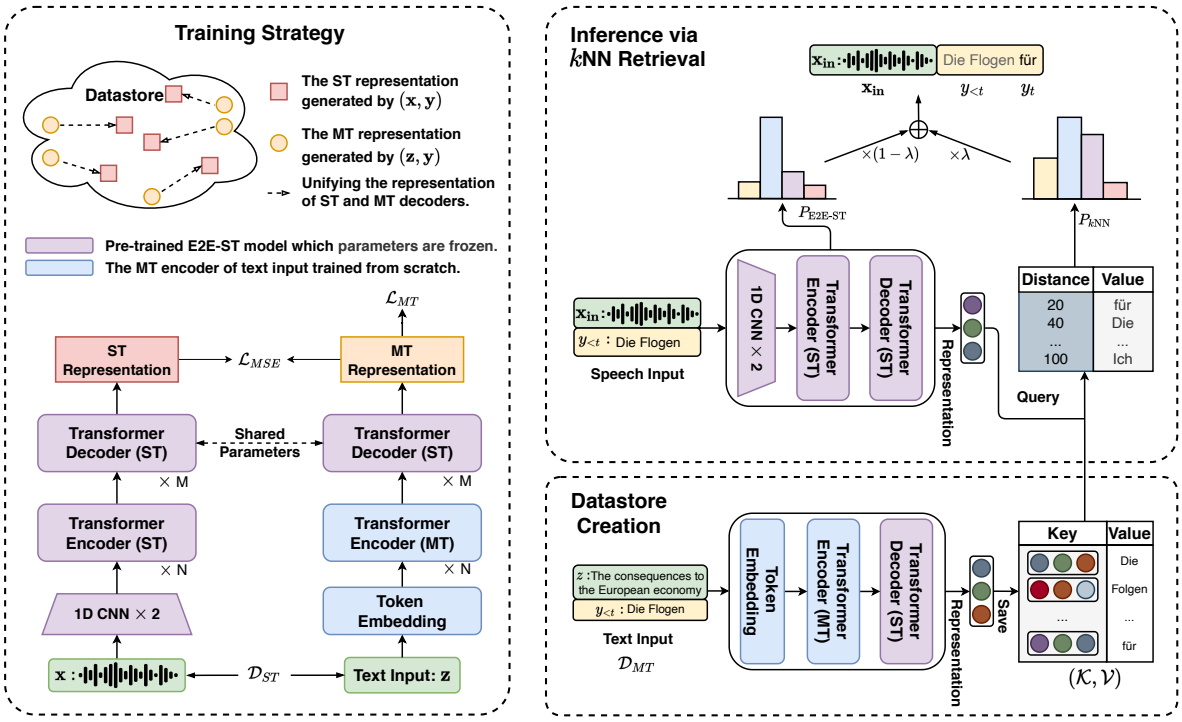
$$\mathcal{L}_{ST}(\theta) = \frac{1}{N}\sum_{n=1}^{N} \log P\left(\mathbf{y^{(n)}} \mid \mathbf{x^{(n)}}; \theta\right), \quad (1)$$

where a single encoder-decoder structure is adopted to fit the conditional distribution $P(\mathbf{y^{(n)}}|\mathbf{x^{(n)}})$ and θ is the model parameter. To develop high-quality E2E-ST systems, ASR and MT tasks ($(\mathbf{x}^{(n)}, \mathbf{z}^{(n)})$ and $(\mathbf{z}^{(n)}, \mathbf{y}^{(n)})$) are typically used to pre-train the encoder and decoder, respectively. (Bansal et al., 2019; Wang et al., 2020c). However, in practice, it is not realistic to obtain a large amount of high-quality speech-translation data in every domain that we are interested in, while in-domain text translation corpus is usually cheaper and easier to collect. Thus, it is essential to investigate the capability of E2E-ST model that uses large-scale in-domain text translation corpus to achieve domain adaptation, making E2E-ST systems more practical.

2.2 Nearest Neighbor Machine Translation

Recently, kNN-MT (Khandelwal et al., 2020) has shown the promising capability of directly augmenting the pre-trained NMT model with domain-specific token-level kNN retrieval to improve the translation performance without retraining. kNN-MT mainly involves two steps: datastore creation and token inference with cached datastore.

Datastore Creation. The datastore of kNN-MT is the cache of a set of key-value pairs. Given a parallel sentence pair $(\mathbf{z}, \mathbf{y}) \in (\mathcal{Z}, \mathcal{Y})$, the pre-trained NMT model generates the context representation $f_\theta(\mathbf{z}, y_{<t})$ at each timestep t. Then the whole datastore $(\mathcal{K}, \mathcal{V})$ is constructed by taking the output



(a) Unifying text and speech representation. (b) Performing domain adaptation via kNN retrieval.

Figure 1: The overview of our non-parametric domain adaptation framework for E2E-ST (NPDA-kNN-ST).

hidden state $f_\theta(\mathbf{z}, y_{<t})$ as key and y_t as value:

$$(\mathcal{K}, \mathcal{V}) = \bigcup_{(\mathbf{z},\mathbf{y})\in(\mathcal{Z},\mathcal{Y})} \{(f_\theta(\mathbf{z}, y_{<t}), y_t), \forall y_t \in \mathbf{y}\}. \tag{2}$$

Inference via kNN Retrieval. In the inference stage, kNN-MT model predicts the probability distribution of t-th target token y_t with the context representation $f_\theta(\mathbf{z}, y_{<t})$. Specifically, kNN-MT utilizes the context representation to query the cached datastore $(\mathcal{K}, \mathcal{V})$ and retrieves k nearest neighbor key-value pairs w.r.t. Euclidean distance. Then the probability distribution of y_t generated by kNN retrieval is calculated as follow:

$$p_{k\text{NN}}(y_t|\mathbf{z}, y_{<t}) \propto \sum_{(h_i,v_i)\in\mathcal{R}} \mathbb{1}_{y_t=v_i} \exp\left(\frac{-d(h_i, f_\theta(\mathbf{z}, y_{<t}))}{T}\right), \tag{3}$$

where $\mathcal{R} = \{(h_i, v_i), i \in \{1, 2, ..., k\}\}$ is the set of k nearest neighbors, $d(\cdot, \cdot)$ represents the squared Euclidean distance and T is the temperature to control the sharpness of softmax function. The final output distribution is an interpolation between distributions from the NMT model and kNN retrieved neighbors with a tuned parameter $\lambda \in [0, 1]$:

$$\begin{aligned} p(y_t|\mathbf{z}, y_{<t}) = \lambda\, & p_{k\text{NN}}(y_t|\mathbf{z}, y_{<t}) \\ & + (1-\lambda)\, p_{\text{NMT}}(y_t|\mathbf{z}, y_{<t}). \end{aligned} \tag{4}$$

3 Method

When we apply kNN-MT in the speech translation task, it needs the real speech-translation corpus to build an effective datastore for kNN retrieval. However, this requirement could not be satisfied in the domain adaptation scenario mentioned before, as there is no available in-domain speech-translation corpus. In this paper, we focus on this setting and target to replace the domain-specific parameter updating with a non-parametric search to achieve domain adaptation. We design a novel **N**on-**P**arametric **D**omain **A**daptation framework based on kNN-MT for E2E-ST, named as NPDA-kNN-ST. The overview framework of NPDA-kNN-ST is illustrated in Figure 1, which is mainly divided into two parts: a) unifying text and speech representation to enable datastore creation; b) performing domain adaptation through kNN retrieval. Next, we would introduce the model architecture, training objective and inference process in detail.

3.1 Unifying Text and Speech Representation

The NPDA-kNN-ST aims to directly build an in-domain effective datastore with only text translation corpus, making it play a similar role with the real in-domain speech-translation data. It means that whether word tokens or speech signals are treated as input, we should construct the universal output representation for them in a unified model. As shown in Figure 1(a), we introduce an additional transformer encoder and reuse the original transformer decoder of the pre-trained E2E-ST model for source text modeling. In this way, we only increase a few parameters for our approach.

Based on this model structure, we further attempt to make the decoder's output representations for text and speech translation tasks close, by which the text translation data could be leveraged to build an effective in-domain datastore. We achieve this by leveraging out-of-domain triplet data $\mathcal{D}_{ST}$, which is also adopted to build the pre-trained E2E-ST model. More specifically, given a triplet data point in the corpus $(\mathbf{x}, \mathbf{z}, \mathbf{y}) \in \mathcal{D}_{ST}$, the original E2E-ST model takes speech-translation pair $(\mathbf{x}, \mathbf{y})$ as input and generates output representation $f_{(\theta_e,\theta_d)}(\mathbf{x};\ y_{<t})$ for each target token y_t. Meanwhile, with corresponding text translation pair $(\mathbf{z}, \mathbf{y})$, the model with an additional transformer encoder produces another representation for y_t, which can be denoted as $f_{(\theta'_e,\theta_d)}(\mathbf{z};\ y_{<t})$. Next, we take the end-to-end paradigm and directly update the introduced transformer encoder by minimizing the squared Euclidean distance of the two sets of decoder representations and optimizing the MLE loss of text translation pair:

$$\mathcal{L}_{MSE}(\theta'_e) = \frac{1}{N} \sum_{(\mathbf{x},\mathbf{z},\mathbf{y})\in\mathcal{D}_{ST}} (\frac{1}{|\mathbf{y}|} \sum_{t=1}^{|\mathbf{y}|} ||f_{(\theta_e,\theta_d)}(\mathbf{x}; y_{<t}) - f_{(\theta'_e,\theta_d)}(\mathbf{z}; y_{<t})||^2),$$

$$\mathcal{L}_{MT}(\theta'_e) = \frac{1}{N} \sum_{n=1}^{N} \log P\left(\mathbf{y^{(n)}} \mid \mathbf{z^{(n)}}; \theta'_e, \theta_d\right),$$

$$\mathcal{L}(\theta'_e) = \mathcal{L}_{MT}(\theta'_e) - \mathcal{L}_{MSE}(\theta'_e), \quad (5)$$

where θ_e and θ_d are parameters of encoder and decoder in the pre-trained E2E-ST model respectively, θ'_e represents the parameter of the new transformer encoder and token embedding, and we keep θ_e and θ_d fixed during this training process to avoid the E2E-ST performance degradation in the inference stage. The out-of-domain validation set and its correspondent loss are adopted to select the best model in our experiments.

3.2 Domain Adaptation via kNN Retrieval

We consider the domain adaptation scenario of E2E-ST that only domain-specific text translation corpus $\mathcal{D}_{MT} = \left\{(\mathbf{z}^{(m)}, \mathbf{y}^{(m)})\right\}_{m=1}^{M}$ is available. During domain adaptation, the entire inference process is illustrated in Figure 1(b). Once we gain the well-trained model with Equation 5, the new transformer encoder and original transformer decoder of pre-trained E2E-ST model are utilized to forward pass the text translation corpus $\mathcal{D}_{MT}$ to create an in-domain datastore $(\mathcal{K}, \mathcal{V})$. This construction process is the same as the kNN-MT. Due to the universal decoder's representation, this datastore is directly used for the kNN retrieval when translating in-domain speech input $\mathbf{x_{in}}$. The final probability of NPDA-kNN-ST to predict the next token y_t is an interpolation of two distributions with a tuned hyper-parameter λ:

$$\begin{aligned} p(y_t|\mathbf{x_{in}}, y_{<t}) = \lambda\ & p_{k\text{NN}}(y_t|\mathbf{x_{in}}, y_{<t}) \\ & +(1-\lambda)\ p_{\text{E2E-ST}}(y_t|\mathbf{x_{in}}, y_{<t}), \end{aligned} \quad (6)$$

where $p_{\text{E2E-ST}}$ indicates the general domain E2E-ST prediction and p_{kNN} represents the in-domain retrieval based on Equation 3. Actually, this prediction way could also be substituted with other kNN variants (Zheng et al., 2021a; He et al., 2021; Meng et al., 2021) to achieve better model performance or inference speed.

4 Experiments

4.1 Setup

We conduct experiments to evaluate our proposed approach in two aspects: a) domain adaptation on Europarl-ST benchmark with the E2E-ST model pre-trained on MuST-C dataset, and vice versa; b) the performance comparisons on MuST-C benchmark when speech-translation and text-translation data are leveraged to build datastore, respectively.

MuST-C Dataset. MuST-C (Gangi et al., 2019) is a publicly large-scale multilingual ST corpus, which is built from English TED Talks and consists of triplet data: source speech, source transcription, and target translation. It 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). See Appendix A.1 for detailed statistics.

Europarl-ST Dataset. Europarl-ST (Iranzo-Sánchez et al., 2020b) collects from the official transcriptions and translations of European Parliament debate. For domain adaptation, we select seven languages (DE, FR, IT, RO, NL, PT, and ES) that intersect with MuST-C. The training size of Europarl-ST is one-ninth of MuST-C, and our method only leverages the bilingual text in the entire data to achieve domain adaptation. The detailed statistics of the dataset are shown in Appendix A.1.

Europarl-MT Dataset. In order to further verify the performance of our proposed method with large-scale text translation data, we introduce the easily accessible in-domain parallel corpus – Europarl-MT (Koehn, 2005). In our experiments, we randomly select 2M sentence pairs for each translation direction, except for the EN-RO translation direction. We adopt the entire Europarl-MT for EN-RO, which consists of almost 400k bilingual samples.

Baselines. We compare our proposed approach (NPDA-kNN-ST) with several baselines:

- **E2E-ST-Base**: The pre-trained E2E-ST model built by the MuST-C dataset. It is also treated as the pre-trained model for NPDA-kNN-ST.
- **E2E-ST-SP**: The in-domain E2E-ST model on Europarl-ST. Its training process is consistent with E2E-ST-Base.
- **E2E-ST-FT**: The fine-tuned version of E2E-ST-Base using the Europarl-ST corpus.
- **LNA-D**: The multilingual E2E-ST model proposed by Li et al. (2021). It integrates Wave2vec 2.0 (Baevski et al., 2020) and mBART (Chipman et al., 2021), while layernorm and attention layers in the decoder are fine-tuned with CoVoST dataset (Wang et al., 2020a).
- **kNN-MT**: We directly apply kNN-MT (Khandelwal et al., 2020) for E2E-ST-Base and construct the cached datastore with the in-domain speech-translation data.
- **Shallow Fusion**: We utilize Europarl-MT dataset to train in-domain language model (LM). During inference, we re-score hypotheses with the weighted sum of the scores by the E2E-ST-Base and LM models (Gulcehre et al., 2015).
- **E2E-JT-ST-MT**: The joint-training model with the MuST-C and Europarl-MT datasets, which adopts the same model structure as our method and all model parameters are tune-able.

Dataset Pre-processing and Implementation Details. We follow the FAIRSEQ S2T (Wang et al., 2020b) recipes to perform data pre-processing. For the speech data in Europarl-ST and MuST-C, we extract an 80-dimensional log-Mel filter bank as the acoustic feature. For the external text translation data, we delete the bilingual data in Europarl-MT that intersects with the validation/test sets of the Europarl-ST dataset. Refer to Appendix A.2 for dataset pre-processing details. All experiments are implemented based on the FAIRSEQ (Ott et al., 2019) toolkit. For the model structure of all baselines, it consists of two one-dimensional convolutional layers with a downsampling factor of 4, 12 transformer encoder layers and 6 transformer decoder layers. During training, we deploy the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 2e-3 and 10K warm-up updates to optimize model parameters. All models are trained with one Tesla-V100 GPU and we set patience to 5 to select the best checkpoint on the validation set. More implementation details can be found in Appendix A.3 and A.4. In all experiments, we report the case-sensitive BLEU score (Papineni et al., 2002) using sacreBLEU[1]. Our code is open-sourced at https://github.com/duyichao/NPDA-KNN-ST.

4.2 Main Results

Domain Adaptation on Europarl-ST. We first evaluate the domain adaptation performance of NPDA-kNN-ST on Europarl-ST. As illustrated in Table 1, NPDA-kNN-ST obtains the significant improvements over E2E-ST-Base in all language pairs. Once the large-scale Europarl-MT data is involved, NPDA-kNN-ST$^+$ achieves 12.82 BLEU improvements over E2E-ST-Base on average, even significantly outperforming strong in-domain fine-tuning approach E2E-ST-FT. Benefiting from the language model trained on large-scale in-domain data, Shallow Fusion gains similar performance to E2E-ST-SP, but it is still inferior to NPDA-kNN-ST. E2E-JT-ST-MT achieves better performance by jointly training the entire model with large in-domain parallel text and out-of-domain speech data, but still falls short of NPDA-kNN-ST$^+$. These results prove that our proposed method can make full use of in-domain parallel text to achieve domain adaptation when in-domain speech translation data is inaccessible. In addition, NPDA-kNN-ST

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

Model	Used Data				Params. (M)	Target Language							
	MC	EP-ST	EP-MT	Extra.	Tuned/Total	DE	FR	ES	NL	IT	RO	PT	Avg.
E2E-ST-Base	✓	×	×	×	0.0/31.1	15.71	16.45	23.49	16.06	14.25	16.95	18.28	17.31
LNA-D	×	×	×	✓	384.8/793.0	22.50	30.00	32.23	/	21.50	/	28.40	/
E2E-ST-SP	×	✓	×	×	31.1/31.1	16.20	24.52	26.00	19.50	18.35	20.62	21.34	20.93
E2E-ST-FT	✓	✓	×	×	31.1/31.1	21.84	30.97	32.25	23.77	23.36	25.47	26.30	26.28
kNN-MT	✓	✓	×	×	0.0/31.1	18.29	27.69	28.93	20.70	20.45	22.37	23.08	23.07
NPDA-kNN-ST	✓	✓	×	×	17.1/48.1	18.76	27.73	29.01	20.79	20.54	23.54	23.54	23.42
Shallow Fusion	✓	×	✓	×	6.8/37.9	17.72	22.66	26.25	20.12	18.77	20.58	21.59	21.67
E2E-JT-ST-MT	✓	×	✓	×	48.1/48.1	22.10	33.62	31.28	21.35	22.18	23.21	23.62	25.34
NPDA-kNN-ST$^+$	✓	×	✓	×	17.1/48.1	**23.23**$^{\uparrow}$	**35.26**$^{\Uparrow}$	**33.71**$^{\Uparrow}$	**27.71**$^{\Uparrow}$	**33.76**$^{\Uparrow}$	**28.29**$^{\Uparrow}$	**28.96**$^{\Uparrow}$	**30.13**

Table 1: BLEU score [%] of different methods on the Europarl-ST dataset. "Tuned Params." refers to the number of fine-tuned parameters. "NPDA-kNN-ST$^+$" directly uses large-scale Europarl-MT data to build the in-domain datastore, while "NPDA-kNN-ST" leverages the text translation part in the Europarl-ST training data. "MC, EP-ST, EP-MT and Extra." means whether the method uses MuST-C, Europarl-ST, Europarl-MT and external data, respectively. "$^{\Uparrow/\uparrow}$" indicates "NPDA-kNN-ST$^+$" significant difference ($p < 0.01/0.05$) from strong in-domain baseline "E2E-ST-FT", tested by bootstrap re-sampling (Koehn, 2004).

Model	Used Data			Params. (M)	Target Language							
	EP-ST	MC-ST	MC-MT	Tuned/Total	DE	FR	ES	NL	IT	RO	PT	Avg.
E2E-ST-SP	✓	×	×	0.0/31.1	4.14	4.60	5.35	4.50	1.94	3.90	5.09	4.22
Shallow Fusion	✓	×	✓	6.8/37.9	4.72	5.33	6.13	4.14	2.01	4.35	5.79	4.64
E2E-JT-ST-MT	✓	×	✓	48.1/48.1	5.73	7.58	7.62	5.85	**3.83**	5.05	6.45	6.02
kNN-MT	✓	✓	×	0.0/31.1	**5.80**$^{\Uparrow}$	8.02$^{\Uparrow}$	8.21$^{\Uparrow}$	6.02$^{\Uparrow}$	3.41$^{\Uparrow}$	**5.23**$^{\Uparrow}$	7.13$^{\Uparrow}$	6.26
NPDA-kNN-ST	✓	×	✓	17.1/48.1	5.70$^{\Uparrow}$	**8.24**$^{\Uparrow}$	**8.28**$^{\Uparrow}$	**6.19**$^{\Uparrow}$	3.45$^{\Uparrow}$	**5.21**$^{\Uparrow}$	**7.19**$^{\Uparrow}$	**6.32**

Table 2: BLEU score [%] of different domain adaptation methods on the MuST-C dataset. "MC-ST/MC-MT" indicates whether the method uses MuST-C ST/MT data, respectively. "$^{\Uparrow}$" indicates the method significant difference ($p < 0.01$) from baseline "E2E-ST-SP".

obtains comparable translation performance with kNN-MT that leverages the truly in-domain speech-translation data to construct a datastore. It further indicates that our method could generate an effective in-domain datastore with text translation data, which is equivalent to the real speech-translation data. We also compare our proposed method with LNA-D that builds the large multilingual E2E-ST model based on Wave2vec and mBART. In spite of adopting a huge model scale and pre-training techniques, this approach still fails to outperform NPDA-kNN-ST$^+$ due to the domain shift problem. This result shows the necessity of domain adaptation when applying large-scale general E2E-ST models in a certain domain.

Domain Adaptation on MuST-C. We further reverse the domain adaptation direction to verify the performance of our approach, such as domain adaptation to MuST-C using E2E-ST-SP. From Table 2, we can see that NPDA-kNN-ST still significantly outperforms E2E-ST-SP and Shallow Fusion, yielding comparable results to E2E-JT-ST-MT. Actually, Europarl-ST data is too small to build a good generic model, and its domain coverage is too narrow (i.e., only the political domain), resulting in the poor transfer performance of our method and low translation results of all methods. It also brings an interesting research direction that incorporates our method with the large E2E-ST model, such as LNA-D, and we leave it as future work.

E2E-ST Performance on MuST-C. We investigate the effect of unifying text and speech representation with an additional encoder on MuST-C. In this experiment, we compare the translation performance when speech and text translation data are leveraged to construct the datastore respectively, and verify the improvement of combining kNN retrieval with traditional E2E-ST models at the same time. As illustrated in Table 3, we consider both bilingual and multilingual settings, and compare our method with other baselines, including AFS (Zhang et al., 2020), LNA-D and Adapter Tuning (Le et al., 2021). When directly incorporating kNN retrieval into E2E-ST-Base, NPDA-kNN-ST

	Model	Params. (M) Tuned/Total	Target Language DE	FR	ES	NL	IT	RO	PT	RU	Avg.
Bilingual	E2E-ST-Base	31.1/31.1	22.57	32.61	27.08	27.46	22.74	21.80	28.07	15.45	24.72
	AFS	-	22.40	31.60	26.90	24.90	23.00	21.00	26.30	14.70	23.85
	kNN-MT	0.0/31.1	22.97$^{\uparrow}$	33.00$^{\uparrow}$	27.99$^{\Uparrow}$	27.93$^{\uparrow}$	**23.55**$^{\uparrow}$	22.16	28.80$^{\uparrow}$	15.73$^{\uparrow}$	25.27
	NPDA-kNN-ST	17.1/48.1	**23.08**$^{\uparrow}$	**33.24**$^{\Uparrow}$	**28.03**$^{\Uparrow}$	**28.11**$^{\uparrow}$	23.44$^{\uparrow}$	**22.22**$^{\uparrow}$	**28.83**$^{\Uparrow}$	**15.82**$^{\uparrow}$	**25.35**
Multilingual	E2E-ST-Base	76.3/76.3	24.18	34.98	28.28	28.80	24.62	23.22	31.13	15.88	26.39
	LNA-D	76.3/76.3	24.16	34.52	28.30	28.35	24.46	23.29	30.51	15.84	26.18
	Adapter Tuning	76.3/76.3	24.63	34.75	28.73	28.80	24.96	23.70	30.96	15.89	26.61
	kNN-MT	0.0/76.3	25.15$^{\uparrow}$	**35.67**$^{\Uparrow}$	**30.22**$^{\Uparrow}$	**30.36**$^{\Uparrow}$	25.83$^{\Uparrow}$	23.66	**31.67**$^{\uparrow}$	17.16$^{\Uparrow}$	27.47
	NPDA-kNN-ST	23.7/100.0	**25.21**$^{\Uparrow}$	35.56$^{\uparrow}$	30.05$^{\Uparrow}$	30.31$^{\Uparrow}$	**25.91**$^{\Uparrow}$	**23.90**$^{\uparrow}$	31.66$^{\uparrow}$	**17.23**$^{\Uparrow}$	**27.48**

Table 3: BLEU score [%] of different E2E-ST methods on the MuST-C dataset. "AFS" and "Adapter Tuning" represent the methods proposed by Zhang et al. (2020) and Le et al. (2021), respectively. Besides, Le et al. (2021) reproduce the translation performance of "LNA-D" on the MuST-C dataset for fair comparison. "$\Uparrow/\uparrow$" indicates "NPDA-kNN-ST/kNN-ST" significant difference ($p < 0.01/0.05$) from the backbone "E2E-ST-Base".

Metric	Model	DE	FR	ES	NL	IT	RO	PT	Avg.
BLEU Score(↑)	NPDA-kNN-ST	**18.76**	**27.73**	**29.01**	**20.79**	**20.54**	**23.54**	**23.54**	**23.42**
	- w/o MSE Loss	18.44	26.66	28.10	19.93	19.89	22.20	22.45	22.52
	- Optimize Embedding Only	18.50	27.42	28.64	20.44	20.15	22.92	23.09	23.02
Cosine Similarity (↑)	NPDA-kNN-ST	**0.865**	**0.874**	**0.858**	**0.860**	**0.867**	**0.861**	**0.850**	**0.862**
	- w/o MSE Loss	0.827	0.836	0.811	0.817	0.825	0.828	0.809	0.822
	- Optimize Embedding Only	0.844	0.857	0.839	0.844	0.849	0.845	0.832	0.844
Squared Euclidean Distance (↓)	NPDA-kNN-ST	**5.387**	**4.723**	**5.050**	**5.637**	**5.098**	**4.996**	**5.707**	**5.228**
	- w/o MSE Loss	6.260	5.566	6.070	6.650	6.040	5.938	6.690	6.173
	- Optimize Embedding Only	5.610	4.863	5.400	5.950	5.434	5.266	6.043	5.509

Table 4: BLEU score [%], cosine similarity and squared euclidean distance of our approach's variants on the Europarl-ST dataset. "w/o MSE Loss" means that the MSE loss function is removed. "Optimize Embedding Only" indicates that only the token embedding is introduced to the pre-trained E2E-ST model and fine-tuned.

yields 0.63 and 1.09 BLEU improvements on average in bilingual and multilingual settings, respectively. These results indicate the benefit of introducing kNN retrieval, even when the E2E-ST model is trained on the same data. In addition, NPDA-kNN-ST achieves similar performance with kNN-MT in both bilingual and multilingual settings, which proves the effectiveness of our proposed method on unifying text and speech representation again.

4.3 Analysis

Ablation Study. To analyze different modules in our method, we conduct an ablation study on the Europarl-ST dataset, including removing the MSE loss function and introducing only token embedding for unifying text and speech representation. Except for the BLEU score, we measure the cosine similarity and squared euclidean distances between the synthetic representations generated by our method and ideals generated using ground-truth speech-translation data. As shown in Table 4, even without in-domain speech-translation data, NPDA-kNN-ST generates the representations that are close enough to the ideals (0.86 on cosine similarity and 5.2 on squared euclidean distances), leading to the efficient in-domain retrieval. Two training losses, MSE and MLE, contribute significantly to the excellent performance of our approach. Among that, the MT loss is more important, as optimizing the model with MSE loss only could not achieve effective domain adaptation. Another observation is that our model could be smaller by introducing the token embedding and reusing the transformer encoder of the pre-trained E2E-ST model, causing small performance degradation.

The Impact of Datastore Size. As mentioned before, the datastore constructed by the bigger domain-specific text translation corpus seems to obtain better translation performance when using NPDA-kNN-ST. We investigate the performance differences caused by different datastore sizes on

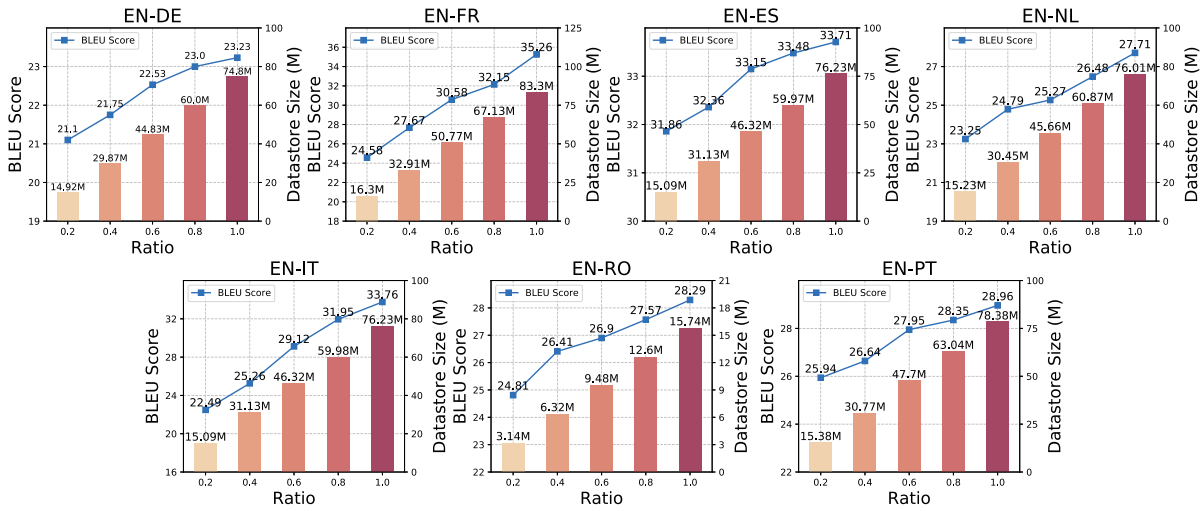


Figure 2: BLEU score [%] of NPDA-kNN-ST with different datastore sizes on the Europarl-ST dataset.

Europarl-ST. For each translation direction, we adopt a ratio range of $(0.2, 0.4, 0.6, 0.8, 1.0)$ to randomly sample from Europarl-MT corpus to build the datastore of different scales for quick experiments. The detailed results are shown in Figure 2. In general, the translation performance in all directions is positively correlated with the datastore size. More specifically, for EN-FR and EN-IT, model performance is increasing rapidly with the expansion of the datastore, exceeding 10 BLEU scores. The performance improvement in the DE, ES, NL and PT directions is relatively smooth. Since the overall datastore size of EN-RO is small, it still shows a reliable performance improvement. Thus, an enormous domain-specific text translation corpus further improves E2E-ST performance with NPDA-kNN-ST, but brings a larger datastore and slow inference speed, which is the trade-off in practice. Refer to Appendix A.6 for inference speed.

5 Related Work

Speech Translation. Previous ST methods (Ney, 1999; Sperber et al., 2017; Zhang et al., 2019; Lam et al., 2021) cascade the ASR and MT tasks. 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 is still very challenging. Various techniques have been proposed to ease the training process by using source transcriptions, including pre-training (Wang et al., 2020c), 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., 2021), agreement-based training (Du et al., 2022) and adapter tuning (Le et al., 2021). We first investigate the domain adaptation for E2E-ST and propose a non-parametric domain adaptation method to make the E2E-ST system more practical.

Domain Adaptation. The domain adaptation approaches in MT field are mainly divided into two categories: 1) model-centric, which focuses on modifying the model architecture or the training objective to learn domain-related information (Wang et al., 2017; Wuebker et al., 2018; Bapna et al., 2019; Guo et al., 2021); 2) data-centric, focusing on utilization of the monolingual corpus (Zhang and Zong, 2016; Zhang et al., 2018), synthetic corpus (Hu et al., 2019; Wei et al., 2020), or parallel corpus (Chu et al., 2017) in the specific domain for fine-tuning strategies. Recently, non-parametric methods provide a new paradigm for domain adaptation by retrieving the datastore of similar instances (Gu et al., 2018; Khandelwal et al., 2020; Zheng et al., 2021a,b; He et al., 2021; Wang et al., 2022). We follow this research line and extend this non-parametric method in the domain adaptation scenario for E2E-ST.

Alignment of Speech and Text Representation. Recent research has shown that unified speech and text representations are helpful for downstream tasks (Chung et al., 2018; Bapna et al., 2021; Akbari et al., 2021; Tang et al., 2021). SLAM (Bapna et al., 2021) train a single encoder on large-scale

text and speech data in a unsupervised manner, and further design corresponding speech-text alignment losses for downstream tasks. Tang et al. (2021) propose cross-attention regularization and online knowledge distillation to reduce the encoder representation differences between different modalities. In this work, we make the decoder's output representation for ST and MT tasks close by reducing the inconsistency of their representation in the training triple data to enable the construction of a cross-modality datastore.

6 Conclusion

In this paper, we present a novel non-parametric method that leverages in-domain bilingual text to achieve domain adaptation for the E2E-ST system. This approach builds the universal output representation for text and speech translation tasks by a carefully designed architecture and loss function. Based on that, a kNN classifier is introduced to improve translation performance with an external datastore constructed by the in-domain text translation data. Experimental results demonstrate that our proposed method obtains significant improvement over pre-trained E2E-ST models when using large-scale in-domain bilingual text corpus. In the future, we would like to explore the combination of our method and the large-scale E2E-ST model, such as LNA-D.

Limitations

The proposed approach constructs a datastore using text translation data from the target domain and utilizes kNN retrieval to assist pre-trained E2E-ST models for domain adaptation. Our approach achieves a significant performance improvement over the basic model, but also brings time and space costs, i.e., storage overhead for datastore and time costs for kNN retrieval. In practice, these costs are acceptable since we adopt FAISS to speed up kNN retrieval and reduce the storage requirement (as shown in Table 13). We also encourage future work to further investigate how to build a smaller datastore as well as improve the efficiency of kNN retrieval. Since the promising domain adaptation performance of our approach benefits from the strong foundation model, another interesting direction is to explore the combination of our method and the large-scale E2E-ST model, such as LNA-D.

Acknowledgements

This work was supported by the grants from National Natural Science Foundation of China (No.U20A20229, 62072423), CAAI-Huawei MindSpore Open Fund (CAAIXSJLJJ-2021-007B), and the USTC Research Funds of the Double First-Class Initiative (No.YD2150002009). We appreciate Linan Yue, Yanqing An and Li Wang for the fruitful discussions. We thank the anonymous reviewers for helpful feedback on early versions of this work.

References

Hassan Akbari, Li Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. 2021. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *ArXiv*, abs/2104.11178.

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. In *NAACL*.

Alexei Baevski, Henry Zhou, Abdel rahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *NeurIPS*.

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

Ankur Bapna, N. Arivazhagan, and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *EMNLP*.

Ankur Bapna, Yu-An Chung, Nan Wu, Anmol Gulati, Ye Jia, J. Clark, Melvin Johnson, Jason Riesa, Alexis Conneau, and Yu Zhang. 2021. Slam: A unified encoder for speech and language modeling via speech-text joint pre-training. *ArXiv*, abs/2110.10329.

Luisa Bentivogli, Mauro Cettolo, Marco Gaido, Alina Karakanta, Alberto Martinelli, Matteo Negri, and Marco Turchi. 2021. Cascade versus direct speech translation: Do the differences still make a difference? In *ACL-IJCNLP*.

Hugh A. Chipman, Edward I. George, Robert E. McCulloch, and Thomas S. Shively. 2021. mbart: Multidimensional monotone bart. *Bayesian Analysis*.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *ACL*.

Yu-An Chung, Wei-Hung Weng, Schrasing Tong, and James R. Glass. 2018. Unsupervised cross-modal alignment of speech and text embedding spaces. *ArXiv*, abs/1805.07467.

Qianqian Dong, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. 2021. Consecutive decoding for speech-to-text translation. In *AAAI*.

Yichao Du, Zhirui Zhang, Weizhi Wang, Boxing Chen, Jun Xie, and Tong Xu. 2022. Regularizing end-to-end speech translation with triangular decomposition agreement. In *AAAI*.

Mattia Antonino Di Gangi, R. Cattoni, L. Bentivogli, Matteo Negri, and M. Turchi. 2019. Must-c: a multilingual speech translation corpus. In *NAACL*.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alex Smola. 2006. A kernel method for the two-sample-problem. In *NeurIPS*.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. Search engine guided neural machine translation. In *AAAI*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Demi Guo, Alexander M. Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *ACL-IJCNLP*.

Chi Han, Mingxuan Wang, Heng Ji, and Lei Li. 2021. Learning shared semantic space for speech-to-text translation. In *Findings of ACL-IJCNLP*.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient nearest neighbor language models. In *EMNLP*.

Junjie Hu, M. Xia, Graham Neubig, and Jaime G. Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. In *ACL*.

S. Indurthi, HJ 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*, pages 7904–7908.

Javier Iranzo-Sánchez, Adrià Giménez-Pastor, J. Silvestre-Cerdà, Pau Baquero-Arnal, Jorge Civera Saiz, and Alfons Juan-Císcar. 2020a. Direct segmentation models for streaming speech translation. In *EMNLP*.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020b. Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7:535–547.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. In *ICLR*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Ali Can Kocabiyikoglu, Laurent Besacier, and Olivier Kraif. 2018. Augmenting librispeech with French translations: A multimodal corpus for direct speech translation evaluation. In *LREC*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MTSUMMIT*.

Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. 2021. Cascaded models with cyclic feedback for direct speech translation. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7508–7512.

Hang Le, J. Pino, Changhan Wang, Jiatao Gu, D. Schwab, and L. Besacier. 2021. Lightweight adapter tuning for multilingual speech translation. In *ACL-IJCNLP*.

Xian Li, Changhan Wang, Yun Tang, C. Tran, Yuqing Tang, J. Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Multilingual speech translation from efficient finetuning of pretrained models. In *ACL-IJCNLP*.

Yuchen Liu, Hao Xiong, Zhongjun He, Jiajun Zhang, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-end speech translation with knowledge distillation. In *INTERSPEECH*.

Yuchen Liu, Jiajun Zhang, Hao Xiong, Long Zhou, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2020. Synchronous speech recognition and speech-to-text translation with interactive decoding. In *AAAI*.

Dominik Machácek, Matús Zilinec, and Ondrej Bojar. 2021. Lost in interpreting: Speech translation from source or interpreter? *ArXiv*, abs/2106.09343.

Yuxian Meng, Xiaoya Li, Xiayu Zheng, Fei Wu, Xiaofei Sun, Tianwei Zhang, and Jiwei Li. 2021. Fast nearest neighbor machine translation. *ArXiv*, abs/2112.08152.

H. Ney. 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.

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

Kishore Papineni, S. Roukos, T. Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Daniel S. Park, William Chan, Y. Zhang, C. Chiu, Barret Zoph, E. D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *INTERSPEECH*.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in nlp—a survey. *ArXiv*, abs/2006.00632.

Matthias Sperber, Graham Neubig, J. Niehues, and A. Waibel. 2017. Neural lattice-to-sequence models for uncertain inputs. In *EMNLP*.

Matthias Sperber, Graham Neubig, J. Niehues, and A. Waibel. 2019. Attention-passing models for robust and data-efficient end-to-end speech translation. *Transactions of the Association for Computational Linguistics*.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021. Improving speech translation by understanding and learning from the auxiliary text translation task. In *ACL)*, pages 4252–4261.

Laura Cross Vila, Carlos Escolano, José A. R. Fonollosa, and Marta Ruiz Costa-jussà. 2018. End-to-end speech translation with the transformer. In *IberSPEECH*.

Changhan Wang, J. Pino, Anne Wu, and Jiatao Gu. 2020a. Covost: A diverse multilingual speech-to-text translation corpus. In *LREC*.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and J. Pino. 2020b. Fairseq s2t: Fast speech-to-text modeling with fairseq. In *AACL*.

Chengyi Wang, Yu Wu, Shujie Liu, M. Zhou, and Zhenglu Yang. 2020c. Curriculum pre-training for end-to-end speech translation. In *ACL*.

Dongqi Wang, Haoran Wei, Zhirui Zhang, Shujian Huang, Jun Xie, Weihua Luo, and Jiajun Chen. 2022. Non-parametric online learning from human feedback for neural machine translation. In *AAAI*.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *EMNLP*.

Hao-Ran Wei, Zhirui Zhang, Boxing Chen, and Weihua Luo. 2020. Iterative domain-repaired back-translation. In *EMNLP*.

Ron J. Weiss, J. Chorowski, Navdeep Jaitly, Yonghui Wu, and Z. Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *INTERSPEECH*.

Joern Wuebker, Patrick Simianer, and John DeNero. 2018. Compact personalized models for neural machine translation. In *EMNLP*.

Biao Zhang, Ivan Titov, B. Haddow, and Rico Sennrich. 2020. Adaptive feature selection for end-to-end speech translation. In *Findings of EMNLP*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *EMNLP*.

Peidong Zhang, Boxing Chen, Niyu Ge, and Kai Fan. 2019. Lattice transformer for speech translation. In *ACL*.

Zhirui Zhang, Shujie Liu, Mu Li, M. Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *AAAI*.

Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021a. Adaptive nearest neighbor machine translation. In *ACL-IJCNLP*.

Xin Zheng, Zhirui Zhang, Shujian Huang, Boxing Chen, Jun Xie, Weihua Luo, and Jiajun Chen. 2021b. Non-parametric unsupervised domain adaptation for neural machine translation. In *Findings of EMNLP*.

A Appendix

A.1 Dataset Statistics

The statistics of MuST-C and Europarl-ST datasets are shown in Table 5 and 6.

A.2 Dataset Preprocessing

We follow the FAIRSEQ S2T (Wang et al., 2020b) recipes to perform data pre-processing. For speech data, both in Europarl-ST 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. The SpecAugment method (Park et al., 2019) is used in all experiments and we remove samples consisting of more than 3k frames. For external text translation data, we delete the bilingual data in Europarl-MT that intersects with validation/test sets of the Europarl-ST dataset. We adopt unigram sentencepiece[2] to build 5K and 8K sub-word vocabularies for the transcriptions and the translations, respectively. For the multilingual model, both vocabulary sizes are set to 10K. In all experiments, the MuST-C dataset is only used to construct the sub-word dictionary.

A.3 Implementation Details

All experiments are implemented based on the FAIRSEQ[3] (Ott et al., 2019) toolkit. For the model structure of all baselines, it consists of two one-dimensional convolutional layers with a downsampling factor of 4, 12 transformer encoder layers, and 6 transformer decoder layers. The additional encoder in our approach includes 12 transformer encoder layers and token embedding, and all parameters are initialized randomly. The input embedding size of the transformer layer is 256, the FFN layer dimension is 1024, and the number of self-attention heads is 4. We adopt 6 transformer decoder layers and the same parameters for LM training. For the multilingual model, the above parameters are set to 512, 2048 and 8 respectively. During training, we deploy the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 2e-3 and 10K warm-up updates to optimize model parameters. Both label smoothing coefficient and dropout rate are set to 0.1. The batch size is set to 20K tokens, and we accumulate the gradient for every 4 batches. We train all models with one Tesla-V100 GPU and set patience to 5 to select the best checkpoint on the validation set. The FAISS[4] (Johnson et al., 2021) is leveraged to construct the in-domain datastore and carry out fast nearest neighbor search. We utilize the FAISS to learn 8192 cluster centroids for each translation direction. During inference, the beam size and length penalty are set to 5 and 0.6 for all methods and we search 64 clusters for each target token when using FAISS. The performance of kNN-MT and NPDA-kNN-ST is highly related to the choice of hyper-parameters. The hyper-parameters (k, λ and T) for kNN retrieval are tuned on the in-domain validation set. We adopt grid search of $k \in \{4, 8, 16, 32\}$, $\lambda \in \{0.1, 0.2, ..., 0.9\}$ and $T \in \{1, 10, 20, 50, 100, 200\}$ for each translation direction on Europarl-ST/MuST-C validation sets when using kNN-MT and NPDA-kNN-ST. The optimal choices of different datasets are shown in Table 7, 8 and 9.

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

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

	Speech Duration	Train Pairs	Dev Pairs	Test Pairs
DE	408 hrs	225,278	1,419	2,588
FR	492 hrs	269,256	1,409	2,579
ES	504 hrs	260,050	1,313	2,450
IT	465 hrs	248,155	1,305	2,521
NL	442 hrs	243,516	1,419	2,563
PT	385 hrs	201,462	1,365	2,449
RO	432 hrs	231,471	1,366	2,503
RU	489 hrs	259,531	1,313	2,460

Table 5: The statistics of all EN-X translation directions in the MuST-C dataset.

	Speech Duration	Train Pairs	Dev Pairs	Test Pairs
DE	83 hrs	32,629	1,321	1,254
FR	81 hrs	31,778	1,282	1,215
ES	81 hrs	31,608	1,273	1,268
IT	80 hrs	29,553	1,123	1,131
NL	80 hrs	31,402	1,270	1,236
PT	81 hrs	31,751	1,295	1,263
RO	72 hrs	28,599	1,071	1,096

Table 6: The statistics of all EN-X translation directions in the Europarl-ST dataset.

A.4 Statistics of Datastore

The statistics of datastore used in our experiments are shown in Table 10 and 11. Note that the datastore statistics of kNN-MT are exactly the same as those of NPDA-kNN-ST, due to the same number

[4]https://github.com/facebookresearch/faiss

	DE	FR	ES	NL	IT	RO	PT
				kNN-MT			
$\boldsymbol{k}$	16	16	16	16	16	8	8
$\boldsymbol{\lambda}$	0.5	0.7	0.6	0.6	0.7	0.6	0.6
$\boldsymbol{T}$	10	20	10	20	20	50	50
				NPDA-kNN-ST			
$\boldsymbol{k}$	16	32	16	16	32	16	32
$\boldsymbol{\lambda}$	0.5	0.7	0.6	0.7	0.7	0.7	0.7
$\boldsymbol{T}$	10	10	10	20	10	10	10
				NPDA-kNN-ST$^+$			
$\boldsymbol{k}$	32	4	8	8	4	8	8
$\boldsymbol{\lambda}$	0.8	0.8	0.8	0.8	0.8	0.8	0.8
$\boldsymbol{T}$	10	10	10	10	10	10	10

Table 7: The optimal choice of hyper-parameters for all EN-X translation directions on Europarl-ST validation set in domain adaptation experiments.

	DE	FR	ES	NL	IT	RO	PT
				kNN-MT			
$\boldsymbol{k}$	8	16	8	16	16	16	16
$\boldsymbol{\lambda}$	0.5	0.5	0.6	0.6	0.8	0.1	0.7
$\boldsymbol{T}$	200	20	50	50	100	50	50
				NPDA-kNN-ST			
$\boldsymbol{k}$	16	32	16	16	8	32	32
$\boldsymbol{\lambda}$	0.6	0.6	0.6	0.7	0.5	0.3	0.2
$\boldsymbol{T}$	100	20	20	100	100	10	20

Table 8: The optimal choice of hyper-parameters for all EN-X translation directions on MuST-C validation set in domain adaptation experiments.

of ground truth tokens when building datastores.

A.5 Comparison with Cascade Methods

Table 12 shows the performance comparisons of NPDA-kNN-ST$^+$ with different cascade systems, including Cascade-SP, Cascade-ST and Cascade-ST*. The Cascade-SP is built by Iranzo-Sánchez et al. (2020b) and we further reproduce cascade methods with two dictionary settings. Cascade-ST adopts the same dictionary as NPDA-kNN-ST$^+$, while Cascade-ST* constructs a 40K byte-pair dictionary with MuST-C and Europarl-MT datasets for NMT model. Both Cascade-ST and Cascade-ST* adopt the same ASR model that is trained on the MuST-C dataset. The model structure of NMT in these two cascade systems contains a 6-layer transformer encoder and a 6-layer transformer decoder, in which input dimension, FFN layer dimension and attention heads are 512, 1024 and 4 respectively. We can observe that NPDA-kNN-ST$^+$ outperforms Cascade-ST in all translation directions, and the inference speed is more competitive (see Table 13). Cascade-ST* obtains significant improvement over Cascade-ST thanks to the better dictionary built on both MuST-C and Europarl-MT datasets. We believe that our proposed method could also benefit from such dictionary, but it requires leveraging such a dictionary to train the E2E-ST model at the beginning.

	DE	FR	ES	NL	IT	RO	PT	RU
				kNN-MT				
$\boldsymbol{k}$	8	16	8	16	16	16	16	8
$\boldsymbol{\lambda}$	0.2	0.3	0.2	0.2	0.3	0.1	0.2	0.3
$\boldsymbol{T}$	10	20	20	50	10	50	10	20
				NPDA-kNN-ST				
$\boldsymbol{k}$	32	16	8	16	8	32	16	8
$\boldsymbol{\lambda}$	0.4	0.3	0.3	0.3	0.3	0.3	0.2	0.3
$\boldsymbol{T}$	10	20	20	50	10	10	20	20

Table 9: The optimal choice of hyper-parameters for all EN-X translation directions on MuST-C validation set in E2E-ST experiments.

Intuitively, introducing in-domain triplet data could yield a better contextual representation for our method, which may help kNN to retrieve more accurate candidates and improve the final performance. We directly apply our method for the E2E-ST-FT model, in which we use in-domain data (Europarl-ST) to build the aligning representation (named NPDA-kNN-ST++).We conduct the experiments in the En-X translation directions, and the results are illustrated in Table 12. As we expected, NPDA-kNN-ST++ achieves better performance than NPDA-kNN-ST+.

A.6 Inference Speed Comparison

We compare the inference speed of four methods (E2E-ST-Base, E2E-ST-FT, NPDA-kNN-ST and NPDA-kNN-ST+) on Europael-ST EN-DE test set with different hyper-parameters ($k = 4, 8, 16, 32$) and batch sizes (batch $= 1, 8, 16, 32$). As shown in Table 13, the inference time of NPDA-kNN-ST increases with the bigger datastore size. The larger datastore means that more keys need to be retrieved during the inference phase, which reduces the inference speed. Nonetheless, when in-domain speech translation data is inaccessible to fine-tune the E2E-ST-Base model, it is still worth sacrificing part of the time and storage for higher performance.

		DE	FR	ES	NL	IT	RO	PT
NPDA-kNN-ST	$(\mathcal{K}, \mathcal{V})$	1,220,631	1,265,862	1,160,737	1,153,677	1,139,009	1,083,567	1,194,161
	Datastore	597 MB	619 MB	568 MB	568 MB	557 MB	530 MB	584 MB
	Faiss index	93 MB	96 MB	89 MB	89 MB	87 MB	83 MB	91 MB
NPDA-kNN-ST$^+$	$(\mathcal{K}, \mathcal{V})$	74,795,371	83,303,733	76,226,723	76,011,171	75,981,836	15,738,321	78,375,866
	Datastore	36 GB	40 GB	37 GB	37 GB	37 GB	7.6 GB	38 GB
	Faiss index	3.1 GB	3.5 GB	3.2 GB	3.1 GB	3.1 GB	224 MB	3.2 GB

Table 10: The statistics of datastore for all EN-X translation directions on Europarl-ST dataset.

		DE	FR	ES	NL	IT	RO	PT	RU
NPDA-kNN-ST (Bilingual)	$(\mathcal{K}, \mathcal{V})$	5,909,910	7,843,906	7,028,102	6,006,360	6,591,640	6,339,525	5,345,744	7,150,960
	Datastore	2.9 GB	3.8 GB	3.4 GB	2.9 GB	3.2 GB	3.1 GB	2.6 GB	3.5 GB
	Faiss index	415 MB	547 MB	491 MB	421 MB	461 MB	444 MB	376 MB	500 MB
NPDA-kNN-ST (Multilingual)	$(\mathcal{K}, \mathcal{V})$	7,587,793	9,530,628	8,507,191	7,572,305	8,121,129	7,819,137	6,626,153	9,460,741
	Datastore	7.3 GB	9.1 GB	8.2 GB	7.3 GB	7.8 GB	7.5 GB	6.4 GB	9.1 GB
	Faiss index	538 MB	671 MB	601 MB	537 MB	575 MB	554 MB	472 MB	666 MB

Table 11: The statistics of datastore for all EN-X translation directions on MuST-C dataset.

Model	**Used Data**				**Params. (M)**	**Target Language**							
	MC	**EP-ST**	**EP-MT**	**Extra.**	**Tuned/Total**	**DE**	**FR**	**ES**	**NL**	**IT**	**RO**	**PT**	**Avg.**
E2E-ST-Base	✓	×	×	×	0.0/31.1	15.71	16.45	23.49	16.06	14.25	16.95	18.28	17.31
Cascade-SP	×	✓	×	✓	/	22.40	23.40	28.00	/	/	/	/	/
Shallow Fusion	✓	×	✓	×	6.8/37.9	17.72	22.66	26.25	20.12	18.77	20.58	21.59	21.67
E2E-JT-ST-MT	✓	×	✓	×	48.1/48.1	22.10	33.62	31.28	21.35	22.18	23.21	23.62	25.34
Cascade-ST	✓	×	✓	×	67.7/67.7	22.63	34.58	33.08	25.43	26.05	26.60	26.29	27.81
Cascade-ST*	✓	×	✓	×	88.4/88.4	24.71	34.60	33.70	26.55	29.94	26.38	30.18	29.44
NPDA-kNN-ST$^+$	✓	×	✓	×	17.1/48.1	23.23	35.26	33.71	27.71	33.76	28.29	28.96	30.13
NPDA-kNN-ST^{++}	✓	✓	✓	×	48.1/48.1	**24.90**	**35.95**	**34.32**	**28.44**	**34.69**	**29.68**	**32.83**	**31.54**

Table 12: BLEU score [%] of different methods on the Europarl-ST dataset. "Cascade-SP " is the cascade model built by Iranzo-Sánchez et al. (2020b). We also reproduce the performance of cascade methods with two dictionary settings, in which "Cascade-ST" adopts the same dictionary as our method and 'Cascade-ST*' constructs the dictionary with both MuST-C and Europarl-MT datasets.

Model	**Hard Disk Space**		k	**Inference Speed** (ms/sentence)			
	Datastore	Faiss Index		batch=1	batch=8	batch=16	batch=32
E2E-ST-Base	-	-	0	347.8	64.7	37.1	21.9
E2E-ST-FT	-	-	0	349.5 (×1.00)	63.7 (×0.98)	37.5 (×1.01)	21.5 (×0.98)
Cascade-ST	-	-	0	690.3 (×1.98)	127.3 (×1.97)	68.8 (×1.85)	40.1 (×1.83)
Cascade-ST*	-	-	0	788.0 (×2.27)	142.6 (×2.20)	80.1 (×2.16)	46.2 (×2.11)
NPDA-kNN-ST	597 MB	93 MB	4	375.8 (×1.08)	70.5 (×1.09)	40.3 (×1.09)	24.5 (×1.12)
			8	379.2 (×1.09)	70.9 (×1.10)	41.4 (×1.12)	24.7 (×1.13)
			16	381.4 (×1.10)	70.1 (×1.08)	40.8 (×1.10)	24.2 (×1.11)
			32	374.9 (×1.08)	71.5 (×1.11)	39.8 (×1.07)	24.2 (×1.11)
NPDA-kNN-ST$^+$	36 GB	3.1 GB	4	419.3 (×1.21)	94.1 (×1.45)	63.4 (×1.71)	46.9 (×2.14)
			8	419.6 (×1.21)	96.3 (×1.49)	64.7 (×1.74)	46.9 (×2.14)
			16	420.6 (×1.21)	94.3 (×1.46)	63.6 (×1.71)	47.0 (×2.15)
			32	420.2 (×1.21)	93.3 (×1.44)	64.0 (×1.73)	46.9 (×2.14)

Table 13: Inference speed of different methods in EN-DE direction of Europael-ST. All results are the average of three runs on a server with 96-core Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz and Tesla V100-SXM2-32GB GPU.

Note that NPDA-kNN-ST only needs to load the Faiss index to perform kNN retrieval and we could further replace the prediction way used in our paper with other kNN variants (He et al., 2021; Meng et al., 2021) to reduce the inference time.