Pretrained Speech Encoders and Efficient Fine-tuning Methods for Speech Translation: UPC at IWSLT 2022

Ioannis Tsiamas*, **Gerard I. Gállego***, **Carlos Escolano**,
José A. R. Fonollosa, **Marta R. Costa-jussà**
TALP Research Center, Universitat Politècnica de Catalunya, Barcelona
{ioannis.tsiamas,gerard.ion.gallego,carlos.escolano
jose.fonollosa,marta.ruiz}@upc.edu

Abstract

This paper describes the submissions of the UPC Machine Translation group to the IWSLT 2022 Offline Speech Translation and Speech-to-Speech Translation tracks. The offline task involves translating English speech to German, Japanese and Chinese text. Our Speech Translation systems are trained end-to-end and are based on large pretrained speech and text models. We use an efficient fine-tuning technique that trains only specific layers of our system, and explore the use of adapter modules for the non-trainable layers. We further investigate the suitability of different speech encoders (wav2vec 2.0, HuBERT) for our models and the impact of knowledge distillation from the Machine Translation model that we use for the decoder (mBART). For segmenting the IWSLT test sets we fine-tune a pretrained audio segmentation model and achieve improvements of 5 BLEU compared to the given segmentation. Our best single model uses HuBERT and parallel adapters and achieves 29.42 BLEU at English-German MuST-C tst-COMMON and 26.77 at IWSLT 2020 test. By ensembling many models, we further increase translation quality to 30.83 BLEU and 27.78 accordingly. Furthermore, our submission for English-Japanese achieves 15.85 and English-Chinese obtains 25.63 BLEU on the MuST-C tst-COMMON sets. Finally, we extend our system to perform English-German Speech-to-Speech Translation with a pretrained Text-to-Speech model.

1 Introduction

In the last few years, *end-to-end* (or *direct*) Speech Translation (ST) models have gained popularity in the research community. These systems differ from the classical *cascade* ones in their architecture, where instead of concatenating an Automatic Speech Recognition (ASR) model and a Machine Translation (MT) system, they directly translate speech into the target language without an intermediate transcription. This approach solves some limitations of cascade ST systems, like error propagation and slow inference times. But on the other hand, such approaches require more data to be competitive, which are not as abundant as ASR and MT data (Sperber and Paulik, 2020). However, the performance gap between the two approaches has become very small in the last years (Bentivogli et al., 2021), with end-to-end approaches having the best performances for the IWSLT 2020 test set in the last two evaluation campaigns (Ansari et al., 2020; Anastasopoulos et al., 2021).

Following this research trend, we participate in the Offline Speech Translation task of IWSLT 2022 (Anastasopoulos et al., 2022) with end-to-end systems that are built on top of our last year's submission (Gállego et al., 2021). The approach we follow is to leverage large pretrained speech and text models, in order to reduce the required amount of data usually needed to train competitive end-to-end ST systems (§2.1). As a speech encoder, we consider wav2vec 2.0 (Baevski et al., 2020) and HuBERT (Hsu et al., 2021), both already fine-tuned on English ASR data. As a text decoder, we use an mBART50 (Tang et al., 2020) fine-tuned on multilingual MT (one-to-many). These two modules are coupled with a *length adaptor* block, that reduces the length discrepancy. Although powerful, combining these modules results in a substantially large system, that is hard to train on normal hardware, given its computational and memory requirements. We thus follow a minimalistic fine-tuning strategy Li et al. (2021), which trains only specific modules in the network (§2.2). In addition, we extend this approach by adding *parallel adapters* (He et al., 2022) to the frozen layers (§2.3). We also explore the use of *knowledge distillation* (Hinton et al., 2015) from MT (Liu et al., 2019; Gaido et al., 2020) with mBART as the teacher (§2.4). Finally, we use SHAS (Tsiamas et al., 2022) to approximate

* Equal contribution

Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022), pages 265 - 276
May 26-27, 2022 ©2022 Association for Computational Linguistics

the optimal segmentation for the IWSLT test sets (§5).

In summary, our contributions with this work are: (1) We perform a comparison of wav2vec 2.0 and HuBERT for building an ST model. (2) We extend the fine-tuning strategy proposed by Li et al. (2021) with parallel adapters. (3) We study the effect of Knowledge Distillation for ST, in the context of pre-trained models.

2 Methodology

In this section, we describe the main parts of the proposed system 1, along with our approach for knowledge distillation and the Text-to-Speech model.

2.1 Pretrained modules

Our system is initialized with two pretrained models, an ASR encoder and an MT decoder. These two components were originally trained with self-supervised learning (SSL) strategies, and then fine-tuned with supervised learning on the ASR and MT tasks, respectively. Following, we describe these models, and we give details on how we couple them to build an ST system.

Speech Encoders We experiment with two different pretrained speech encoders: wav2vec 2.0 (Baevski et al., 2020) and HuBERT (Hsu et al., 2021). Thanks to the SSL pretraining, these models can achieve very competitive results with only a few labelled data points. Both speech encoders are based on the same architecture. The first block consists of a stack of seven 1D convolutional layers, which extract features from the raw waveform input. Next, a Transformer encoder (Vaswani et al., 2017) further processes these features, and extracts contextualized representations. The main difference between these two speech encoders lies on the pretraining strategy they follow. On the one hand, wav2vec 2.0 is pretrained to identify the true speech representation from a masked time step, by solving a contrastive task on quantized representations. On the other hand, HuBERT predicts the masked time steps by computing the loss against pseudo-labels, which are obtained from an iterative offline clustering.

Text Decoder We use the decoder of mBART to initialize the decoder of our system (Liu et al., 2020). Similarly to the speech encoders, mBART is also pretrained with SSL and then fine-tuned for a downstream task. It follows the same strategy used to pretrain BART (Lewis et al., 2020), but in this case, the model is trained with multilingual data. Concretely, it is trained as a denoising autoencoder, with the objective of reconstructing the original text input, which has been intentionally corrupted. After pre-training, mBART can be fine-tuned with supervised data on the (multilingual) MT task.

Length Adaptor To build our system, we combine two components that were designed for different modalities. Hence, there is a length discrepancy between the actual encoder representations and the ones expected by the decoder. To reduce this gap, we introduce a simple module to shorten the sequence length of the encoder outputs (Li et al., 2021). The length adaptor is a stack of convolutional layers that reduces the sequence length by 8, thus achieving a better coupling of the two main blocks.

2.2 LNA Fine-tuning

The LayerNorm and Attention (LNA) fine-tuning strategy consists of just training some specific layers in an ST system initialized by pretrained speech and text models. By avoiding a full fine-tuning, it is feasible to train the combination of these massive pretrained components in a time and memory efficient way. Specifically, we use the version of this strategy that fine-tunes the layer normalization, the encoder self-attention and the decoder cross-attention layers. LNA fine-tuning approaches the results of a full fine-tuning, while training just the 20% of the total parameters (Li et al., 2021).

2.3 Parallel Adapters

Although LNA fine-tuning has been shown to yield very competitive results, it almost entirely neglects the feed-forward blocks in the Transformer, where lie most of the parameters of every layer. Recent studies have unveiled the contribution of these blocks in promoting concepts in the vocabulary space (Geva et al., 2022). Hence, totally freezing them could hinder the performance of the system in a new domain. Instead of fine-tuning the parameters of a layer, another popular approach is to use adapters (Houlsby et al., 2019; Le et al., 2021) to approximate its output. An adapter module is a feed-forward network with a bottleneck dimension and ReLU activation. In this research, we use adapters to compliment the LNA fine-tuning technique (§2.2) by adding adapters to the (frozen) feed-

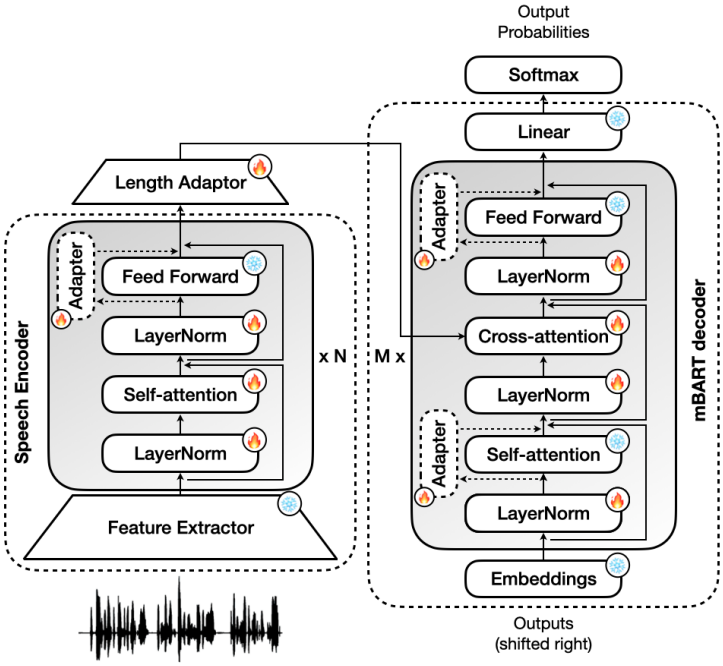

Figure 1: System overview. Fire indicates that a block is fine-tuned, and snowflake that it is frozen.

forward layers of the transformer layers. We also add them to the (frozen) decoder self-attention layers, since the number of extra parameters are negligible. Following He et al. (2022), we used adapters with a scaled parallel insertion form, which was found to provide higher performance gains than with a sequential insertion.

2.4 Knowledge Distillation

Apart from efficient fine-tuning methods, we experimented with using knowledge distillation (KD) (Hinton et al., 2015), which has been successfully applied for training an end-to-end ST model (student) (Liu et al., 2019; Gaido et al., 2020), by transferring knowledge from a pretrained MT model (teacher). The effectiveness of KD stems from the fact that the MT task is less complex than the ST task, and thus the student can benefit from learning the teacher distribution. In this work, we are using word-level KD, where the output probabilities of the MT model act as soft labels for the ST model. The loss is a weighted sum of the standard Cross Entropy and the Kullback-Leibler (KL) divergence between the student and teacher output distributions. The importance of each term in the loss is controlled by a hyperparameter $\lambda \in (0, 1)$. Since we are initializing the decoder of our models with the mBART decoder, we are also using it as the teacher for KD. Following (Gaido et al., 2020), we extract the top-k output probabilities with mBART offline and thus there is no additional computational impact during training with KD, while it also does not affect negatively the learning process (Tan et al., 2019; Gaido et al., 2020) Due to extracting only the top-k logits from the teacher, the teacher distribution tends to be sharper than normal, and thus we used a temperature $T > 1$, to soften it.

3 Data

3.1 Datasets

To train our models we used data from three speech translation datasets, MuST-C v2 (Di Gangi et al., 2019), Europarl-ST (Iranzo-Sánchez et al., 2020) and CoVoST-2 (Wang et al., 2020). More specifically, we used the English-German (en-de), English-Japanese (en-ja) and English-Chinese (en-zh) from MuST-C and CoVoST, and the en-de from Europarl-ST. MuST-C is based on TED talks, Europarl-ST on the European Parliament proceedings, and CoVoST is derived from the Common Voice (Ardila et al., 2020) corpus. Since only MuST-C has in-domain data, we used the dev and tst-COMMON splits for development and testing, while from Europarl-ST and CoVoST, we used their respective dev and test splits as additional training data. Furthermore, the IWSLT test sets of 2019 and 2020 (Niehues et al., 2019; Ansari et al., 2020), which do not have ground truth segmentations, serve as development data for en-de. Finally,

we submit our predictions for the IWSLT test set of 2021 (en-de) (Anastasopoulos et al., 2021) and the test sets of 2022 (en-de, en-ja, en-zh) (Anastasopoulos et al., 2022).

Dataset	en-de	en-ja	en-zh
MuST-C v2	436	526	545
Europarl-ST †	83	-	-
CoVoST 2 †	413	413	413
Total	942	939	958

Table 1: Training data measured in hours. †: train, dev and test splits are considered.

3.2 Data Filtering

We removed examples with duration longer than 25 seconds to avoid memory issues. To ensure that our training data are of high quality, we applied two stages of filtering by either modifying the transcriptions and translations (text filtering) or to completely removing an example (speech filtering).

Text filtering. We applied this filtering in both the transcription and translation of each example, and the process is different for each dataset. For MuST-C we removed the speaker names, that are in-audible and usually appear at the beginning of the sentences when multiple speakers are active in a talk. We also removed events like "Laughter" and "Applause" that are not expected to be generated by our ST systems during evaluation. For Europarl-ST we converted the number format to match the one in MuST-C, by using commas as the thousands-separator in large numbers instead of spaces. No specific text filtering is applied on the CoVoST data. Finally, to minimize the differences between the datasets, we applied punctuation and spacing normalization with Sacremoses[1].

Speech filtering. To identify and remove noisy examples, that would potentially hurt the performance of our models, we applied speech filtering on all source audios in our training data. We performed ASR inference with a pretrained wav2vec 2.0[2] using the Transformers library (Wolf et al., 2020), and removed the examples that had a word error rate (WER) higher than 0.75. WER was calculated after removing punctuation and multiple spaces, lower-casing the ground-truth transcriptions and converting numbers from digits to their spelled-out words format. The average WER per dataset was 0.141 for MuST-C, 0.175 and 0.152 for CoVoST, and the speech-filtering process resulted in removing 1.5% of MuST-C, 1% of Europarl-ST and 2% of CoVoST.

3.3 Data Augmentation

To enrich and diversify our data, we perform audio augmentation. This process is done on-the-fly during training using WavAugment (Kharitonov et al., 2021). Each training example has a probability of 0.8 to be augmented, in which case the *tempo* and *echo* effects are applied. Modifying the tempo of an audio allows our ST models to adapt to speeches of different speeds, while the echo effect simulates the echoing that is present in large rooms, where usually TED talks take place. The tempo augmentation parameter is sampled uniformly in the range of (0.85, 1.3), while the echo-delay and echo-decay parameters, which control the echo augmentation, are sampled from the ranges of (20, 200) and (0.05, 0.2) respectively.

4 Experiments

Here we describe the experiments we carried out in this work with their implementation details.

4.1 Experimental Setup

LNA-wav2vec. We build on top of our submission to IWSLT 2021 (Gállego et al., 2021), where we combined a wav2vec 2.0 encoder, with an mBART decoder, and the whole system is trained with the LNA technique. This year, we reproduce this experiment, with two main differences. First, we perform a hyperparameter tuning for the learning rate and use the entire CoVoST dataset (out-of-domain) instead of sub-sampling it.

LNA-HuBERT. In the next experiment, we explore the effect that different speech encoders bring in our system. Thus, we initialize the speech encoder of our ST model, with HuBERT.

LNA-Adapters. Last year, we found it to be beneficial, to use an adapter, at the output of the speech encoder. We expand this idea, and perform an experiment where we instead of using a single adapter, we use scaled parallel adapters in all frozen sub-layers of our system. These are the feed-forward layers of both the encoder and decoder, as well as

[1] https://github.com/alvations/sacremoses

[2] https://huggingface.co/facebook/wav2vec2-large-960h-lv60-self

the self-attention layers in the decoder, that are not part of the LNA fine-tuning.

KD. For the next experiment, we use knowledge distillation from mBART, where the loss of the ST model during training is a weighted sum of the standard cross entropy and the KL divergence between the MT and ST output distributions. We also explored the trade-off between the two loss functions, by tuning the λ parameter that controls it.

Apart from the aforementioned experiments, we apply checkpoint averaging, where we average around the best checkpoint of an experiment (**ckpt AVG**). Furthermore, we continue fine-tuning for few more epochs on only the in-domain data of MuST-C, while also using smaller data augmentation probability (**in-domain FT**). Finally, since the aforementioned experiments have core differences, we hypothesize that they are diverse enough to benefit from ensembling. We experiment with ensemble decoding from various combinations of our best models (**Ensemble**).

4.2 Implementation Details

All our models use the same architectures for the encoder and the decoder. The encoder is either initialized with wav2vec 2.0[3] or HuBERT[4] and are composed of a 7-layer convolutional feature extractor and 24-layer Transformer encoder. Both were pretrained with 60k hours of untranscribed speech from Libri-Light (Kahn et al., 2020), and fine-tuned for ASR with 960 hours of labeled data from Librispeech (Panayotov et al., 2015). The wav2vec 2.0 version we use was also fine-tuned with pseudo-labels (Xu et al., 2020). The decoder is initialized from mBART[5] that has been fine-tuned for multilingual MT, including English to German, Japanese and Chinese. Its decoder is a 12-layer Transformer. The feature extractor of the encoder has 512 channels, kernel sizes of (10, 3, 3, 3, 3, 2, 2) and strides of (5, 2, 2, 2, 2, 2, 2). Each layer in the Transformer encoder and decoder has a dimensionality of 1024, feed-forward dimension of 4096, 16 heads, ReLU activations, and use pre-layer normalization. The length adaptor between the encoder and decoder is a 3-layer convolutional network with 1024 channels, stride of 2 and uses GLU activations. The embedding layer and the linear projection weights of the decoder are shared, and has a size of 250,000. For the experiment with adapters, we are using scaled parallel adapters with a dimensionality of 512 and a scaling factor of 4 (He et al., 2022).

The inputs to the model are waveforms of 16kHz sampling rate, which are normalized to zero mean and unit variance. During training, each source audio is augmented (before normalization) with a probability of 0.8. We train bilingual models on all data of Table 1, with maximum source length of 400,000 and target length of 1024 tokens. We use gradient accumulation and data parallelism to achieve a batch size of approximately 32 million tokens. We use Adam (Kingma and Ba, 2014) with $\beta_1 = 0.99$, $\beta_2 = 0.98$ and base learning rate of $2.5 \cdot 10^{-4}$, which we found in preliminary experiments to be better, compared to the learning rate of 10^{-4} that we used last year (Gállego et al., 2021). The learning rate is controlled by a tri-stage scheduler with phases of 0.15, 0.15 and 0.7 for warm-up, hold and decay accordingly, while the initial and final learning rate has a scale of 0.01 compared to base. Sentence averaging and gradient clipping of 20 are used. We applied dropout of 0.1 before every non-frozen layer, and use time masking for spans of length 10 with probability of 0.2 and channel masking for spans of length 20 with probability of 0.1 in the output of the encoder feature extractor.

The loss is the cross-entropy with label smoothing of 0.2. For the experiments that additionally use KD, the loss is a weighted sum of the standard cross-entropy (no label smoothing) and the KL divergence between the teacher and student distributions, controlled by a hyperparameter λ, which we tune in (0, 1). The teacher distribution for each step is extracted offline with mBART[6] using the Transformers library. We keep the top-8 indices, and both the teacher and student distributions are additionally modified with temperature $T = 1.3$ (Gaido et al., 2020).

For in-domain fine-tuning, we train only on data from MuST-C, and lower the probability of augmentation to 0.2. We train for an additional 4

[3] https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec2_vox_960h_new.pt

[4] https://dl.fbaipublicfiles.com/hubert/hubert_large_ll60k_finetune_ls960.pt

[5] https://dl.fbaipublicfiles.com/fairseq/models/mbart50/mbart50.ft.1n.tar.gz

[6] https://huggingface.co/facebook/mbart-large-50-one-to-many-mmt

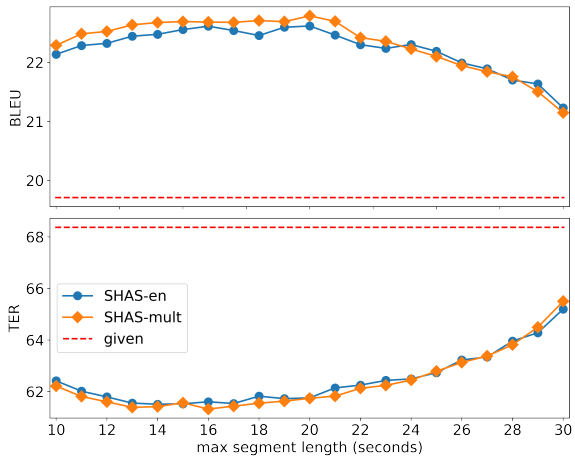

Figure 2: BLEU(↑) and TER(↓) in IWSLT test 2019 for different parameters of max-segment-length for the English and multilingual SHAS methods. With dashed lines are the results for the given segmentation.

epochs with a learning rate of 10^{-5}. The learning rate is increased from $5 \cdot 10^{-7}$ for the first 15% of the training and then decays for the rest of the training.

After training, we pick the best checkpoint according to the BLEU (Papineni et al., 2002) on the development set of MuST-C and average 5 checkpoints around it. For generation, we use a beam search of 5. We used one of our base experiments (LNA-HuBERT) with learning rate of 10^{-4}), to fine-tune SHAS on the 2019 IWLST test set (Niehues et al., 2019) and then use the best configuration to segment the test sets of 2020, 2021 and 2022 (Ansari et al., 2020; Anastasopoulos et al., 2021, 2022). We choose our best model based on the BLEU of the 2019 test set and report results on MuST-C tst-COMMON and the IWSLT test set of 2020. For choosing the best segmentation (§5), apart from BLEU, we additionally evaluate with TER (Snover et al., 2006). Our models are implemented in fairseq (Ott et al., 2019) and are trained using NVIDIA apex[7] and 16 floating point precision. The code for our experiments is available in a public repository[8].

5 Audio Segmentation

Although our training data contain ground truth segmentations derived from strong punctuation of the transcriptions, the IWSLT test sets, are unsegmented and thus require an intermediate step of audio segmentation, before applying our ST models. Past evaluation campaigns of IWSLT have shown light to the importance of accurate audio segmentation for end-to-end ST, where top-performing participants used their own segmentation algorithms to get large improvements in translation quality. For our submission, we are using SHAS, a segmentation method that can effectively learn the manual segmentation from a labelled speech corpus (Tsiamas et al., 2022). It relies on a segmentation frame classifier and a probabilistic Divide-and-Conquer (pDAC) algorithm to obtain the segmentation for a given audio. The frame classifier is a Transformer encoder with a binary classification layer, that predicts the splitting frames in the audio using as inputs contextual representations extracted with a frozen XLS-R (Babu et al., 2021). The pDAC segmentation algorithm is based on the method of (Potapczyk and Przybysz, 2020) and progressively splits on the frames of the lowest probability, until all resulting segments are shorter than a prespecified max-segment-length parameter. Segmentations created with SHAS approach the translation quality of the manual segmentation on the en-de tst-COMMON set of MuST-C v2.0, retaining 95% of the manual BLEU.

We used the public implementation of SHAS[9] and tested two available pretrained models for the frame classifiers, one trained on English source audio from MuST-C v2 and a multilingual which is additionally trained on Spanish, French, Portuguese, and Italian data from mTEDx (Salesky et al., 2021). We obtain the frame probabilities for the audios of the 2019 IWSLT test set (Niehues et al., 2019) with the English and multilingual classifiers, and then used the pDAC algorithm with a varying max-segment-length to segment them. To find the best parameters, we maximize the translation quality of the segmentation by the following process: (1) Translate the resulting segments with our ST model, (2) align the translations with the references using the mwerSegmenter tool (Matusov et al., 2005) and (3) compute the BLEU and TER scores.

In figure 2 we observe that values of max-segment-length in the range of 14 and 20 seconds for pDAC, result in the best segmentation, with BLEU scores of 22.5 and TER scores of 61.5. Additionally, in that range, SHAS with a multilingual classifier performs better than the English

[7] https://github.com/NVIDIA/apex
[8] https://github.com/mt-upc/iwslt-2022
[9] https://github.com/mt-upc/SHAS

one, with small improvements of approximately 0.2 BLEU. The highest BLEU score overall is obtained with the multilingual classifier and at max-segment-length of 20 seconds, but given that there is an increase in the TER score, we decided to continue with max-segment-length of 16 seconds, which seems to have more consistent results. Thus, for our final results (§6) for the test sets of 2019 and 2020, as well as for our submissions for 2021 and 2022, we used SHAS with the multilingual classifier and a max-segment-length of 16 seconds (SHAS-mult-16). Due to the absence of available test sets to fine-tune SHAS for the Japanese and Chinese, we also use SHAS-mult-16 to segment the en-ja and en-zh IWSLT 2022 test sets.

6 Results

In this section, we analyze the results of our experiments. We base our experimentation on the en-de language pair, to compare the results with our last year's submission (Gállego et al., 2021; Anastasopoulos et al., 2021). Hence, first we analyze the results for this language pair (Table 2) and then present the results for en-ja and en-zh (Table 3).

6.1 English-German

In our main results for en-de (Table 2), we also include our last year's submission (row 0). In (1), we repeat the same experiment, with the main differences being an increase of the learning rate to $2.5 \cdot 10^{-4}$, no sub-sampling of the CoVoST data, and using SHAS for the segmentation of the IWLST data at inference. These changes are already providing us an increase of 2.3 BLEU in MuST-C and 3 BLEU at IWSLT tst2019. In (2), we substitute the wav2vec 2.0 encoder for a HuBERT encoder, which brings further improvements of 0.6 to 0.8 BLEU in all test sets. With the addition of adapters (3a), we observe improvements in the IWSLT test sets but a drop in MuST-C. We hypothesize that complimenting LNA with adapters (§2.3) results in overfitting in MuST-C, but at the same time, the additional parameters provide an extra flexibility to the model regarding data from different segmentation (IWSLT test sets). With checkpoint averaging (3b), we get improvements in all test sets, providing the overall best results from a single model. Next, we apply knowledge distillation (4a), which initially results in a slight drop for the IWSLT test sets and in an increase in MuST-C (as compared to 3a). We believe that, since knowledge distillation from MT (§2.4) uses manually segmented data (MuST-C), those are the data that could benefit from it (§6.3). With in-domain fine-tuning and checkpoint averaging (4b, 4c), we get small improvements of 0.2 BLEU in all test sets. By ensembling our two best models (5a), we get improvements in all test sets. Finally, since our models are diverse enough (speech encoder, adapters, knowledge distillation), we ensemble all four of them (5c) and obtain our best results, with 30.83 BLEU on MuST-C tst-COMMON, and 25.39, 27.78 on the 2019 and 2020 test IWSLT test sets. The segmentation algorithm also plays a key role in the performance of our models, with improvements of 4 to 5.5 BLEU in all experiments, as compared to the given one.

6.2 English-Japanese & English-Chinese

From the results of en-ja and en-zh (Table 3), we observe that similarly to en-de, the addition of adapters brings a slight drop in performance for MuST-C. Still, we hypothesize that this would turn into an increase for the unsegmented IWSLT test sets, although we cannot confirm it since there are no data available from previous editions. Moreover, we noticed that MT with mBART performed worse than our ST model (11.63 BLEU for en-ja and 19.51 BLEU for en-zh on dev), meaning that knowledge distillation would most likely cause a drop in performance. Therefore, we do not perform KD for those languages. Finally, we ensemble the two models (after checkpoint averaging), with which we obtain on tst-COMMON 15.85 BLEU for en-ja and 25.63 BLEU for en-zh.

6.3 Analysis on Knowledge Distillation

We carry out an analysis on knowledge distillation, to better understand its impact to our system (Table 2, row 4). Specifically, we analyze the trade-off between the standard cross entropy and the teacher-student KL divergence, by varying the lambda in [0.25, 0.5, 0.75, 1]. In figure 3 we provide the BLEU scores for the dev and tst-COMMON sets of MuST-C and the IWSLT test sets of 2019 and 2020, which are segmented with SHAS-mult-16. We also provide the results for an experiment that does not use KD, but instead of the standard cross entropy, it was trained with the label-smoothed one. We also provide the performance of the MT teacher (dashed line) on the dev set of MuST-C, which can be seen as an upper bound for the student. Firstly, we observe that relying completely on the teacher

		Dataset	**MuST-C**		**IWSLT**			
		split	dev	tst-COMMON	tst2019		tst2020	
		segmentation			given	SHAS	given	SHAS
0		LNA-wav2vec (Gállego et al., 2021)	26.76	26.23	17.25	20.06	-	-
1		LNA-wav2vec	29.08	28.50	18.37	23.03	19.61	25.33
2		LNA-HuBERT	28.97	29.27	19.02	23.72	20.09	25.61
3	a	LNA-Adapters-HuBERT	28.92	28.53	19.51	24.07	20.66	26.35
	b	↪ ckpt AVG	29.41	**29.42**	**20.48**	**24.88**	**21.19**	**26.77**
4	a	LNA-Adapters-HuBERT-KD	**29.44**	28.79	19.37	23.74	20.25	26.10
	b	↪ in-domain FT	29.43	28.97	19.52	23.87	20.67	26.17
	c	↪ ckpt AVG	29.42	28.87	19.71	23.92	20.93	26.32
5	a	Ensemble (3b, 4c)	30.07	30.33	20.51	24.98	21.85	27.38
	b	Ensemble (3b, 4c, 2)	30.33	30.44	**20.69**	25.34	22.30	27.61
	c	Ensemble (3b, 4c, 2, 1)	**30.53**	**30.83**	20.65	**25.39**	**22.40**	**27.78**

Table 2: BLEU scores for en-de MuST-C and IWSLT sets. In bold are the best scores by single models, and in underlined bold are the best scores overall. LNA-wav2vec (Gállego et al., 2021) uses a different segmentation algorithm and results are not available for tst2020.

Language Pair	**en-ja**		**en-zh**	
split	dev	test	dev	test
LNA-HuBERT	**12.45**	15.20	**22.55**	24.84
↪ ckpt AVG (a)	12.32	15.36	22.28	**24.95**
LNA-Adapters-HuBERT	12.26	14.89	22.29	24.48
↪ ckpt AVG (b)	12.07	**15.46**	22.07	24.85
Ensemble (a, b)	**12.45**	**15.85**	**22.98**	**25.63**

Table 3: BLEU scores on dev and test (tst-COMMON) sets of MuST-C v2 for en-ja and en-zh. In bold are the best scores by single models, and in underlined bold are the best scores overall.

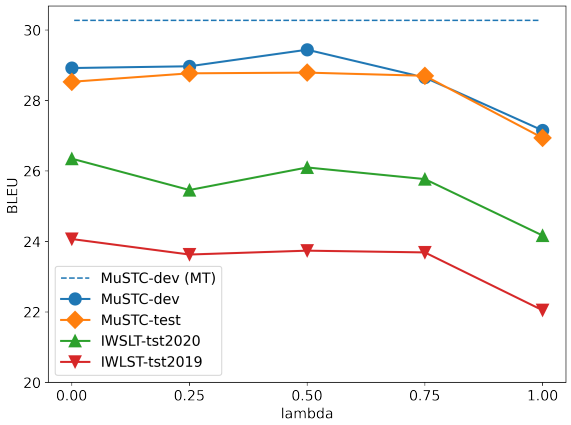

Figure 3: BLEU scores for knowledge distillation with varying lambda for en-de. IWSLT test sets are segmented with SHAS-mult-16.

degrades the translation quality in all sets. This is contrary to previous research suggesting that $\lambda = 1$ is optimal (Liu et al., 2019). This conflicting results likely stems from the small differences between our ST and MT models, which in dev set of MuST-C is approximately 1.5 BLEU, while in (Liu et al., 2019) the gap is more than 10 BLEU. Secondly, we observe that there is an increase in BLEU when the ST model is trained with a mixture of the two losses for MuST-C ($\lambda = 0.5$), but there is a drop for the IWSLT test sets. We believe that these differences are a consequence of the training-testing segmentation mismatch, where the MuST-C sets have the same segmentation as the training data, while for IWSLT sets, this segmentation is only approximated with SHAS. This difference is expected to make it harder for the ST model to utilize the MT knowledge from the ground truth segmentations.

6.4 Submission Results

In Table 4 we present our results on the official test sets of IWSLT 2022 (Anastasopoulos et al., 2022). All test sets were segmented with SHAS (§5), and the models used are the best ensembles for each language (Tables 2, 3). For the en-de test set of 2021 (Anastasopoulos et al., 2021), we obtain a BLEU of 24.5 (ref-1)[10]. This result, compared to the ones of IWSLT 2021 (Anastasopoulos et al., 2021), stands 2.7 BLEU above our submission (Gállego et al., 2021), 1.9 BLEU above the best end-to-end submission (Bahar et al., 2021) and only 0.1 BLEU

[10]IWSLT systems were ranked with this reference in 2021.

IWSLT test set	BLEU		
	ref-1	ref-2	both
en-de 2021	24.5	20.9	34.8
en-de 2022	23.0	20.8	32.3
en-ja 2022	15.1	15.6	24.7
en-zh 2022	29.2	29.9	36.4

Table 4: Official submission results for en-de (2021, 2022) and en-ja, en-zh (2022). BLEU is measured for two different references and for both together. Different models are used for each language. For en-de we used Ensemble of Table 2 - row 5c and for en-ja and en-zh the Ensembles of Table 3.

below the best overall[11]. For the test sets of 2022 we obtain 23 BLEU for en-de, 15.1 BLEU for en-ja and 29.2 BLEU for en-zh. The reader can refer to Anastasopoulos et al. (2022) for a comparison with the other submitted systems.

7 Speech-to-Speech

We have also submitted our system to the Speech-to-Speech (S2S) translation task[12], by building a cascade system. This is composed of the main end-to-end Speech-to-Text translation model and a Text-to-Speech (TTS) system. We used a pretrained[13] VITS model (Kim et al., 2021) for synthesizing the German speech. It is based on normalizing flows (Rezende and Mohamed, 2015), adversarial training and a stochastic duration predictor. It is capable of generating speech in different pitches and rhythms, resulting in more natural sounding audio utterances.

8 Conclusions

We described the submission of the UPC Machine Translation group for the IWSLT 2022 Offline ST and Speech-to-Speech tasks. Our system is end-to-end and leverages ASR and MT pretrained models to initialize the encoder and decoder. Due to the large size of the system, we employed efficient fine-tuning methods that train only specific layers and provide evidence that the addition of parallel adapters to the non-trainable layers can bring further improvements. We showed that a HuBERT encoder is more suitable than wav2vec 2.0 for our system and brings improvements in all test sets. We also explored the use of knowledge distillation, which provided only minor improvements to the test sets with ground-truth segmentations, most likely because the MT model was borderline better than our ST model. Additionally, we show that the SHAS method provides high-quality segmentations of the IWSLT test sets, bringing improvements up to 5 BLEU compared to the given segmentation. Our best single model, uses a HuBERT encoder and LNA with parallel adapters, and achieved 29.42 BLEU on MuST-C tst-COMMON set, and 24.88 and 26.77 BLEU on IWSLT 2019 and IWSLT 2020 test sets. We ensembled 4 different systems for our final submission, which further increased the BLEU in the aforementioned sets by 1 to 1.5 points. We also described our submissions for the English-Japanese and English-Chinese pairs that scored 15.85 and 25.63 MuST-C tst-COMMON. Finally, we also submitted a Speech-to-Speech system, by using a pretrained German TTS model to the generated translations.

For future work, we are planning to explore more pretrained speech encoders and text decoders, and dive deeper into the ways that we can optimally combine them and efficiently fine-tune for end-to-end ST. We will also investigate how to gain the most from an MT teacher, in such scenarios where there is a small gap between the MT and the ST models.

[11] Cascade system by HW-TSC, no paper available

[12] Results not available at time of submission, the reader can refer to Anastasopoulos et al. (2022)

[13] https://github.com/jmp84/vits

Acknowledgements

This work was supported by the project ADAVOICE, PID2019-107579RB-I00 / AEI / 10.13039/501100011033

References

Antonios Anastasopoulos, Luisa Bentivogli, Marcely Z. Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Marcello Federico, Christian Federmann, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel M. Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, Juan Pino, Elizabeth Salesky, Jiatong Shi, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alex Waibel, Changhan Wang, and Shinji Watanabe. 2022. FINDINGS OF THE IWSLT 2022 EVALUATION CAMPAIGN. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, Dublin, Ireland. Association for Computational Linguistics.

Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alex Waibel, Changhan Wang, and Matthew Wiesner. 2021. Findings of the IWSLT 2021 Evaluation Campaign. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, Online.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondrej 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 H. Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation, IWSLT 2020, Online, July 9 - 10, 2020*, pages 1–34. Association for Computational Linguistics.

R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, et al. 2021. XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale. *arXiv preprint arXiv:2111.09296*.

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

Parnia Bahar, Patrick Wilken, Mattia A. Di Gangi, and Evgeny Matusov. 2021. Without further ado: Direct and simultaneous speech translation by AppTek in 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 52–63, Bangkok, Thailand (online). Association for Computational Linguistics.

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.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. 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)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Marco Gaido, Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2020. End-to-end speech-translation with knowledge distillation: FBK@IWSLT2020. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 80–88, Online. Association for Computational Linguistics.

Gerard I. Gállego, Ioannis Tsiamas, Carlos Escolano, José A. R. Fonollosa, and Marta R. Costa-jussà. 2021. End-to-end speech translation with pre-trained models and adapters: UPC at IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 110–119, Bangkok, Thailand (online). Association for Computational Linguistics.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

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

J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux.

2020. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673. https://github.com/facebookresearch/libri-light.

Eugene Kharitonov, Morgane Rivière, Gabriel Synnaeve, Lior Wolf, Pierre-Emmanuel Mazaré, Matthijs Douze, and Emmanuel Dupoux. 2021. Data augmenting contrastive learning of speech representations in the time domain. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 215–222.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5530–5540. PMLR.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2021. Lightweight adapter tuning for multilingual speech translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 817–824, Online. Association for Computational Linguistics.

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

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

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. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. 2005. Evaluating Machine Translation Output with Automatic Sentence Segmentation. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.

J. Niehues, R. Cattoni, S. Stüker, M. Negri, M. Turchi, Elizabeth Salesky, Ramon Sanabria, Loïc Barrault, Lucia Specia, and Marcello Federico. 2019. The iwslt 2019 evaluation campaign. In *Proceedings of the 16th International Workshop on Spoken Language Translation*.

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

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

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Tomasz Potapczyk and Pawel Przybysz. 2020. SR-POL's System for the IWSLT 2020 End-to-End Speech Translation Task. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 89–94, Online. Association for Computational Linguistics.

Danilo Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

Elizabeth Salesky, Matthew Wiesner, Jacob Bremerman, Roldano Cattoni, Matteo Negri, Marco Turchi, Douglas W. Oard, and Matt Post. 2021. Multilingual tedx corpus for speech recognition and translation. In *Proceedings of Interspeech*.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Matthias Sperber and Matthias Paulik. 2020. 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*, pages 7409–7421, Online. Association for Computational Linguistics.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

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. *arXiv preprint arXiv:2008.00401*.

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2022. Shas: Approaching optimal segmentation for end-to-end speech translation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Changhan Wang, Anne Wu, and Juan Pino. 2020. Covost 2: A massively multilingual speech-to-text translation corpus. *arXiv preprint arXiv:2007.10310*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. 2020. Self-training and pre-training are complementary for speech recognition. *arXiv preprint arXiv:2010.11430*.