Zero-shot Speech Translation

Tu Anh Dinh
Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, The Netherlands

***Abstract*—Speech Translation (ST) is the task of translating speech in one language into text in another language. Traditional cascaded approaches for ST, using Automatic Speech Recognition (ASR) and Machine Translation (MT) systems, are prone to error propagation. End-to-end approaches use only one system to avoid propagating error, yet are difficult to employ due to data scarcity. We explore zero-shot translation, which enables translating a pair of languages that is unseen during training, thus avoid the use of end-to-end ST data. Zero-shot translation has been shown to work for multilingual machine translation, yet has not been studied for speech translation. We attempt to build zero-shot ST models that are trained only on ASR and MT tasks but can do ST task during inference. The challenge is that the representation of text and audio is significantly different, thus the models learn ASR and MT tasks in different ways, making it non-trivial to perform zero-shot. These models tend to output the wrong language when performing zero-shot ST. We tackle the issues by including additional training data and an auxiliary loss function that minimizes the text-audio difference. Our experiment results and analysis show that the methods are promising for zero-shot ST. Moreover, our methods are particularly useful in the few-shot settings where a limited amount of ST data is available, with improvements of up to +11.8 BLEU points compared to direct end-to-end ST models and +3.9 BLEU points compared to ST models fine-tuned from pre-trained ASR model.**

***Index Terms*—speech translation, zero-shot, few-shot**

I. Introduction

Speech Translation (ST) is the task of translating speech audio in a source language into text in a target language. Traditional approaches for ST include two cascaded steps: Automatic Speech Recognition (ASR) and Machine Translation (MT). ASR transcribes speech into text of the same language, and MT translates the text output by ASR to the text in the target language. These approaches are prone to errors propagated from the ASR step to the MT step [1]. Due to that, end-to-end Speech Translation has recently been gaining more interest. In end-to-end ST, the speech audio in a source language is translated directly into text in a target language and error propagation is no longer an issue. The challenge with end-to-end ST is the lack of appropriate end-to-end data, i.e, samples of speech in the source language and the corresponding text translation in the target language [2].

To tackle the data scarcity issue, Zero-shot Speech Translation will be explored in this paper. Zero-shot, in the context of translation models, is an approach that enables translating a pair of languages even when no end-to-end data of that particular pair was observed during training [2]. Zero-shot has been shown to work for multilingual machine translation, i.e., translating unseen pairs of languages in text format [3]. We attempt to build zero-shot models for speech translation: models that are trained only on ASR and MT tasks but can do ST task during inference. Here only one model is used to perform ST task during inference, which makes zero-shot ST an end-to-end approach. As an example, a model is trained with samples of English audio to English text (an ASR task) and samples of English text to German text (an MT task). Using zero-shot approaches, the model is expected to have the ability to translate English audio to German text (an ST task), which has not been seen in the training data. Thus, zero-shot ST avoids the use of end-to-end ST data for training.

We encounter several challenges when building zero-shot ST models. First, the zero-shot models using only ASR and MT training data output the wrong language while performing ST task (they output the source language instead of the target language). Second, the difference in representation of text and audio makes the models learn the ASR and MT tasks in different ways, which makes it non-trivial to perform zero-shot. We study and propose several approaches to tackle the issues: (1) removing the residual connections of a middle encoder layer to encourage language-independent representation of the data [4]; (2) using an auxiliary loss function to minimize the difference in text and audio representation [5]; (3) using augmented training data to force the models to learn to output the correct language and (4) using additional opposite training data also to force outputting the correct language. We find that the first approach, which was originally used for zero-shot multilingual MT, does not work for zero-shot ST. The other approaches prove to be promising, although they have not improved zero-shot ST to the point that it can be used practically. We also find these approaches to be particularly useful in the few-shot setting, when a limited amount of ST data is available, by comparing our few-shot models to some end-to-end baselines using a similar model architecture. Our few-shot models outperform direct end-to-end ST models, i.e., models trained on the same amount of ST data from scratch, by up to +11.8 BLEU points. Our models also outperform ST models fine-tuned from a pre-trained ASR model using the same amount of ST data by up to +3.9 BLEU points.

Our study addresses the following research questions. (1) How data-efficient are end-to-end and cascaded models? (2) Can techniques from zero-shot multilingual MT be applied to end-to-end ST? (3) How can we model the different modalities

This thesis was prepared in partial fulfilment of the requirements for the Degree of Bachelor of Science in Data Science and Artificial Intelligence, Maastricht University. Supervisor(s): Dr. Jan Niehues

in zero-shot ST?

The rest of this paper is structured as follows. Section II discusses the related work. Section III describes our approaches for zero-shot ST. Section IV explains our experiment setup; Section V discusses the experiment results. We further provide an in-depth analysis of our models in Section VI. Finally, Section VII draws conclusions and outlines future work.

II. Related work

In this section, we discuss existing work on Speech Translation (ST) as well as Zero-shot translation.

Different approaches for ST have been explored over the decades, as summarized in [2]. We highlight some related points in [2] as follows. Cascaded ST is the traditional approach, which was first introduced in [6]. Cascaded ST uses two separately-built systems: ASR and MT. Cascaded ST is a strong approach thanks to the well-established research and the abundance of data in ASR and MT. One shortcoming of cascaded ST is error propagation [1]. Therefore, more attempts have been made towards the end-to-end approach, which uses only one system to perform ST. Examples of end-to-end ST include direct ST models trained on end-to-end ST data from scratch [7], models pre-trained on an ASR task and fine-tuned on ST task [8], models trained on ST data generated by augmenting ASR or MT data [9, 10] and models co-trained on MT and ST task [11]. However, a major challenge of end-to-end ST is the lack of appropriate data. The available ST data are very limited in size and language coverage. For this reason, despite the efforts, many end-to-end ST approaches have not been able to outperform the traditional cascaded approach.

For zero-shot translation, it has been shown to work for multilingual MT. In [3], zero-shot multilingual MT is enabled by adding a language token to the beginning of the input sequence to indicate the required target language. The authors in [12] enable zero-shot multilingual MT by additionally concatenating language codes to every word. Several studies have been done to improve the quality of zero-shot multilingual MT. In [13], target language embeddings are used as a model feature instead of modifying the raw data to reduce vocabulary size. Multiple approaches to encourage a source-language-independent representation are proposed in [5], such as using a fixed size encoder for different languages. In [4], a language-independent representation is encouraged by disentangling positional information of the input and output tokens.

Inspired by the ability of zero-shot learning for multilingual MT, we study the applicability of similar approaches on Speech Translation, described in the following section.

III. Zero-shot Speech Translation

Fig. 1 illustrates the proposed approach for training a zero-shot ST model. The model is trained on two tasks simultaneously: Automatic Speech Recognition (ASR) and Machine Translation (MT). ASR training data include samples of audio in a source language and the corresponding text in the same language (*SRC audio* → *SRC text*). MT training data include samples of text in the source language and the corresponding text in a target language (*SRC text* → *TGT text*). Using zero-shot, the model is expected to be able to perform ST task during inference, i.e., translating *SRC audio* → *TGT text*. In order for zero-shot to work, it is necessary that the model represents *SRC audio* and *SRC text* in a similar way so that it can leverage the ASR and MT tasks learnt during training to perform the ST task during inference.

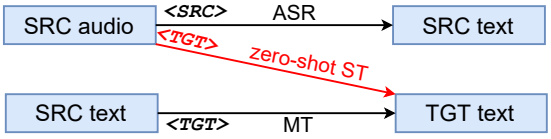

Fig. 1. Idea for zero-shot ST. Black arrows are training directions. Red arrow is zero-shot inference. Tags in the brackets are target-language tokens.

We use the Transformer architecture as described in [14] and [15], with the attention-based encoder and decoder. We extend it by including two parallel encoders, one for text input and one for audio input to fit our multi-modality training data. All layers of the text encoders are shared with the last layers of the audio encoder, similar to the ones in [11]. The overall structure is shown in Fig. 2. To enable zero-shot, we apply the same method as [5], which was originally used for zero-shot multilingual MT. We add target-language tokens to the beginning of input sequences and concatenate the target language embeddings to every decoder input to enforce the model outputting the language of interest. As can be seen in Fig. 1, with the same *SRC audio* input, if we want the model to output *SRC text*, we add the target-language token <*SRC*> to the input; if we want the model to output *TGT text*, we add the target-language token <*TGT*> to the input.

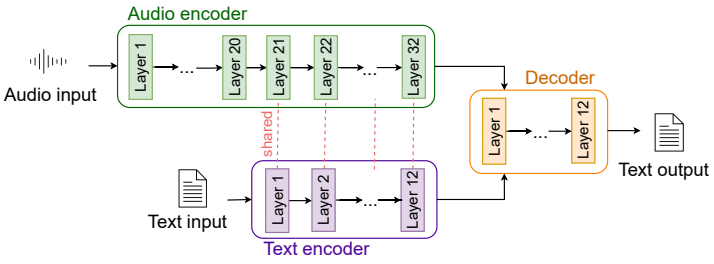

Fig. 2. Overall structure of multi-modality models.

We observe that this plain zero-shot model ignores the target-language tokens during training and is unable to do zero-shot ST (which is discussed further in section V-B). Hence, the rest of this section describes the approaches to further encourage the zero-shot ability of the model.

A. Disentangling Positional Information

The objective of a zero-shot ST model is to be able to switch between outputting *SRC* and *TGT* language based on the specified target-language token. Because of that, it is useful for the model to learn language-independent representation of the data. Therefore, we apply the Disentangling Positional Information approach introduced in [4]. This approach was

originally used to improve zero-shot multilingual MT. The idea is to relax the strong positional correspondence of the output to input tokens, hence give the model more freedom on word reordering to encourage a language-independent representation of the data. The authors in [4] achieved this by removing residual connections in a middle encoder layer.

As described earlier, our model has two parallel encoders, one for text and one for audio input. All layers of the text encoders are shared with the last layers of the audio encoder. Using the Disentangling Positional Information approach, we remove the residual connections in the middle layer of the text encoder, which is also shared with the audio encoder.

B. Auxiliary loss function

Another way to encourage zero-shot is to use an auxiliary loss function that minimizes the difference between encoder output of audio and text sentences. Using this auxiliary loss function, the model is expected to learn a modality-independent representation of the data, as shown in Fig. 3. If we have a sample sentence in text form and that same sentence in spoken (audio) form, the model should be able to represent the two samples similarly. In this way, it is expected that the model will be able to switch between outputting different languages during inference, hence encourage zero-shot. The metric chosen for the encoder output difference is the squared error of mean-pool over time, similar to the one suggested in [5]. That is, given a pair of aligned text and audio sentences, for each sentence's encoder output, we take the average over time of the encoder state (i.e., mean-pooling), then calculate the squared error between the two mean-pooled vectors.

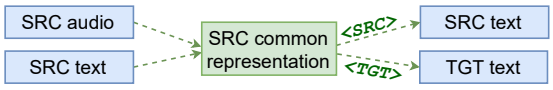

Fig. 3. Motivation for auxiliary loss. By minimizing text-audio difference, the model is expected to learn a modality-independent representation of the data (the green box). The tags in angle brackets are the target-language tokens.

C. Data augmentation

As described at the beginning of this section, we use target-language tokens to enforce the model outputting the language of interest. To be able to perform zero-shot, the model must learn the target-language tokens during training. It is observed that the plain model shown in Fig. 1 ignores the language tokens during training and unable to do zero-shot ST. For *SRC audio* input, even if we set the target token to be $<TGT>$, the model still outputs *SRC text*. One hypothesized reason is as follows. When training with only ASR and MT tasks as in Fig. 1, audio input always has *SRC* language output and text input always has *TGT* language output. Since the representation of text and audio input is different, the model ignores the target-language tokens and decides on the output language based on the modality instead, i.e., outputting *SRC text* for all audio input and outputting *TGT text* for all text input.

To tackle this issue, we create artificial data to be trained along with the main ASR and MT data, avoiding the need of searching for another real dataset with a real language. The amount of artificial data used is about half of the main training data. The idea is that input in each modality (text and audio) should have more than one target language output during training. In that way, the model will be forced to look at the target-language tokens to decide which language to output.

We create artificial training data by augmenting the ASR and MT data that we already have. We introduce an artificial language created from the *SRC* language as follows:

- Reverse source language sentences character-wise
- Lowercase the letters and strip away all punctuations
- Restore the common rules: capitalize the beginning of sentences, punctuation at the end of sentences

For example, "Hello world!" in English will be "Dlrow olleh!" in reversed English. We denote the artificial language as *SRC-R*. The set of training data is shown in Fig. 4. In addition to the main ASR and MT samples, we add *SRC audio* $\rightarrow$ *SRC-R text* and *SRC text* $\rightarrow$ *SRC-R text* samples. In this way, *SRC audio* and *SRC text* now have two target languages, which forces the model to learn the language tokens to know which language to output. We also include *SRC-R text* $\rightarrow$ *SRC text* and *SRC-R text* $\rightarrow$ *TGT text* samples so that the model can learn to switch between outputting *SRC* and *TGT* languages.

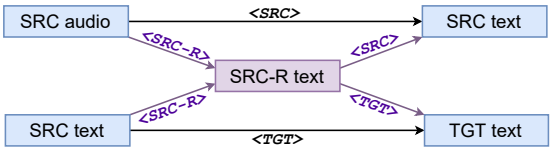

Fig. 4. Augmented training data. The black arrows are the main training directions. The purple arrows are the artificial directions.

D. Additional opposite training data

Recall that our main training data are ASR and MT data of *SRC* $\rightarrow$ *TGT* direction. In addition to the main training data, we include ASR and MT data of the opposite direction *TGT* $\rightarrow$ *SRC*. The motivation here is the same as in Section III-C: input in each modality should have more than one target language output during training. Hence, the training data are:

- *SRC audio* $\rightarrow$ *SRC text*
- *SRC text* $\rightarrow$ *TGT text*
- *TGT audio* $\rightarrow$ *TGT text*
- *TGT text* $\rightarrow$ *SRC text*

Here, each input modality (text and audio) has two target languages: *SRC* and *TGT*, which encourages the model to learn the target-language tokens. One disadvantage of this approach is that it requires the audio data of the *TGT* language.

E. Few-shot Speech Translation

As mentioned previously, the issue with end-to-end ST models is the lack of ST data for training. In addition to building zero-shot ST models, which require no ST data for training, we build few-shot models by fine-tuning the zero-shot models with a small amount of ST data to see how the models perform when a limited amount of ST data is available. It is expected that the fine-tuning process with ST data will help boosting the ST performance compared to zero-shot models.

IV. Experimental setup

The goal of the experiments is to build and evaluate zero-shot ST models to translate English audio to German text, denoted as *EN audio* $\rightarrow$ *DE text*.

A. Dataset and preprocessing

We use the CoVoST 2 dataset [16], a multilingual speech-to-text translation corpus including speech audio samples and the corresponding transcription and translation of different language pairs. The audio contains short spoken sentences with an average length of 5 seconds. The transcriptions and translations have an average length of 9 words per sentence.

The main data used in the experiments are English audio along with English transcription and German translation. The source language (*SRC*) is *EN*. The target language (*TGT*) is *DE*. This dataset contains 289K samples for training, 15K samples for validation and 15K samples for testing. We experiment with different portions of training data (10%, 25%, 33% and 100%). The validation set and test set are the same across all models, regardless of the amount of data used for training. Additionally, in some experiments, we also use German audio along with German transcription and English translation data, i.e, the *DE* $\rightarrow$ *EN* dataset. The amount of the *DE* $\rightarrow$ *EN* data used is always half of the amount of the *EN* $\rightarrow$ *DE* data.

For audio data, we extract and normalize 40-dimensional log scale mel filterbank concatenated with its delta coefficient to use as input features. For text data, we remove the double quotes at the beginning and the end of the sentences if any, and use SentencePiece [17] without pre-tokenization and pre-normalization to build subword-based vocabularies.

B. Model configurations

Our models use the Transformer architecture with attention-based encoder and decoder [14, 15]. For single-task models, we closely follow the hyperparameter choices in [18]. Models trained on a single audio-related task (ASR or ST) have 32 encoder layers, 12 decoder layers and death rate of 0.5. Models trained on a single MT task have 8 encoder layers, 8 decoder layers and no death rate. We adapt the hyperparameters to our mix-modality, multi-task models: 32 audio encoder layers, 12 text encoder layers, 12 decoder layers and no death rate; the 12 text encoder layers are shared with the last 12 audio encoder layers; shared weights for encoder and decoder word embeddings. All models use embedding size of 512, inner size of 2048, dropout rate of 0.2, attention dropout rate of 0.2, word dropout rate of 0.1, embedding dropout rate of 0.1, label smoothing rate of 0.1, Adam optimizer with learning rate of 0.01 and 8000 warm-up steps.

When training multiple datasets in one model (e.g., ASR and MT datasets), the batches from each dataset are ordered alternatively with a ratio such that each dataset is iterated through once in every epoch. Our models are trained for 64 epochs. One exception is the fine-tuned models, where we check the validation loss after every epoch and stop training as soon as the validation loss stops reducing. The checkpointed model with the lowest validation loss is used for final evaluation on the test set.

C. Evaluation metrics

For ASR tasks, the metric used is the Word Error Rate (WER) in percentage, calculated using VizSeq [19]. Before evaluation, the models' output transcriptions and the human-labeled transcriptions are lowercased, tokenized and the punctuation marks are removed, except for apostrophes and hyphens. The lower the WER, the better the performance.

For MT and ST tasks, the metric used is the BLEU score, calculated using sacreBLEU [20]. Before evaluation, the models' output is detokenized and the case is kept as it is to calculate case-sensitive BLEU score. The higher the BLEU score, the better the performance.

V. Results and Discussion

A. Cascaded approach versus end-to-end approach

Table I shows the results of training individual tasks with different amounts of the data as well as the performance of cascaded ST using ASR and MT models. With smaller amounts of data, ASR and MT tasks are easier to learn compared to ST task. With 25% of the training data, we achieve a reasonable score for ASR and MT (43.6% WER and 23.1 BLEU points, respectively), which is not the case for direct end-to-end ST (with 0.8 BLEU points). As a result, cascaded ST is much more data-efficient than direct end-to-end ST. Using 33% of the training data, we obtain 14.0 BLEU points by cascaded ST, which is only 0.9 points less than using all of the data to train a direct end-to-end ST model.

TABLE I
Performance of models trained on single tasks.

Data portion	ASR	MT	Cascaded ST	Direct end-to-end ST
25%	43.6	23.1	11.5	0.8
33%	37.5	26.3	14.0	1.6
100%	25.8	33.0	20.6	14.9

By this experiment, we confirm that zero-shot ST is an attractive approach. In zero-shot ST, we attempt to build a model that is trained on ASR and MT, which are the two data-efficient tasks, to perform ST during inference. Here only one model is trained and used to perform ST during inference, which makes zero-shot ST an end-to-end approach and avoids the error propagation issue of the cascaded approach. Since ASR and MT tasks have reasonable performance with only 25% of the training data, we use 25% of the data for the subsequent experiments and only train the most promising models on the full data.

B. Plain zero-shot Speech Translation models

The performance of plain zero-shot models is shown in Table II. We also include the performance of single-task models for comparison. Zero-shot models' MT performance is close to models trained on only MT task, with differences less than 0.4 BLEU points. For ASR, zero-shot models' performance is

slightly worse than models trained only on ASR, by at most 5% WER. Overall, for supervised tasks (ASR and MT), the plain zero-shot models have reasonable performance.

However, the zero-shot models are unable to perform zero-shot ST. We observe that all zero-shot predictions are in the wrong language. The models always output *EN text* for *EN audio* input, even when we set the target-language token to be <*DE*>. A possible reason is as follows. In the training data, *EN audio* input always has *EN text* output and *EN text* input always has *DE text* output. Since the representation of audio and text are different, the model ignores the target-language tokens and decides on the output language based on the modality of the input (audio or text), hence unable to perform zero-shot. One of our experiment results supports this hypothesis, which will be discussed in detail in Section V-E.

TABLE II
PLAIN ZERO-SHOT MODELS VERSUS SINGLE-TASK MODELS.

Data portion	Model type	ASR	MT	Zero-shot ST
25%	Single-task	43.6	23.1	–
25%	Plain zero-shot (ZS)	48.1	23.5	0.3
100%	Single-task	25.8	33.0	–
100%	Plain zero-shot (ZS)	28.4	32.8	0.6

We build few-shot ST models by fine-tuning the plain zero-shot model with 10% of ST data. We compare the few-shot models to some baselines: direct end-to-end models trained on 10% of ST data from scratch and ST models fine-tuned from ASR with 10% of ST data. We observe that direct end-to-end ST models have poor performance, at 0.5 BLEU points. On the other hand, the fine-tuned ASR models have better performance, hence we compare them to our few-shot models as shown in Table III. Our models fine-tuned from plain zero-shot outperform the models fine-tuned from ASR. When the amount of pre-training data (i.e., data that the models were trained on before fine-tuning) increases, the performance gap becomes more significant: from +0.3 to +1.4 BLEU points.

TABLE III
FEW-SHOT MODELS FINE-TUNED FROM PLAIN ZERO-SHOT AND ASR.

Pre-training data portion	Pre-trained model type	Fine-tuning data	ST score
25%	ASR	10% ST	3.7
25%	Plain zero-shot (ZS)	10% ST	4.0 **(+0.3)**
100%	ASR	10% ST	8.4
100%	Plain zero-shot (ZS)	10% ST	9.8 **(+1.4)**

Numbers in the brackets are comparison to the fine-tuned ASR models.

In the next experiments, we compare the performance of few-shot models using additional approaches to the few-shot models using the plain setting (instead of comparing them to the direct end-to-end ST and fine-tuned ASR models) to see the effect of the additional approaches.

C. Disentangling Positional Information

For the zero-shot model with residual connections in the middle layer of the shared encoder part removed, we observe that the model failed to learn the ASR task. The validation loss for the ASR task could not converge, and the WER score on the test set is 102.2% (i.e., the model output transcriptions are mostly wrong). However, for the MT task, the validation loss can converge, and the BLEU score on the test set is 13.7. This is worse compared to the plain zero-shot model (with 23.1 BLEU points on MT task), but it still proves that MT task can be learned with the middle residual connections removed. When performing zero-shot ST, the model still outputs the wrong language (*EN* instead of *DE*).

The depth of the audio encoder is believed to be the reason why the model cannot learn the ASR task when we apply this approach. As described in Section III-A, we remove the residual connections of the middle text encoder layer, which is shared with the audio encoder. In Fig. 5, for the audio encoder, in the backward pass, there are 25 layers below the residual-connection-removed layer. Therefore, these 25 layers are learnt less, which makes it difficult to learn the ASR task. For the text encoder, there are only 5 layers below the residual-connection-removed layer. Thus, the model can still learn the MT task, although the performance is worse than the usual setting. The difference between text and audio input is also a part of the problem. The idea of this approach was to relax the strong positional correspondence of the output to input tokens. Audio input consists of feature frames, while text input consists of subwords. Unlike the subwords, the feature frames are highly correlated to each other; e.g., some neighboring frames might belong to the same spoken phoneme. Therefore, the position of the frames is important, and it is difficult for the upper 6 layers of the audio encoder to capture the necessary positional information. In the original paper [4], this approach was applied on zero-shot multilingual MT (involving only text) with a smaller number of encoder layers, hence it worked on their setting but did not work in our case.

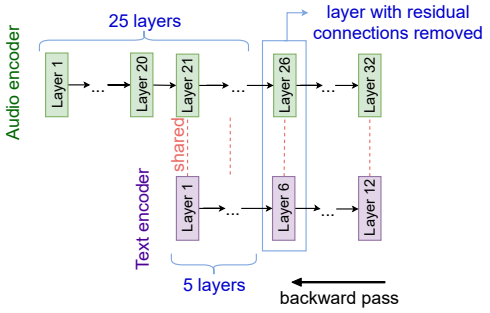

Fig. 5. Illustration of the Disentangling Positional Information approach.

D. Auxiliary loss function

Table IV shows the results of using an auxiliary loss to minimize the difference between text and audio encoder output in comparison with no auxiliary loss. As can be seen, with different weights of the auxiliary loss, the BLEU scores for zero-shot ST remain very low (all under 0.4 when training on 25% of the data and 0.65 when training on full data), meaning that the models are still unable to perform zero-shot ST. We

observe that all models in this experiment output the wrong language (*EN* instead of *DE*) when performing zero-shot ST.

TABLE IV
ZERO-SHOT MODELS WITH AUXILIARY LOSS.

Data portion	Aux. loss weight	ASR	MT	Zero-shot ST
25%	0.0	48.1	23.5	0.32
25%	0.1	45.7	22.7	0.38
25%	1.0	45.7	22.5	0.38
25%	5.0	43.9	21.9	0.39
100%	0.0	28.4	32.8	0.63
100%	5.0	26.9	32.5	0.65

Recall that our metric for encoder output difference is the squared error of mean-pool over time. We observe that the number of time steps of an *EN audio* sample is much higher than that of the aligned *EN text* sample, since audio input contains many audio frames while text input consists of subwords. When we calculate the squared error of mean-pool over time, i.e., averaging over the time steps, the difference in the number of time steps is not considered. This possibly explains why the representation of audio and text remains different, thus the model cannot perform zero-shot ST.

When training on 25% of the data, we observe that the models with higher auxiliary loss weight have better ASR performance. One possible reason is that, the auxiliary loss function helps the models to represent text and audio more similarly (although still not enough to perform zero-shot ST), hence *EN audio* and *EN text* become more similar, making it easier to transcribe *EN audio* $\rightarrow$ *EN text*. The model using auxiliary loss with weight 5 outperforms the plain model on the ASR task by -4.2% WER. When training on full data, the performance gap is no longer as large, at only -1.5% WER. The reason is that it is easier for the model to learn the ASR task with a large amount of data, hence the help of the auxiliary loss is no longer as significant.

Since we want the audio and text representation to be as similar as possible to encourage zero-shot, we use the weight of 5, which is the highest weight we have experimented with, whenever the auxiliary loss is used in the next experiments.

We build few-shot ST models by fine-tuning the zero-shot model using auxiliary loss with 10% of ST data. We compare this to the plain setting, as shown in Table V. Observe that the auxiliary loss helps improving the ST performance of the few-shot models: the BLEU score increased by +0.3 with 25% pre-training data and +0.8 with full pre-training data.

E. Data augmentation

We experiment with different sets of augmented training data as described in Fig. 6. The amount of artificial data is half of the amount of the main data. In setting (a), both *EN audio* and *EN text* input has two target languages. In setting (b), we have the same *EN-R text* input with *EN* and *DE* output so that the model can learn to switch between *EN* and *DE* targets. In setting (c), we include the artificial data of both (a) and (b).

The results of models trained on 25% of the data are shown in Table VI and Table VII. From Table VI, it can be

TABLE V
FEW-SHOT MODELS FROM SETTING WITH AUXILIARY LOSS.

Pre-training data portion	Pre-trained model type	Fine-tuning data	ST score
25%	ZS + auxiliary loss	10% ST	4.3 **(+0.3)**
100%	ZS + auxiliary loss	10% ST	10.6 **(+0.8)**

Numbers in the brackets are comparison to the corresponding plain setting.

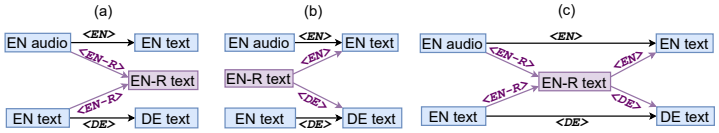

Fig. 6. Different settings for data augmentation. The black arrows are the main training directions. The purple arrows are the artificial directions.

seen that models with augmented training data are unable to perform zero-shot ST task. The BLEU scores are very low, where the highest is only at 0.45. However, we observe that the models output some tokens in the correct language (*DE*) when performing zero-shot ST, which was not the case in the plain setting. Therefore, we report the percentage of tokens in each language, shown in Table VII, to further examine the zero-shot ST output. Note that we do not take the *EN-R* language into account, since all models rarely output *EN-R* tokens while doing zero-shot ST. We report the percentages of tokens belonging to both *EN* and *DE*, tokens belonging to only *EN* and tokens belonging to only *DE*. Since the target language is *DE*, the more tokens output in *DE* the better. Looking at Table VII, the models with augmented training data output more *DE*-only tokens than the plain model. The model with setting (c) has significantly more *DE*-only tokens than the other, at 3.209%. Although this is far from the ground truth (53.485%), it still indicates that the models are starting to learn the target-language tokens using augmented training data. We use the most promising setting, i.e., setting (c), whenever data augmentation is used in the subsequent experiments.

TABLE VI
ZERO-SHOT MODELS TRAINED ON 25% DATA WITH AUGMENTATION.

Data portion	Model type	ASR	MT	Zero-shot ST
25%	Plain zero-shot (ZS)	48.1	23.5	0.32
25%	ZS + augmented data (a)	47.9	24.1	0.37
25%	ZS + augmented data (b)	51.6	22.7	0.34
25%	ZS + augmented data (c)	47.3	23.2	0.45

Another interesting observation from Table VII is that the percentages of tokens belonging to both *EN* and *DE* of the output zero-shot ST (all above 55%) are higher than that of the human-labeled translation (at 46.512%). The zero-shot models seem to output more of these tokens to commit to what they have observed a lot during training (*EN audio* $\rightarrow$ *EN text*) and satisfy the <*DE*> target-language token at the same time.

The result above (model being able to output some tokens

TABLE VII
PERCENTAGE OF OUTPUT TOKENS IN EACH LANGUAGE OF MODELS TRAINED ON 25% OF THE DATA.

Data portion	Model type	EN ∩ DE	EN	DE
25%	Plain zero-shot (ZS)	55.380	44.618	0.002
25%	ZS + augmented data (a)	55.095	44.899	0.004
25%	ZS + augmented data (b)	60.381	39.345	0.274
25%	ZS + augmented data (c)	**63.276**	**33.491**	**3.209**
25%	Human-labeled	46.512	0.000	53.485

ENR's statistics are excluded.

TABLE VIII
ZERO-SHOT MODELS TRAINED ON FULL DATA WITH AUGMENTATION.

Data portion	Model type	ASR	MT	Zero-shot ST
100%	Plain zero-shot (ZS)	28.4	32.8	0.63
100%	ZS + augmented data (c)	47.6	31.6	0.53
100%	ZS + augmented data (c) (fine-tuned from plain)	27.6	32.2	0.68

TABLE IX
PERCENTAGE OF OUTPUT TOKENS IN EACH LANGUAGE OF MODELS TRAINED ON FULL DATA.

Data portion	Model type	EN ∩ DE	EN	DE
100%	Plain zero-shot (ZS)	54.203	45.797	0.000
100%	ZS + augmented data (c)	63.432	35.024	1.538
100%	ZS + augmented data (c) (fine-tuned from plain)	55.700	44.251	0.038
100%	Human-labeled	45.558	0.000	54.441

ENR's statistics are excluded.

in correct language) supports our former hypothesis in Section V-B: plain zero-shot models output the wrong language due to the difference in text-audio representation. In the plain setting, the training samples are *EN audio* $\rightarrow$ *EN text* and *EN text* $\rightarrow$ *DE text*. We stated that, due to the difference between text and audio, the models decide on the output language based on the modality, i.e., output *EN* for all audio input and output *DE* for all text input. With data augmentation, both *EN text* and *EN audio* input have more than one target output language during training, hence the model is forced to learn the target-language token in order to decide which language to output.

We train the most promising setting, i.e., setting (c), on the full data. The results are shown in Table VIII and Table IX. Table VIII shows that this method does not scale well. When being trained on full data, the model with augmented data (c) has a significantly worse performance on ASR (with 47.6% WER) compared to the plain model (with 28.4% WER), while the performance on MT task stays approximately the same. This was not the case when the model was trained on 25% of the data. We suspect that, since there are 4 text-related tasks and only 2 audio-related tasks during training, the model focus more on the text-related tasks (MT tasks) than audio-related task (including ASR task). This did not happen when model was trained on 25% of the data since with 25% of the data, the ASR task cannot be learned any better, therefore the performance gap was not significant.

To overcome the shortcoming of the data-augmentation approach when training on full data, we fine-tune the plain-zero shot model with the augmented data instead of training on it from scratch. Since the plain model has good performance on ASR, we expect it to be a good starting point to achieve decent performance using augmented data in the end. This is indeed the case, as shown in Table VIII. The fine-tuned model has similar performance on supervised tasks compared to the plain zero-shot model. However, the fine-tuned model cannot output as many *DE* tokens as the model trained with augmented data from scratch, as shown in Table IX.

We then build few-shot ST models by fine-tuning the zero-shot model trained on augmented data with 10% of ST data. We compare this to the plain setting, as shown in Table X. The few-shot model using the data augmentation approach also does not scale well: the BLEU score decreases by 4.4 when the pre-training data are large, which is not the case when the pre-training data are small. This is expected, since the model trained on full augmented data from scratch has poor ASR performance, hence it is a worse baseline for fine-tuning. In contrast, the few-shot model using the augmented-data approach fine-tuned from the plain model has a significant improvement. The BLEU score for ST task increases by +1.7, which is more than twice larger than the gap of models pre-trained of 25% of the data, at +0.8 BLEU points.

TABLE X
FEW-SHOT MODELS FROM THE SETTING WITH AUGMENTED DATA (C).

Pre-training data portion	Pre-trained model type	Fine-tuning data	ST score
25%	ZS + augmented data	10% ST	4.8 **(+0.8)**
100%	ZS + augmented data	10% ST	5.4 (−4.4)
100%	ZS + augmented data (fine-tuned from plain)	10% ST	11.5 **(+1.7)**

Numbers in the brackets are comparison to the corresponding plain setting.

F. Additional opposite training data

In this experiment, we train the model with the main ASR and MT data of *EN* $\rightarrow$ *DE* direction and the additional ASR and MT data of the *DE* $\rightarrow$ *EN* direction. The amount of additional data is half of the amount of the main training data.

The results are shown in Table XI. The BLEU scores of zero-shot ST by the models trained on additional opposite data are still low (0.47 when trained on 25% of the data and 1.36 when trained on full data). We observe that the zero-shot ST output by the models is now mostly in the correct language - *DE*. However, in many sentences, it looks like the models are trying to recognize *DE text* from *EN audio*, instead of outputting the *DE* translation from *EN audio* as we hope.

We believe the text-audio difference is the reason why our models with additional opposite data have poor performance on zero-shot ST, although they are able to output the correct

language. Our models try to recognize *DE text* from *EN audio*, meaning that they are using the *DE* ASR task that was presented during training, instead of using what they have learnt from the MT tasks as well. Because of the text-audio difference, the models learn the ASR and MT tasks in very different ways, hence it is difficult to do zero-shot prediction using the knowledge from both the ASR and MT tasks.

TABLE XI
ZERO-SHOT MODELS WITH OPPOSITE DATA.

Data portion	Model type	ASR	MT	Zero-shot ST
25%	Plain zero-shot (ZS)	48.1	23.5	0.32
25%	ZS + opposite data	48.8	24.5	0.47
100%	Plain zero-shot (ZS)	28.4	32.8	0.63
100%	ZS + opposite data	26.8	32.6	1.36

We then build few-shot ST models by fine-tuning the zero-shot model using the additional opposite data approach with 10% of ST data. We compare this to the plain setting, as shown in Table XII. Observe that this approach helps improving the ST performance of the few-shot models: the BLEU score increased by +0.8 when the pre-training data portion is 25% and +0.5 when the pre-training data portion is full.

TABLE XII
FEW-SHOT MODELS FROM SETTING WITH OPPOSITE DATA.

Pre-training data portion	Pre-trained model type	Fine-tuning data	ST score
25%	ZS + opposite data	10% ST	4.8 **(+0.8)**
100%	ZS + opposite data	10% ST	10.3 **(+0.5)**

Numbers in the brackets are comparison to the corresponding plain setting.

G. Combination of approaches

In the previous experiments, we observe that using auxiliary loss, augmented data and additional opposite data are the most promising approaches. Hence, we experiment with the combinations of these approaches.

We first combine data augmentation with auxiliary loss. Section V-E shows that the data augmentation approach does not scale well and only gives decent performance when being fine-tuned from the plain model on full data. Hence, we combine this with the auxiliary loss approach by including the auxiliary loss when fine-tuning. We observe that the performance of the zero-shot model with the combined auxiliary loss is almost the same as without auxiliary loss. The score differences are at most 0.1 for all the ASR, MT and ST tasks. When building the corresponding few-shot model using 10% of ST data, we also observe no difference in the ST performance with and without auxiliary loss: the scores are both 11.5 BLEU points. For the detailed experiment results, see Appendix A.

Next, we combine the additional opposite training data and auxiliary loss approaches. Table XIII shows the experiment results of zero-shot models with this combination compared to the plain models. Observe that the zero-shot models with this combined approach have the same or better performance on the supervised tasks (ASR and MT). For zero-shot ST, this is the best result so far, at 0.72 BLEU points with 25% training data and 1.51 BLEU points with full training data, although more work still needs to be done until zero-shot ST can be put to practical use. This combined approach also improves the performance of few-shot models, as can be seen in Table XIV. The BLEU score increased by +1.7 with 25% pre-training data and +2.5 with full pre-training data.

TABLE XIII
ZERO-SHOT MODELS WITH OPPOSITE DATA AND AUXILIARY LOSS.

Data portion	Model type	ASR	MT	Zero-shot ST
25%	Plain zero-shot (ZS)	48.1	23.5	0.32
25%	ZS + opposite data + aux. loss	39.4	24.1	0.72
100%	Plain zero-shot (ZS)	28.4	32.8	0.63
100%	ZS + opposite data + aux. loss	25.4	32.8	1.51

TABLE XIV
FEW-SHOT MODELS USING OPPOSITE DATA AND AUXILIARY LOSS.

Pre-training data portion	Pre-trained model type	Fine-tuning data	ST score
25%	ZS + opposite data + aux. loss	10% ST	5.7 **(+1.7)**
100%	ZS + opposite data + aux. loss	10% ST	12.3 **(+2.5)**

Numbers in the brackets are comparison to the corresponding plain setting.

From this experiment, we see that auxiliary loss helps improving the ST performance when combined with the opposite data setting, but not with the data augmentation setting. We will further investigate this phenomenon in Section VI.

H. More on few-shot Speech Translation

In the previous experiments, we have seen that although our approaches do not enable zero-shot ST to the point where it can be used practically, they are useful to improve the ST performance when a limited amount of ST data is available. Considering each approach individually, data augmentation is the best approach, where the few-shot model's ST performance is 4.8 BLEU points with 25% pre-training data, and 11.5 BLEU points with full pre-training data (with full training data, we need to fine-tune from the plain model). The approach using opposite data has the same ST performance as using augmented data for few-shot models pre-trained on 25% of the data. However, unlike the opposite data approach, the data augmentation approach does not require any real data corpus in addition to the main ASR and MT data, hence data augmentation would be a better choice. Considering the approaches in combination, we see that adding opposite training data combined with auxiliary loss is the best, where the few-shot model's ST performance is 5.7 BLEU points with 25% pre-training data, and 12.3 BLEU points with full pre-training data. Comparing this best few-shot model to the direct end-to-end model trained on the same limited amount of ST data (with ST performance of 0.5 BLEU points), we see an

improvement of +5.2 BLEU points when the few-shot model is pre-trained on 25% of the data and +11.8 BLEU points when the few-shot model is pre-trained on full data. We also compare our best few-shot model to the more competitive baseline (i.e., ASR model fine-tuned with the same amount of ST data) and observe an improvement of +2.0 BLEU points when the models are pre-trained on 25% data and +3.9 BLEU points when the models are pre-trained on full data.

We also experiment with a higher amount of ST data for fine-tuning (25% instead of 10%). We observe that, with more ST data, the overall ST performance improves, but the gain from additional approaches compared to the plain setting is less significant. We also see that our best few-shot model using opposite data and auxiliary loss has 14.2 BLEU points using 25% of the ST data, which is only 0.7 points lower than that of the direct end-to-end ST model using full ST data (at 14.9). For the detailed experiment result, see Appendix B.

We observe that, when fine-tuning a zero-shot model with only ST data, the model forgets the tasks that it was previously trained on. The model can no longer perform ASR (it output *DE text*, while ASR requires *EN text* output), and the MT performance is significantly worse. This problem can be solved by fine-tuning the zero-shot model with small amounts of all ASR, MT and ST data. We observe this new few-shot model is a middle ground: it has reasonable performance for all three tasks, but the ASR and MT performance is slightly worse than the model before fine-tuning and the ST performance is slightly worse than the model fine-tuned with only ST data. See Appendix C for the detailed experiment result.

VI. Analysis

In this section, we further analyze our models from Section V. We showed that text-audio difference is the main challenge for zero-shot ST. Thus, we investigate the similarity between aligned text and audio mean-pooled encoder output using Singular Vector Canonical Correlation Analysis (SVCCA) [21]. Higher SVCCA scores mean more text-audio similarity, i.e., more modality-independent representation. We use the score of the plain setting as the baseline for comparison.

The analysis of models trained on 25% of the data is shown in Fig. 7. Looking at Fig. 7a, all of our approaches help increasing text-audio similarity, except adding opposite training data. Trivially, the models with auxiliary loss have higher SVCCA scores, since we designed our auxiliary loss to minimize the text-audio difference. For models with augmented training data, we have higher text-audio similarity possibly due to *EN text* and *EN audio* input has the same *ENR text* output during training. Text and audio are represented more similarly in few-shot models than in zero-shot models. This is expected, since few-shot models have seen both *EN audio* and *EN text* being translated to *DE text* during training, while zero-shot models have only seen *EN audio* and *EN text* being translated to different target language during training. Another observation is that the higher the SVCCA score of a zero-shot model, the higher the SVCCA score of the corresponding few-shot model. It suggests that our approaches have already introduced text-audio similarity in the zero-shot settings, but the effect is only clearly shown when we continue to fine-tune with ST data to build few-shot models.

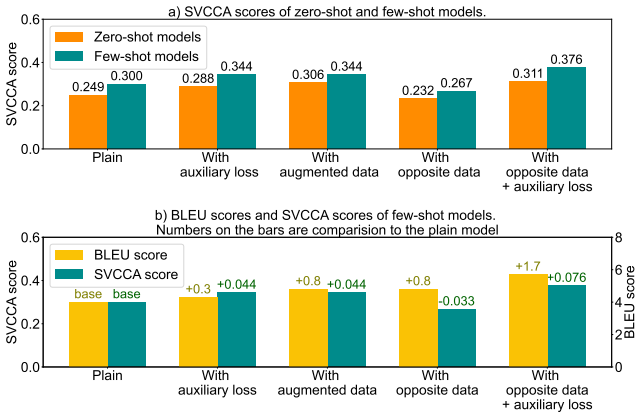

Fig. 7. Analysis of models pre-trained on 25% of the data.

We put the SVCCA scores and the BLEU scores on ST task of few-shot models together to get more insight, as shown in Fig. 7b. Observe that most of the scores agree with our hypothesis: more text-audio similarity means higher BLEU score. The only exception is the model with additional opposite data, which has a higher BLEU score but a lower SVCCA score compared to the plain model.

We further investigate the exceptional case: additional opposite data, compared to the plain setting. The *EN text* - *DE text* similarity is taken into account. The results are shown in Table XV. The SVCCA score of *EN text* - *DE text* for few-shot model in the opposite data setting is 0.607, which is significantly higher than the score of *EN audio* - *EN text* (at 0.267). We also calculate the SVCCA score of *EN text* - *DE text* in the plain setting, and obtain 0.407 points. This indicates that adding opposite data helps the model learn language-independent representation of the data instead of modality-independent representation, and that language-independent representation also helps improving ST performance. A possible reason is that with additional opposite data, we include *EN text* $\rightarrow$ *DE text* and *DE text* $\rightarrow$ *EN text* samples during training, hence increase the *EN text* - *DE text* similarity.

TABLE XV
Modality and Language Similarity: Opposite Data versus Plain.

	SVCCA EN audio – EN text	SVCCA EN text – DE text
Plain few-shot	**0.300**	0.407
Few-shot + opposite data	0.267	**0.607**

The analysis of models trained on full data is shown in Fig. 8. Overall, the SVCCA scores of models trained on full data are higher than those trained on 25% of the data. This is possibly due to the models learn the tasks in a less modality-specific way with a larger amount of training data. We observe similar patterns in the SVCCA scores compared to when the models were trained on 25% data: most approaches

increase the SVCCA score on text-audio similarity; few-shot models have higher SVCCA score than zero-shot models; the higher the SVCCA score of a zero-shot model, the higher the SVCCA score of the corresponding few-shot model; few-shot models with higher SVCCA score have higher BLEU score compared to the plain model. The opposite data approach remains exceptional, where the SVCCA score is approximately the same as the plain setting, but the BLEU score is improved. We also observe that the few-shot model with augmented data and auxiliary loss has the highest SVCCA score on text-audio similarity, yet its ST performance is worse than the few-shot model with opposite data and auxiliary loss. The reason is that models using opposite data can learn language-independent representation of the data, which also helps improving the BLEU score as discussed above. To confirm this, we calculate the SVCCA score of *EN text - DE text* for the two models. The few-shot model using opposite data and auxiliary loss obtains 0.539 points, which is higher than that of the few-shot model using augmented data and auxiliary loss, at 0.397.

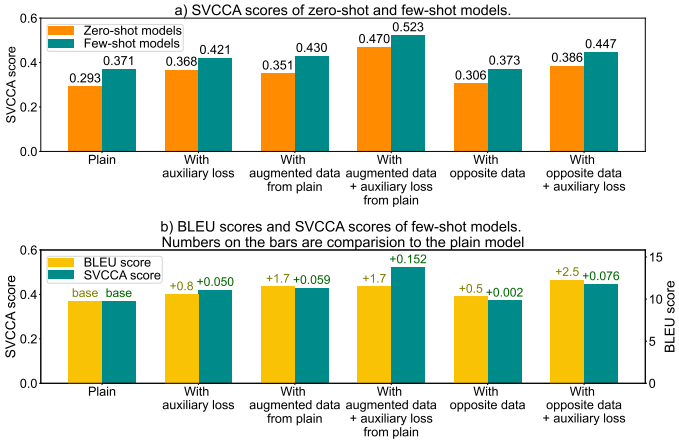

Fig. 8. Analysis of models pre-trained on full data.

The observation in this section also explains why adding auxiliary loss helps improving the performance when combined with additional opposite data, but not for augmented data as shown in Section V-G. Since the data augmentation approach already helps improving the text-audio similarity, adding auxiliary loss is not significantly useful. In contrast, the opposite data approach does not improve text-audio similarity, hence using auxiliary loss gives a significant improvement.

In addition to the SVCCA score, which measures text-audio similarity on a sentence level through their mean-pooled encoder output, we use another approach to measure text-audio similarity on a token level. We train a linear projection of the encoder output tokens to the modality labels (i.e., text and audio) [4]. Since the number of audio tokens is significantly higher than the number of text tokens, we consider the True Positive Rate (TPR, proportion of audio tokens identified correctly) and the True Negative Rate (TNR, proportion of text tokens identified correctly) of the classification output. Lower rates mean worse classification, meaning text and audio encoder output tokens are more similar. We observe that for all models using auxiliary loss, the TPR is over 99.9% and the TNR is under 10%. This means that the classifier has a poor performance where it predicts most of the tokens as audio, which indicates high similarity between text and audio encoder output tokens. On the other hand, for all the models that do not use auxiliary loss, both the TPR and TNR are over 99.9%, meaning that text and audio encoder output tokens are very distinguishable. Thus, we conclude that using auxiliary loss indeed increases text-audio similarity on a token level.

VII. Conclusions

The answers to the research questions in Section I are as follows. (1) Our experiment results confirm that cascaded ST is more data-efficient than direct end-to-end ST, since the two cascaded components (i.e., ASR and MT) are more data-efficient. There we emphasize the motivation for zero-shot ST: training an end-to-end model on two data-efficient tasks to perform ST task during inference. (2) We discover that not all approaches on zero-shot multilingual MT can be applied to zero-shot ST. An example is the Disentangling Positional Information approach, which did not work due to the model depth and the text-audio difference. (3) To model the different modalities, we use additional training data and auxiliary loss function, which, by in-depth analysis, prove to enhanced the text-audio similarity in both token and sentence levels.

We find that our approaches are promising for zero-shot ST. We observe that both language-independent and modality-independent representation of the data improve the ST performance. Our best zero-shot model with opposite training data and auxiliary loss obtains 1.51 BLEU points using no ST data. However, improving zero-shot ST to a practical point remains a difficult task. When a limited amount of ST data is available, our approaches show significant improvements compared to some baselines. We observe improvements of up to +11.8 BLEU points compared to the direct end-to-end ST approach and +3.9 BLEU points compared to models fine-tuned from the ASR task using the same amount of ST data.

For future work, we recommend continuing to enhance models' modality-independent representation of the data. Our approaches, although prove to have improved text-audio similarity, did not consider the difference in the number of time steps between audio and text. Therefore, we suggest investigating into making the time steps of text and audio more similar or the same. One direction is to use a fixed-size encoder as suggested in [5]. Since the number of time steps of audio input is a lot higher than text input, other directions could be further downsampling audio input, or use Connectionist Temporal Classification (CTC) to compress audio input [22]. Another approach to encourage text-audio similarity is to represent text input as spoken form using phoneme sequence [11] and use phone feature for audio input [23].

Acknowledgment

I am grateful to Dr. Jan Niehues for the guidance throughout the thesis. I also thank Danni Liu and my colleagues at Mediaan for their supports.

References

[1] N. Ruiz and M. Federico, "Assessing the impact of speech recognition errors on machine translation quality," in *11th Conference of the Association for Machine Translation in the Americas (AMTA), Vancouver, BC, Canada*, 2014.

[2] M. Sperber and M. Paulik, "Speech translation and the end-to-end promise: Taking stock of where we are," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp. 7409–7421. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.661

[3] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.

[4] D. Liu, J. Niehues, J. Cross, F. Guzmán, and X. Li, "Improving zero-shot translation by disentangling positional information," *arXiv preprint arXiv:2012.15127*, 2020.

[5] N.-Q. Pham, J. Niehues, T.-L. Ha, and A. Waibel, "Improving zero-shot translation with language-independent constraints," in *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 13–23.

[6] F. Stentiford and M. Steer, "Machine translation of speech." *British Telecom technology journal*, vol. 6, pp. 116–123, 04 1988.

[7] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Low-resource speech-to-text translation," *Annual Conference of the International Speech Communication Association (InterSpeech)*, 2018.

[8] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 58–68.

[9] Y. Jia, M. Johnson, W. Macherey, R. J. Weiss, Y. Cao, C.-C. Chiu, N. Ari, S. Laurenzo, and Y. Wu, "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)*. IEEE, 2019, pp. 7180–7184.

[10] J. Pino, L. Puzon, J. Gu, X. Ma, A. D. McCarthy, and D. Gopinath, "Harnessing indirect training data for end-to-end automatic speech translation: Tricks of the trade," *International Workshop on Spoken Language Translation (IWSLT)*, 2019.

[11] Y. Tang, J. Pino, C. Wang, X. Ma, and D. Genzel, "A general multi-task learning framework to leverage text data for speech to text tasks," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6209–6213.

[12] T.-L. Ha, J. Niehues, and A. Waibel, "Toward multilingual neural machine translation with universal encoder and decoder," *arXiv preprint arXiv:1611.04798*, 2016.

[13] T.-L. Ha, J. Niehues, and A. Waibel, "Effective strategies in zero-shot neural machine translation," *arXiv preprint arXiv:1711.07893*, 2017.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[15] N.-Q. Pham, T.-S. Nguyen, J. Niehues, M. Müller, S. Stüker, and A. Waibel, "Very deep self-attention networks for end-to-end speech recognition," *arXiv preprint arXiv:1904.13377*, 2019.

[16] C. Wang, A. Wu, and J. Pino, "Covost 2: A massively multilingual speech-to-text translation corpus," *arXiv preprint arXiv:2007.10310*, 2020.

[17] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71.

[18] N.-Q. Pham, T.-S. Nguyen, T.-L. Ha, J. Hussain, F. Schneider, J. Niehues, S. Stüker, and A. Waibel, "The iwslt 2019 kit speech translation system," in *Proceedings of the 16th International Workshop on Spoken Language Translation (IWSLT 2019)*, 2019.

[19] D. C. J. G. Changhan Wang, Anirudh Jain, "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*, 2019.

[20] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 186–191.

[21] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 6076–6085.

[22] M. Gaido, M. Cettolo, M. Negri, and M. Turchi, "Ctc-based compression for direct speech translation," *arXiv preprint arXiv:2102.01578*, 2021.

[23] E. Salesky and A. W. Black, "Phone features improve speech translation," *arXiv preprint arXiv:2005.13681*, 2020.

APPENDIX

A. Combination of data augmentation and auxiliary loss

TABLE XVI
ZERO-SHOT MODELS USING DATA AUGMENTATION IN COMBINATION WITH AUXILIARY LOSS.

Data portion	Model type	ASR	MT	Zero-shot ST
100%	Plain zero-shot (ZS)	28.4	32.8	0.63
100%	ZS + augmented data (fine-tuned from plain)	27.6	32.2	0.68
100%	ZS + augmented data + auxiliary loss (fine-tuned from plain)	27.7	32.3	0.67

TABLE XVII
FEW-SHOT MODELS FROM SETTING WITH DATA AUGMENTATION IN COMBINATION WITH AUXILIARY LOSS.

Pre-training data portion	Pre-trained model type	Fine-tuning data	ST score
100%	ZS + augmented data (fine-tuned from plain)	10% ST	11.5 **(+1.7)**
100%	ZS + augmented data + auxiliary loss (fine-tuned from plain)	10% ST	11.5 **(+1.7)**

Numbers in the brackets are comparison to the corresponding plain setting.

B. Few-shot models overall performance

TABLE XVIII
RESULT ON FINE-TUNING ZERO-SHOT MODELS USING ST DATA.

Pre-training data portion	Pre-trained model type	Fine-tuning data	ST score
100%	Plain zero-shot (ZS)	10% ST 25% ST	8.4 12.4
100%	ZS + auxiliary loss	10% ST 25% ST	10.6 **(+0.8)** 13.2 **(+0.8)**
100%	ZS + augmented data (fine-tuned from plain)	10% ST 25% ST	11.5 **(+1.7)** 13.5 **(+1.1)**
100%	ZS + opposite data	10% ST 25% ST	10.3 **(+0.5)** 13.2 **(+0.8)**
100%	ZS + augmented data + auxiliary loss (fine-tuned from plain)	10% ST 25% ST	11.5 **(+1.7)** 13.7 **(+1.3)**
100%	ZS + opposite data + auxiliary loss	10% ST 25% ST	12.3 **(+2.5)** 14.2 **(+1.8)**

Numbers in the brackets are comparison to the plain setting using the same amount of ST data.

C. Few-shot models fine-tuned on different data

TABLE XIX
FEW-SHOT MODELS USING ST DATA VERSUS USING ASR, MT, ST DATA.

Pre-training data portion	Pre-trained model type	Fine-tuning data	ASR	MT	ST
100%	Plain zero-shot	-	**28.4**	**32.8**	0.6
100%	Plain zero-shot	10% ST	98.2	28.3	**9.8**
100%	Plain zero-shot	10% ASR + 10% MT + 10% ST	29.5	31.4	9.1