Efficient Speech Translation with Pre-trained models

Anonymous ACL submission

Abstract

When building state-of-the-art speech translation models, the need for large computational resources is a significant obstacle due to the large training data size and complex models. The availability of pre-trained models is a promising opportunity to build strong speech translation systems efficiently. In a first step, we investigate efficient strategies to build cascaded and end-to-end speech translation systems based on pre-trained models. Using this strategy, we can train and apply the models on a single GPU. While the end-to-end models show superior translation performance to cascaded ones, the application of this technology has a limitation on the need for additional end-to-end training data. In a second step, we proposed an additional similarity loss to encourage the model to generate similar hidden representations for speech and transcript. Using this technique, we can increase the data efficiency and improve the translation quality by 6 BLEU points in scenarios with limited end-to-end training data.

1 Introduction

Speech translation (ST) is a process of recognizing the audio of the source language and translating it into the text of the target language. Automatic ST is widely used in daily cases, such as remote meetings, distance education, and online communication, to lower language barriers and enable efficient communication. There are two popular approaches to building ST systems: cascaded and end-to-end. The cascaded approach uses an Automatic Speech Recognition (ASR) model to generate the transcript from the audio in the source language and then a Machine Translation (MT) model to translate it into the target language. On the contrary, the end-to-end approach (Berard et al., 2016; Weiss et al., 2017) does not have the intermediate transcript and directly translates the speech in the source language into the target languages. The cascaded system has advantages in data availability and flexibility to incorporate with new ASR and MT developments. In contrast, the end-to-end system goes outstanding with mitigating error propagation, improving computational efficiency, and decreasing latency.

Building a successful ST system from scratch is not always possible because of limitations in training data and computation resources. Therefore, a promising approach is fine-tuning pre-trained models on the speech translation task. In practical scenarios, computation limitation is one challenge for building a successful ST model. Recent works indicate that increasing pre-trained model size still leads to performance improvements on downstream NLP tasks (Sanh et al., 2020). Consequently, the size of the pre-trained model has been getting larger and larger, leading to sometimes impractical to fine-tune the pre-trained models. Data scarcity is another challenge to building ST models. Collecting the end-to-end data is expensive for finding high-quality data, aligning audio, transcript, and translation, filtering wrong and poor alignment. In order to address the above challenges, this research focuses on improving computational efficiency and data efficiency with the usage of pre-trained models for speech translation.

Our first contribution is to compare the performances between the cascaded system and the end-to-end system, both using pre-trained models. With the advent of deep learning, the end-to-end approach has been developed recently and proven to have comparable performance to the cascaded approach (Niehues et al., 2018; Ansari et al., 2020; Bentivogli et al., 2021). However, there is no claim about which approach has clear advantages on performance. This work investigates the performance comparison of two systems by directly combining the pre-trained model without architecture modification. Our result shows that the end-to-end system outperforms the cascaded system on the English-German speech translation of the CoVoST2 dataset

in terms of fine-tuning efficiency and accuracy.

As the second contribution, we propose two fine-tuning strategies to improve computational efficiency. Rather than fine-tuning the entire ST model, the first strategy is fine-tuning the encoder of the MT model. The strategy is motivated to bridge the discrepancy between the generated latent speech representation and the text. Besides, we present fine-tuning adapter is an effective alternative for speech translation. Three Bidirectional Long Short-Term Memory (BLSTM) layers get inserted between the ASR and MT module in the end-to-end model. The adapter approach fine-tunes less than one-tenth parameter and achieves comparable performance to the cascaded model.

The third contribution is that we present a novel similarity loss to mitigate the data scarcity issue. Unlike the end-to-end data that are challenging to acquire, speech-to-transcript data is more accessible. We develop the similarity loss that measures the difference between latent representations for the audio and the transcript. The motivation is that the speech translation model should represent similar hidden state representations for aligned audio and transcript. Consequently, minimizing the similarity loss is proposed to improve speech translation performance. Our result shows that involving similarity loss improves data efficiency and boosts model performance.

2 Related work

Fine-tuning the pre-trained models is an effective approach to building ST models. (Stoian et al., 2020) proves that the speech translation task benefits from the language-universal phonetic information learned by the pre-trained ASR model. Besides, (Alinejad and Sarkar, 2020) demonstrates the pre-trained machine translation contributes to speech translation. Usually, the fine-tuning dataset requires fewer data and computational resources than that of pre-training. Thus, this is a practical approach to dealing with limited resources.

The cascaded system can directly leverage the entire pre-trained models by feeding the transcript generated from the ASR module to the MT module. However, the end-to-end system requires adaptation techniques to use the pre-trained models. For the end-to-end system, leveraging the pre-trained model was firstly proposed in (Bérard et al., 2018). The approach is initializing the encoder and decoder of the ST model with the parameters from the pre-trained ASR encoder and MT decoder, then fine-tuning the ST model with the end-to-end training data. (Jia et al., 2019; Inaguma et al., 2019; Gaido et al., 2021) shows the approach is effective when the pre-training and fine-tuning domain are the same. (Le et al., 2021; Li et al., 2021) proves the approach benefits to leveraging the knowledge from other domains by using the pre-trained parameters. However, this approach discards pre-trained sub-nets and improves fine-tuning efficiency. Thus, it might lose valuable semantic information captured by the sub-nets (Wang et al., 2019) and consequently harm model performance.

The computational limitation is one major obstacle to the end-to-end ST system. (Li et al., 2021) presents that fine-tuning the layer normalization and multi-head attention parameters is effective to improve computational efficiency. (Le et al., 2021) showed that fine-tuning residual adapt modules that are transplanting between the encoders and decoders is a promising approach. These researches show the possibility of fine-tuning components of pre-trained models and motivate this work to explore other efficient fine-tuning approaches.

The lack of end-to-end training data is another obstacle to the end-to-end ST system. (Weiss et al., 2017; Bérard et al., 2018; Anastasopoulos and Chiang, 2018) shown multi-task learning improves speech translation performance by weighting the losses of ASR, MT and ST. Besides, (Liu et al., 2019; Gaido et al., 2020) proven the effectiveness of knowledge distillation by learning a student ST model from a teacher MT model using the ASR data. In addition, (Jia et al., 2019; Pino et al., 2020) present the benefits of involving synthetic data using the pre-trained models. The above methods leverage the available data resources, i.e., speech-to-transcript and transcript-to-translation data, to reduce the reliance on the end-to-end training data.

3 Speech translation using Pre-trained models

For building a ST system for a new task, e.g., translating a single sentence from English to German, large amounts of training data typically need to be collected. Besides, training a ST model requires significant computational resources. In this work, we want to increase the efficiency of building a speech translation system by facilitating pre-trained speech recognition and text translation models, which are nowadays widely available for a

large variety of languages. These pre-trained models are typically trained in a self-supervised way using annotated data. However, the annotated data is often not from the targeted domain.

As shown in Figure 1, this research proposes the cascaded and end-to-end combinations of pre-trained models to build ST systems. A first baseline approach is to combine the two models in a cascaded manner. However, the cascaded approach has several drawbacks. Therefore, we also investigated possibilities to combine the two pre-trained models into one end-to-end speech translation model.

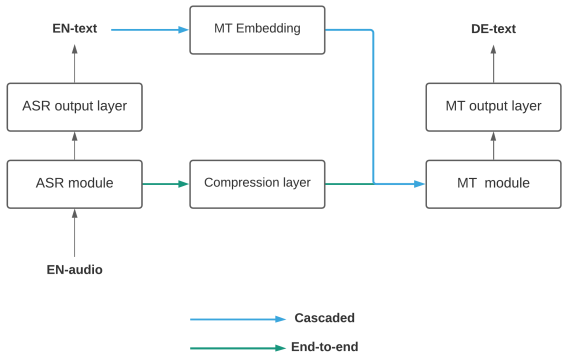

Figure 1: Overview of cascaded and end-to-end combinations with the pre-trained models.

3.1 Cascaded speech translation

The cascaded system consists of two sub-modules: an ASR component and a MT component.

In the ASR stage, the pre-trained ASR module inputs acoustic data and outputs a sequence of speech representations. The generated representations then get passed through the output layer to map the representations to characters. The output layer is a linear transformation whose input size equals the hidden state size, and output size equals the vocabulary size. Afterwards, we use the CTC algorithm to generate the most probable character sequence. Figure 2 illustrates the generation stage. Firstly, each representation maps to a character that has the highest probabilities. Next, the generated character sequence gets collapsed by merging the repeated characters and removing the non-semantic tokens.

In the following MT stage, the transcript gets first segmented into sub-words according to the vocabulary of the translation system. Then the input is fed into the pre-trained model to generate translation.

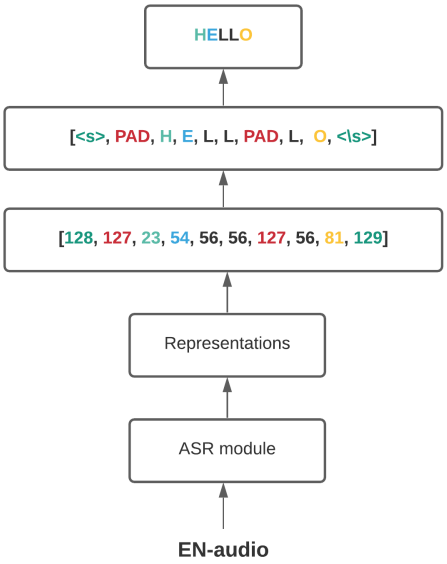

Figure 2: Illustration of CTC algorithm in ASR generation. The numbers indicate the indices of the highest probability for the representations. <s> and <\s> indicate the beginning and end of the sentence. PAD indicates the blank token to distinguish repetition and separation.

3.2 End-to-end speech translation

Instead of generating the intermediate transcript, we combine two pre-trained models by feeding the speech representation generated from the ASR module to the MT module. In order to achieve this, we addressed three challenges: How to integrate both models given the different types of representation granularity used within the models? How to enable fine-tuning of the combined model given the larger size of the two pre-trained models? How to minimize the need for additional end-to-end training data?

Integration

The output of the ASR model is a speech representation for a fixed size input window. However, the MT module assumes one embedding for each sub-word. Therefore, for the same segment, the lengths of speech and text sequences are very different. The length inconsistency is hard to learn by the ST model, harming model performance.

For mitigating the length inconsistency, we insert a CTC-based compression layer between two modules (Gaido et al., 2021). The compression layer uses the CTC algorithm to determine which speech representations are aligned to the same character. Then the adjacent representations aligned to the

same character are averaged, therefore compressing the redundant and uninformative vectors. While we significantly reduce the length inconsistency by this approach, there is still an inconsistency remaining. The compressed representations have one speech representation for each character, while the original MT system uses one representation per sub-word. We did not reduce the representation further since the average operation may significantly lose information if it performs over a very long sequence, and we expect the MT module to learn the mapping between characters and sub-words.

Fine-tuning

One additional challenge when using the end-to-end model compared to the cascaded model is that we need to run both pre-trained models in parallel, while they can be loaded one after the other in the cascaded model. While the end-to-end model has the advantage of lower latency, it needs significantly more memory. The memory consumption is especially challenging during training, where the derivations also need to be stored beside the weights. Therefore, we proposed a two-stage training for the end-to-end system: In the first stage, we fine-tune the pre-trained models on the individual speech recognition or text translation tasks. In this case, all parameters of the model get updated. We investigate whether it is helpful to fine-tune the ASR, MT, or both components. In a second stage, we jointly train the entire model on the end-to-end task, but only train part of the parameters:

1. MT encoder: Rather than fine-tuning all parameters, we propose only to fine-tune the encoder of the MT module and freeze the rest. The motivation is to solve the discrepancy between the speech representation from the ASR module and the text representation for the decoder.

2. Adapter: Fine-tuning adapter has been proven an efficient approach for machine translation (Bapna et al., 2019) and a promising approach for multilingual speech translation (Le et al., 2021). As the name suggests, adapter works on adapting the model to new tasks. Specifically, only the adapt layers are fine-tuned, and the rest parameters are frozen. We propose a simple adapter of three BLSTM layers. As shown in Figure 3, the adapter gets inserted between the ASR and MT modules. In this way, semantic information of these two modules keeps integrated. BLSTM layer has backward and forward directions and concatenates the hidden states of both directions, consequently preserving information from both the past and the future at each time step and matching the characteristics of ST tasks.

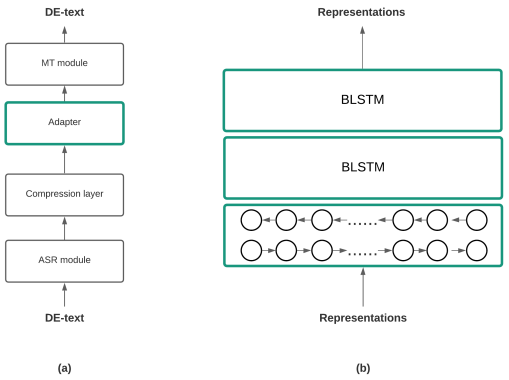

Figure 3: Illustration of the adapter. (a) presents the workflow of end-to-end speech translation with the adapter. (b) illustrates the composition of the adapter.

Data efficiency

Finally, the drawback of end-to-end models is the need for additional end-to-end training data. While the cascaded models are only trained on ASR and text translation training data, the end-to-end model also needs data aligned between the source language audio and the target language text. Since this data is typically difficult to acquire, we investigate possibilities to limit the need for this data.

We propose a similarity loss function to increase the similarity by the speech representation of the ASR model and the source language representation of the text translation encoder. The advantage is that this only requires speech-to-transcript data, not end-to-end data.

Inspired by (Pham et al., 2019), the similarity loss minimizes the difference between the audio and text representation from the MT module encoder. Figure 4 illustrates the workflow of similarity loss function. The last hidden states of the MT encoder get averaged over time steps to produce the representing vectors. Afterwards, the Mean Squared Error gets calculated between the representing vectors as the loss.

In standard bilingual ST, the model learns the target language from translation data. As we propose only using the speech-to-transcript training data, the model does not know the target language.

Therefore, we implement the target forcing mechanism by generating a target-language-specific embedding with the MT pre-trained embedding layer and prepending the embedding to the speech representations (Gangi et al., 2019).

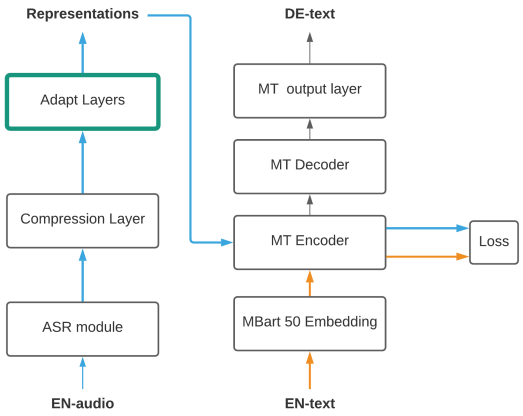

Figure 4: An overview of the similarity loss with an end-to-end system.

In fine-tuning, if the similarity loss is back-propagated to all parameters, the parameters might be forced to be zeros to minimize the similarity loss to the optimum zero. Consequently, we implement the similarity with the end-to-end system together with the adapter. We only fine-tune the adapter and freeze the rest.

4 Experimental Setup

4.1 Dataset

The proposed approaches are evaluated on the CoVoST2 (Wang et al., 2020) dataset.CoVoST2 is a speech-to-text translation corpus. Precisely, one data sample consists of audio, transcript and translation. The dataset was collected from more than 10 thousand speakers and 60 accents. Therefore it is a robust and comprehensive testbed for ST tasks. In this research, we focus on the English-Germen translation direction. The average audio length is 5 seconds. The transcript and translation have an average of 60 and 66 words, respectively.

4.2 Metric

This research aims to explore the cascaded and end-to-end combinations of pre-trained models. Therefore it has three tasks: ASR, MT and ST. This research calculates Word Error Rate (WER) for ASR evaluation using VizSeq [1] (Changhan Wang, 2019) and Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002) for MT and ST evaluation using sacreBLEU [2] (Post, 2018).

WER counts the wrong words of the speech recognition result and divides the number by the total number of words of the reference. The lower the WER score is, the better performance the model has. BLEU is calculated based on N-grams and works on word level. A higher BLEU score indicates a better model performance.

4.3 Pre-processing

Firstly, we remove the double quotes at the beginning and the end of the transcript and translation. Next, the uncompleted data which has no transcript or translation gets removed. Each languages pair has about three uncompleted data. Besides, we build a custom vocabulary for ASR tasks. Although the transcript is in English, some characters are out of the English alphabet for place and name in CoVoST2. Instead of regarding these characters as unknown, we extract all distinct characters of the training data and build our vocabulary. The vocabulary has 128 tokens composed of all distinct letters among transcripts, two tokens for the beginning and end of the sentence, one blank token, and one unknown token.

4.4 Pre-trained models

This research uses a pre-trained wav2vec2.0 (Baevski et al., 2020) for speech recognition task and a pre-trained MBart50 (Liu et al., 2020; Tang et al., 2020) for machine translation task.

wav2vec2.0 is a self-supervised model to learn powerful representations from raw audio data. It consists of a multi-layer convolutional feature encoder that learns latent representations from the raw speech, a context network that follows the Transformer architecture to learn relative positional embedding, and a quantisation model that discretises feature encoders and select the quantised representations. We select the pre-trained wav2vec2.0 with the large architecture. The model is pre-trained on Librispeech (Panayotov et al., 2015) corpus that contains 960hours of unlabeled data and then fine-tuned with the audio-transcript data pair from the same dataset. The pre-training loss consists of the contrastive loss to generate accurate representation

[1] https://github.com/facebookresearch/vizseq

[2] https://github.com/mjpost/sacrebleu

and the diversity loss to encourage equal distribution. The fine-tuning loss is Connectionist Temporal Classification (CTC) loss (Graves et al., 2006) that is used for the downstream speech recognition tasks.

MBart50 is a sequence-to-sequence denoising auto-encoder model that has been pre-trained on large-scale monolingual and multilingual data. The model gets optimised by minimising a denoising loss function that randomly masks 35% of the input. MBart50 is pre-trained with 50 languages from diverse language families to adapt to downstream MT tasks.

5 Results

We evaluated the cascaded model as well as the different proposed methods for the end-to-end model on the CoVoST2 dataset. In addition, we investigated how we can reduce the need for additional end-to-end training data in a second series of experiments.

5.1 Cascaded system

In cascaded combination, we explore the efficiency of fine-tuning each component. Firstly we experiment on initializing with parameters of the pre-trained models to provide references. Then, we fine-tune the pre-trained wav2vec2.0 and MBart50 models with speech-to-transcript and transcript-to-translation training data, respectively. We experiment with different combinations of the pre-trained and fine-tuned parameters to explore the efficiency.

As Table 1 shows, for the cascaded combination, fine-tuning either ASR or MT module benefits performance, but fine-tuning the MT module contributes more to speech translation compared with fine-tuning the ASR module. Fine-tuning both modules leads to the best improvements of 4.9 BLEU points compared with no fine-tuning and 3.3 BLEU points improvement on the CoVoST2 Cascaded. Besides, we observe that improvement roughly equals the sum of the gains from fine-tuning each model.

Rather than fine-tune the entire module, we find that fine-tuning the encoder of the MT model slightly improves performance according to cascaded experiments 2 and 5. Compared with cascaded experiments 1, 4, and 5, we observe that with 25% parameters of MBart50, fine-tuning the encoder achieves 41% improvements.

5.2 End-to-end system

In the end-to-end combination, we apply a two-stage training scheme. The first stage is fine-tuning the pre-trained models, and the second stage is fine-tuning the end-to-end model with the end-to-end data. The two-stage training is motivated to reduce the cost of computational resources by leveraging the available data resource as described in Section 3.2 .

In light of building the cascaded model directly using the ASR and MT modules, we first experiment with the initialization on the end-to-end system. E2E Experiments 1, 2, and 3 show that the end-to-end model does not work without the end-to-end training data.

We experiment with the end-to-end training data on different fine-tuning modules in the first stage and fine-tuning the MT encoder or adapter in the second stage to address computation efficiency. We find that fine-tuning the ASR module in the first stage is necessary to enable speech translation for the end-to-end combination. On the contrary, fine-tuning the MT module in the first stage barely influences speech translation performance.

In conclusion, the two-stage approach that trains the ASR component independently in the first stage and the MT encoder using the end-to-end data in the second stage is promising to build end-to-end ST systems. With this configuration (E2E 4), we can achieve a better translation quality than the cascaded system. Furthermore, the end-to-end combination achieves 4.5 BLEU points improvements compared with the CoVoST2 E2E.

A second approach is to integrate an additional adapt layer. In this case, we only need to train 67M parameter instead of 150M ones. The final performance is 2 BLEU point worse. However, the pre-trained MT model is not changed and therefore can for example still be used to perform text translation in parallel.

5.3 Data efficiency

In a second series of experiments we evaluated the data efficient of the end-to-end model with respect to end-to-end training data. Therefore, we investigated the effect of the similarity loss on the best performing system using the adapt layer (E2E 7). In order to use the same hyperparameters as the previous experiments in model training, we scale up the similarity loss value by 100 to make it at the same scale as the original experiments. We evalu-

	Initialization		Fine-tune strategy			Evaluation task		
Experiment	ASR module	MT module	MT encoder	Adapter	#params	ASR	MT	ST
Cascaded 1	PT	PT	-	-	-	29.7	32.5	16.7
Cascaded 2	FT	PT	-	-	315M	22.3	-	18.4
Cascaded 3	PT	FT	-	-	610M	-	37.3	19.5
Cascaded 4	FT	FT	-	-	925M	-	-	**21.6**
Cascaded 5	FT	PT	Yes	-	152M	-	36.1	18.7
CoVoST2 Cascaded	-	-	-	-	-	21.4	29.0	18.3
E2E 1	PT	PT	-	-	-	-	-	0
E2E 2	FT	PT	-	-	-	-	-	0.5
E2E 3	PT	FT	-	-	-	-	-	0.1
E2E 4	FT	PT	Yes	-	152M	-	-	**22.8**
E2E 5	PT	PT	Yes	-	152M	-	-	0.4
E2E 6	FT	FT	Yes	-	152M	-	-	22.0
E2E 7	FT	PT	-	Yes	67M	-	-	20.9
CoVoST2 E2E	-	-	-	-	-	-	-	16.3

Table 1: Experiment results for model combination and fine-tuning strategy. Cascaded represents the cascaded combination, and E2E represents the end-to-end combination. PT means the pre-trained parameters, FT means the fine-tuned parameters that are from fine-tuning the entire model. Fine-tune strategy means the efficient strategy except fine-tuning the entire mode. For E2E, #params means the fine-tuned parameters in the second stage. We report WER score for ASR tasks and BLEU score for MT and ST tasks. The CoVoST2 results are from (Wang et al., 2020) that uses the exact same dataset as us.

ate model performance on different portions of the training data to evaluate the data efficiency.

In a first experiment, we evaluated the model on the zero-shot condition, where no end-to-end training data is available. As Table 2 shows, this approach fails to enable speech translation. We observe that the translation are all in English, although with the target forcing mechanism. Therefore we expect that involving a few end-to-end data would solve the issue.

Continue from the model trained with the similarity loss, we experiment on training with the original loss using different amounts of data. We observe that training with 10% training data enables the model to translate into the correct target language but poorly. In the case of fine-tuning with 20% data, adding similarity loss improves 51% compared with that without the loss. The evaluation score reaches 17.8 BLEU points, which achieves 85% performance of the best end-to-end model. Besides, compared with the learning curve of the original loss, adding the similarity loss enables the model to fulfil speech translation tasks with less training data. The advantages of adding the similarity loss demonstrate a promising approach to improve data efficiency. In addition, we observe that with all training data, training the model with the similarity loss gains 0.7 BLEU point improvement. Therefore, we conclude that involving the similarity loss increases data efficiency and benefits to improving model performance.

6 Conclusion

One major challenge of building speech translation systems is the computational and data requirements. In this work, we proposed using pre-trained speech recognition and text translation models to build a state-of-the-art speech translation system with limited resources. While a cascaded combination directly achieves relatively good performance, we develop several techniques to enable the end-to-end system to use these models.

We propose integrating both models into one single end-to-end speech translation model that can deal with independently pre-trained models and handle different word representations used in the pre-trained models. Secondly, we propose two training strategies that allow the training and inference on a single GPU. Finally, we present an additional training loss to reduce the need for end-to-end training data. Using all these techniques, the proposed model can outperform the cascaded model.

In future work, we will investigate how additional training signals can reduce the need for end-to-end training data even further, leading to the need for no end-to-end training data. Another promising direction is the use of a multi-lingual

Experiment	Without	10%	15%	20%	All
Only original loss	-	0.32	1.98	11.8	20.9
After similarity loss	0	0.98	10.7	17.8	21.6

Table 2: Experiment results on the similarity loss. For similarity loss, *without* means training only with the similarity loss; the following portion means, continue from the *without*, training with that portion of data with the original loss. The result is reported in the BLEU score.

system. In this case, end-to-end training data from another direction might be sufficient to translate speech in a new direction.

References

Ashkan Alinejad and Anoop Sarkar. 2020. Effectively pretraining a speech translation decoder with machine translation data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8014–8020.

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. *arXiv preprint arXiv:1802.06655*.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online. Association for Computational Linguistics.

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

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. *arXiv preprint arXiv:1909.08478*.

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 *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 2873–2887, Online. Association for Computational Linguistics.

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. IEEE.

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

Danlu Chen Jiatao Gu Changhan Wang, Anirudh Jain. 2019. Vizseq: A visual analysis toolkit for text generation tasks. In *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Marco Gaido, Mauro Cettolo, Matteo Negri, and Marco Turchi. 2021. Ctc-based compression for direct speech translation.

Marco Gaido, Mattia Antonino Di Gangi, Matteo Negri, and Marco Turchi. 2020. End-to-end speech-translation with knowledge distillation: Fbk@iwslt2020.

Mattia Antonino Di Gangi, Matteo Negri, and Marco Turchi. 2019. One-to-many multilingual end-to-end speech translation.

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*, ICML '06, page 369–376, New York, NY, USA. Association for Computing Machinery.

Hirofumi Inaguma, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. 2019. Multilingual end-to-end speech translation.

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. 2021. Lightweight adapter tuning for multilingual speech translation.

Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Multilingual speech translation with efficient finetuning of pretrained models.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Yuchen Liu, Hao Xiong, Zhongjun He, Jiajun Zhang, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-end speech translation with knowledge distillation. *arXiv preprint arXiv:1904.08075*.

Jan Niehues, Roldano Cattoni, Stüker Sebastian, Mauro Cettolo, Marco Turchi, and Marcello Federico. 2018. The iwslt 2018 evaluation campaign.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. pages 5206–5210.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Ngoc-Quan Pham, Jan Niehues, Thanh-Le Ha, and Alex Waibel. 2019. Improving zero-shot translation with language-independent constraints.

Juan Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. 2020. Self-training for end-to-end speech translation. *arXiv preprint arXiv:2006.02490*.

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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Mihaela C Stoian, Sameer Bansal, and Sharon Goldwater. 2020. Analyzing asr pretraining for low-resource speech-to-text translation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7909–7913. IEEE.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Changhan Wang, Anne Wu, and Juan Pino. 2020. Covost 2 and massively multilingual speech-to-text translation.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. 2019. Bridging the gap between pretraining and fine-tuning for end-to-end speech translation.

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