Enhanced Direct Speech-to-Speech Translation Using Self-supervised Pre-training and Data Augmentation

Sravya Popuri[⋆]*, Peng-Jen Chen*[⋆]*, Changhan Wang, Juan Pino,*
Yossi Adi, Jiatao Gu, Wei-Ning Hsu[†]*, Ann Lee*[†]

Meta AI

{spopuri,pipibjc}@fb.com

Abstract

Direct speech-to-speech translation (S2ST) models suffer from data scarcity issues as there exists little parallel S2ST data, compared to the amount of data available for conventional cascaded systems that consist of automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) synthesis. In this work, we explore self-supervised pre-training with unlabeled speech data and data augmentation to tackle this issue. We take advantage of a recently proposed speech-to-unit translation (S2UT) framework that encodes target speech into discrete representations, and transfer pre-training and efficient partial finetuning techniques that work well for speech-to-text translation (S2T) to the S2UT domain by studying both speech encoder and discrete unit decoder pre-training. Our experiments show that self-supervised pre-training consistently improves model performance compared with multitask learning with a BLEU gain of 4.3-12.0 under various data setups, and it can be further combined with data augmentation techniques that apply MT to create weakly supervised training data.[1]

Index Terms: speech-to-speech translation, self-supervised pre-training, data augmentation

1. Introduction

Direct speech-to-speech translation (S2ST) aims at translating speech from one language into speech in another language without relying on text generation as an intermediate step [1–6]. Compared to conventional cascaded approaches [7, 8], which take advantage of automatic speech recognition (ASR), machine translation (MT) or end-to-end speech-to-text translation (S2T) followed by text-to-speech synthesis (TTS), direct S2ST has the advantage of faster inference [4] and can support translation between languages without text writing systems [2–4, 6].

Most recently, [4] proposes to apply a self-supervised speech encoder pre-trained on unlabeled speech to convert target speech into discrete units [9] and build a speech-to-unit translation (S2UT) model for direct S2ST. Self-supervised discrete targets can disentangle linguistic content from speaker identity and prosodic information in speech [10]. Moreover, they enable opportunities for applying techniques from speech-to-text model training, such as ASR and S2T, to direct S2ST.

As we move along the spectrum from multi-stage to direct approaches, the amount of parallel data available becomes much more scarce. Pre-training, including initializing encoder or decoder trained from ASR or MT tasks, or from self-supervised pre-training with unlabeled data [11–14], multitask learning [11, 15] and data augmentation [13, 16–18] have been extensively studied for tackling the data scarcity issue for S2T. For direct S2ST, [1, 4] show that multitask learning is crucial for model convergence, and [1, 19] focus on incorporating pre-trained modules from ASR, S2T, MT or TTS tasks.

In this work, we take advantage of the S2UT framework proposed in [4] and show that large-scale self-supervised pre-training with monolingual speech and text data and data augmentation techniques that benefit S2T model training can also be applied on S2UT model training. For pre-training, we transfer the technique from [14], which performs efficient finetuning with a wav2vec 2.0 speech encoder [20] and an mBART text decoder [21], to S2UT with a wav2vec 2.0 speech encoder and an mBART decoder trained with discrete units extracted from unlabeled speech data. For data augmentation, we utilize ASR, MT and TTS models to create weakly supervised data [16].

The contributions of this work are as follows: we empirically demonstrate that self-supervised encoder and decoder pre-training with unlabeled speech and partial finetuning improve S2ST training under various setups, including training with synthetic single-speaker or real multi-speaker target speech, and low-resource setup (30-hr). We further improve the model via data augmentation with weakly supervised data and conduct experiments with the combination of multiple datasets to achieve strong performance over ASR+MT+TTS baseline systems.

2. Related Work

Self-supervised speech encoder pre-training has led to huge performance improvement on a wide range of applications [22] such as ASR [20, 23], S2T [18], speaker identification [24], etc. SpeechT5 [25] proposes end-to-end pre-training, where the model learns to reconstruct the log Mel-filterbank of the masked regions of the input speech, to improve performance of speech-to-speech tasks such as voice conversion and speech enhancement, while it was not evaluated in the context of direct S2ST.

The improved quality of speech units discretized based on self-supervised speech representations also allows researchers to apply natural language processing (NLP) techniques on speech, such as spoken generative language modeling [9, 26], and emotion conversion cast as a unit-to-unit translation task [27]. [27] further applies text-based autoencoder denoising on units for model pre-training. In this work, we model the target speech as discrete units, extend the monolingual unit pre-training in [27] to a multilingual setup, and perform speech encoder and discrete unit decoder pre-training separately.

Data augmentation is a common method for increasing the size of training data. Synthetic speech data from TTS [16, 28],

[⋆] Equal contribution. [†] Equal supervision.

[1] Audio samples are available at: https://facebookresearch.github.io/speech_translation/enhanced_direct_s2st_units/index.html. Code and pre-trained models are available at: https://github.com/pytorch/fairseq/blob/main/examples/speech_to_speech/docs/enhanced_direct_s2st_discrete_units.md

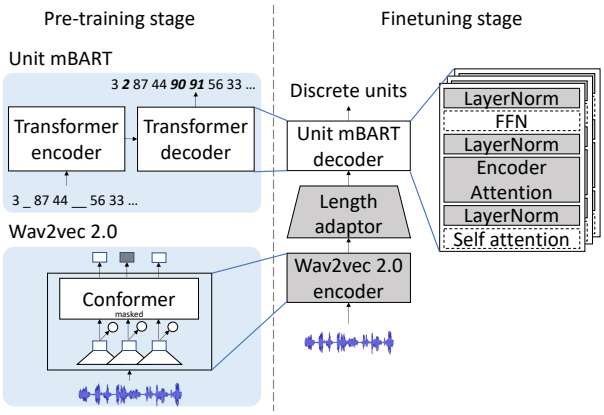

Figure 1: *Flowchart for the speech encoder and decoder pre-training and finetuning process. The blocks in shade illustrate the modules that are finetuned in the* ***LNA-D*** *finetuning strategy (Sec. 3.3), which we find is the most effective.*

self-training [18, 29] or back-translation [30] from unlabeled data have been shown to be useful to ASR or S2T. In this work, we apply MT and TTS to prepare weakly supervised S2ST data from speech in the source language.

3. System

3.1. Speech-to-unit translation (S2UT) model

We follow [4] and encode the target speech as discrete units with a HuBERT model trained on unlabeled speech followed by a k-means model learned from its hidden representations [23]. Waveform input is encoded into a sequence of k-means cluster indices at every 20-ms frame, and we remove consecutive duplicate units to create a *reduced* unit sequence representing the target speech. In the end, the direct S2ST system consists of a sequence-to-sequence S2UT model with a speech encoder and a unit decoder, followed by a unit HiFi-GAN vocoder [10] trained separately for unit-to-waveform conversion. In this work, we explore both encoder and decoder pre-training (Fig. 1).

3.2. Model pre-training

3.2.1. Encoder pre-training: wav2vec 2.0

Wav2vec 2.0 [20] is a self-supervised framework to learn speech representations from unlabeled audio data. It uses a multi-layer convolution neural network to encode the audio followed by a Transformer-based [31] context encoder to build the contextualized representations. The model is trained via contrastive loss with masked spans on the input to the context encoder. In this work, we use Conformer [32] instead of the Transformer [31] for better model performance (Sec. 4.5.2).

3.2.2. Decoder pre-training: unit mBART

mBART [21] was originally proposed for denoising autoencoder over text sequences. During training, the sequence-to-sequence model predicts the original text x given its noisy version, $g(x)$, created by randomly masking spans of x. The starting position of each span is uniformly sampled from all positions, and the span lengths are sampled from a Poisson distribution (λ, or $span\_size$). The masking process is repeated until the total accumulated span lengths take up $p\%$ (or $mask\_ratio$) of the input sequence. The model is trained on text data from multiple languages with language tags. In our case, we treat the *reduced* discrete units extracted from unlabeled speech data as text and apply mBART training. A standard Transformer-based encoder-decoder architecture [31] as in MT is used.

3.3. Model finetuning

We combine the wav2vec 2.0 encoder and the mBART decoder and study the finetuning strategies in [14]. A randomly initialized adaptor layer consisting of a single 1-D convolutional layer with stride 2 is added between the pre-trained modules to increase the model's capacity to alleviate the mismatch between the learned representations, as well as length difference between the source audio and the *reduced* target units.

We examine both full and partial finetuning of the model. For the latter, we focus on the **L**ayer**N**orm and **A**ttention modules (dubbed as "LNA") proposed in [14]. The hypothesis is that LayerNorm parameters reflect the statistics of the pre-training data, and the encoder attention in the unit mBART decoder is optimized for unit sequence input. The adaptor layer is fully finetuned. We explore four finetuning strategies in total: 1. **LNA-E**: The LayerNorm and self attention parameters in the encoder and all the parameters in the decoder are finetuned. 2. **LNA-D**: The whole encoder and the LayerNorm and encoder attention in the decoder are finetuned. We optionally freeze the encoder for the first k updates. 3. **LNA-E,D**: Only LNA parameters are finetuned both on the encoder and the decoder side. 4. **Full**: We finetune the whole model end-to-end with an option of freezing the encoder for the first k updates.

3.4. Data augmentation

We take advantage of speech from ASR data in the source language to increase the size of the parallel S2ST training data. We use a Transformer MT model [31] to translate the text transcription in the source language to text in the target language. To convert text in the target language into target speech, we apply a text-to-unit (T2U) model [6][2], which is a Transformer-based sequence-to-sequence model trained on text and the corresponding discrete unit sequence extracted from the paired audio. The T2U model is a way to bypass the TTS generation and HuBERT unit extraction pipeline for efficient generation of large-scale weakly supervised data. We choose to distill knowledge from MT models instead of pursuing self-training, since a three-stage cascaded system (ASR+MT+TTS) can take advantage of a large amount of data from each component during training and still outperforms existing direct S2ST systems [4].

4. Experiments

We conduct experiments on Spanish-English (Es-En) and English-Spanish (En-Es) translation with FAIRSEQ [33, 34].

4.1. Data

Table 1 summarizes the statistics of all the datasets used in the experiments. We experiment with two types of parallel **S2ST data**. First, we follow the convention of applying single-speaker TTS[3] on the target text of **S2T data** [1, 3–5] (dubbed as "*S2ST-syn*"). We combine S2T datasets from multiple domains to improve the robustness of model training [51, 52], resulting in 196-hr training data for Es-En and 571-hr for En-Es. We also combine the dev sets from all domains for tracking the training process and checkpoint selection, and conduct evaluation on

[2] https://github.com/pytorch/fairseq/blob/main/examples/speech_to_speech/docs/textless_s2st_real_data.md

[3] En: https://huggingface.co/facebook/tts_transformer-en-ljspeech, Es: https://huggingface.co/facebook/tts_transformer-es-css10

Table 1: *Statistics of datasets (train/dev/test splits). TTS is applied on the target text of S2T for creating synthetic S2ST data.*

	# samples	source (hrs)	target (hrs)
S2T, Es-En			
CoVoST-2 [35]	78.9k / 13.3k / 13.2k	112 / 22.0 / 22.7	81.0 / 14.4 / -
Europarl-ST [36]	7.4k / 1.9k / 1.8k	20.6 / 5.4 / 5.1	21.8 / 5.6 / -
mTEDx [37]	35.6k / 888 / 989	63.4 / 1.5 / 1.7	58.6 / 1.4 / -
S2T, En-Es			
Europarl-ST [36]	31.6k / 1.3k / 1.3k	75.6 / 3.0 / 2.9	76.5 / 3.0 / -
MuST-C [38]	260k / 1.3k / 2.4k	495 / 2.5 / 4.1	481 / 2.5 / -
S2ST, Es-En			
VoxPopuli [6, 39]	159k / - / -	532.1 / - / -	513.1 / - / -
S2ST, En-Es			
VoxPopuli [6, 39]	126k / - / -	414.7 / - / -	424.1 / - / -
ASR, Es			
MLS [40]	220k / 2.4k / 2.4k	918 / 10.0 / 10.0	-
CommonVoice 7.0 [41]	196k / 15.3k / 15.3k	290 / 25.7 / 26.2	-
ASR, En			
Librispeech [42]	282k / 5.6k / 5.5k	960 / 10.5 / 10.7	-
TED-LIUM3 [43]	268k / 507 / 1.2k	452 / 1.6 / 2.6	-
Unlabeled Speech, Es			
VoxPopuli [39]	2.0M	16k	-
Unlabeled Speech, En			
VoxPopuli [39]	1.8M	14k	-
Librilight [44]	18.6M	60k	-
Parallel Text, Es-En & En-Es			
OpenSubtitle2018 [45]	64.7M	-	-
UNCorpus [46]	25M	-	-
EUBookshop v2 [47]	5.3M	-	-
Europarl v10 [48]	1.9M	-	-
Wikipedia v1.0 [49]	1.8M	-	-
TED2020 v1 [50]	0.4M	-	-

all test sets. The second S2ST dataset is from VoxPopuli [53], which contains speech from European parliament plenary sessions and the oral interpretation (dubbed as "*S2ST-real*"). We use the same training data in [6] and apply the speech normalizer from the 1-hr setup on the target speech.

The wav2vec 2.0 speech encoder and unit mBART are pre-trained on **unlabeled speech data**. For VoxPopuli [53], we remove utterances from the year 2012 and before to avoid overlap with the Europarl-ST [36] dev and test data. For data augmentation, we use all the **ASR data** and the text transcriptions of the source speech in the S2T datasets to train the ASR models, and all the **parallel text data** to train the MT models.

4.2. Model setup

We use the multilingual HuBERT (mHuBERT) model, k-means model and unit-based HiFi-GAN vocoder from [6][2], to encode target speech into a vocabulary of 1000 units. The mHuBERT and the k-means models are learned from the combination of En, Es and French unlabeled speech data from VoxPopuli [53], while we use them to encode En and Es target speech only.

We train the Conformer wav2vec 2.0 speech encoder with the LARGE configuration [20] using Libri-light [44] for En and VoxPopuli [53] for Es, respectively, for 200k updates with a batch size of 19.4-hr for Es and 14.7-hr for En. We train the unit mBART with the LARGE configuration [31] using the combination of all En and Es unlabeled speech for 500k updates with $\lambda = 10$ and $p = 0.3$, and we do not use the sentence permutation noise. During finetuning, we tune the hyper-parameters including learning rate ([5e-5, 1e-4]), dropout ([0.1, 0.3]) and also encoder specific ones namely mask channel length ([10, 32, 64]), mask probability ([0.1, 0.5]) and channel mask probability ([0.1, 0.5]) on the dev sets.

4.3. Baselines

We build two cascaded baselines, ASR+MT+TTS and S2T+TTS[3], and two supervised S2UT baselines: 1. **ASR**: The En ASR model is finetuned with CTC from the Conformer wav2vec 2.0 model. We apply the same training for Es but find that a supervised ASR model with the `s2t_transformer_l` architecture in FAIRSEQ is better. 2. **MT**: As the ASR models are trained with normalized text (e.g. lowercase, digits in spoken form, etc.), we apply text normalization on both source and target texts as well to train Es-En and En-Es MT models. We use the `transformer_wmt_en_de_big` architecture in FAIRSEQ. The MT models are also used in data augmentation. 3. **S2T**: The S2T model consist of the pre-trained Conformer wav2vec 2.0 encoder and a randomly initialized text decoder with 6 Transformer layers, 8 attention heads, 256 embedding size and 2048 FFN embedding size, and is trained on the S2T datasets without multitask learning. 4. **Supervised S2UT**: We follow the same model configuration in [4] to train Transformer-based S2UT models and explore both without and with multitask learning. For the latter we include two auxiliary tasks that use character sequences from source and target text transcripts as targets.

4.4. Evaluation

To evaluate the translation quality, we use open-sourced ASR models[4] to transcribe the audios and compute BLEU scores using SACREBLEU [54]. The reference text is normalized to lowercase, punctuation is removed, digits are converted to spoken forms, and all words in parentheses like "(Applause)" or "(Music)" are removed. We do not consider samples with empty translation after text normalization. To evaluate the naturalness of the speech output, we collect mean opinion scores (MOS) on a scale of 1 (the worst) to 5 (the best) from human listening tests on a set of 200 utterances randomly sampled across all the test sets for each system, and each sample is rated by 7 raters.

4.5. Results

4.5.1. S2ST with model pre-training

Table 2 shows results from models trained with "*S2ST-syn*" data. We also provide BLEU from the synthetic targets (11) to demonstrate the impact of ASR errors. First, we see that without multitask, the supervised Es-En S2UT model (3) cannot converge properly with the combined 196-hr training set, while multitask learning helps model training (4).

Next, we see that with a pre-trained wav2vec 2.0 encoder and a randomly initialized decoder, we can achieve an average of 4.0 BLEU gain on En-Es test sets and 5.8 BLEU for Es-En (4 vs. 5). As we incorporate a unit mBART decoder, we find that **LNA-D** is the most effective finetuning strategy, yielding an average of 6.3 BLEU gain on En-Es test sets and 8.0 BLEU on Es-En compared to multitask learning (4 vs. 7). Our best En-Es S2ST model performs on par with the S2T+TTS baseline, and the Es-En S2ST model outperforms the cascaded system by 2.7 BLEU (1 vs. 7). Though the S2T system can be further improved with text-based pre-training, this is beyond our scope.

Further incorporating weakly supervised training data from ASR speech can bring +2.1 BLEU on En-Es and +3.3 BLEU on Es-En (7 vs. 10). We can also compare this system with the three-stage cascaded system, as they incorporate information from the same amount of supervised ASR and MT data. We see that the En-Es direct system can outperform ASR+MT+TTS,

[4]En: `https://huggingface.co/facebook/wav2vec2-large-960h-lv60-self`, Es: `https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-spanish`

Table 2: *Dev / test BLEU on all the datasets included in the "S2ST-syn" data. All S2UT systems are decoded with beam size 10. MOS is reported with 95% confidence interval. (w2v2-L: wav2vec 2.0* LARGE*)*

		En-Es			Es-En			
		BLEU		MOS	BLEU			MOS
ID		Europarl-ST	MuST-C	combined	CoVoST-2	Europarl-ST	mTEDx	combined
Cascaded systems:								
1	S2T (w2v2-L)+TTS	33.0 / 32.6	30.3 / 30.1	3.80 ± 0.12	25.9 / 28.4	26.9 / 23.6	25.3 / 21.5	3.53 ± 0.14
2	ASR+MT+TTS	28.9 / 28.8	36.4 / 34.2	-	37.3 / 33.8	33.3 / 29.1	29.3 / 32.4	-
S2UT systems without pre-training:								
3	S2UT (w/o multitask) [4]	23.8 / 24.0	25.0 / 23.3	-	0.0 / 0.0	0.0 / 0.0	0.1 / 0.0	-
4	S2UT (w/ multitasks) [4]	25.5 / 25.8	26.3 / 24.3	3.97 ± 0.09	20.6 / 22.7	20.4 / 18.0	20.2 / 16.9	3.26 ± 0.09
S2UT systems with model pre-training:								
5	w2v2-L	28.9 / 28.4	29.9 / 29.6	3.35 ± 0.15	26.1 / 29.3	25.9 / 23.1	**27.8** / 22.5	3.15 ± 0.14
6	w2v2-L + mBART (LNA-E)	30.6 / 30.4	30.2 / 28.7	-	24.2 / 26.9	24.0 / 21.6	24.5 / 19.6	-
7	w2v2-L + mBART (LNA-D)	**31.5 / 31.9**	**32.1 / 30.7**	4.06 ± 0.10	**27.8 / 30.9**	**28.9 / 25.3**	22.1 / **25.5**	2.81 ± 0.16
8	w2v2-L + mBART (LNA-E,D)	27.8 / 30.2	23.9 / 26.1	-	26.4 / 29.5	27.1 / 24.9	24.0 / 21.7	-
9	w2v2-L + mBART (full)	29.4 / 29.0	30.0 / 27.1	-	26.2 / 29.2	25.3 / 23.0	23.3 / 23.0	-
S2UT systems with model pre-training and data augmentation:								
10	w2v2-L + mBART (LNA-D)	**33.5 / 33.7**	**34.5 / 33.1**	-	**30.9 / 34.1**	**31.9 / 28.2**	**30.8 / 29.3**	-
11	Synthetic target	89.4 / -	86.3 / -	-	81.3 / -	85.8 / -	88.4 / -	-

Table 3: *Dev and test BLEU on Europarl-ST from models trained with the "S2ST-real" data.*

	En-Es		Es-En	
	dev	test	dev	test
w/ multitask [6]	-	21.8	-	18.8
w2v2-L + mBART (LNA-D)	26.2	26.5	25.2	23.1

Table 4: *Dev and test BLEU from models trained with 10-hr, 50-hr, and 100-hr subsets of "S2ST-syn". Results are averaged from the multiple datasets of each language direction.*

	hours	En-Es		Es-En	
		dev	test	dev	test
w/ multitask [4]	10	0.5	0.5	0.8	0.7
w2v2-L + mBART (LNA-D)	10	0.2	0.3	0.5	0.6
w/ multitask [4]	30	7.6	7.9	8.2	7.4
w2v2-L + mBART (LNA-D)	30	10.4	10.7	13.2	12.4
w/ multitask [4]	50	9.7	10.1	11.9	11.1
w2v2-L + mBART (LNA-D)	50	14.8	14.7	20.3	19.5
w/ multitask [4]	100	12.1	13.1	15.6	14.6
w2v2-L + mBART (LNA-D)	100	26.0	25.1	24.8	23.8

while there is a 1.2 BLEU gap for Es-En systems (2 vs. 10).

For MOS, Table 2 shows that the En-Es S2UT system produces more natural speech than Es TTS does (1 vs. 7), while the quality of the Es-En S2UT output is much worse. Note that the naturalness of the output speech is mainly controlled by the unit vocoder, and we use the models from [6] without finetuning.

One advantage of self-supervised pre-training is that it only uses speech data and can work for unwritten languages. We finetune on "*S2ST-real*" data and compares with [6] that incorporates an auto-encoding auxiliary task to predict the discrete units extracted from the source speech. We use dev and test sets from Europarl-ST, since it's in the same domain as the "*S2ST-real*" data. Table 3 shows a 4.7 and 4.3 BLEU gain for En-Es and Es-En with **LNA-D**.

Finally, we study the effect of pre-training, with a decreasing amount of parallel data for finetuning by randomly sampling a subset from the "*S2ST-syn*" training data. Table 4 compares **LNA-D** finetuning with supervised S2UT models trained with multitask learning. When training on less than 100 hours of data, we reduce the supervised S2UT model to 8 encoder layers and 4 decoder attention heads. We see an average 4.6-12.0 BLEU gain from pre-training and **LNA-D** finetuning for both language directions when training with more than 30 hours of data. Pre-training and finetuning with 50-hr data can already outperform supervised systems trained with 100-hr data. However, when the amount of parallel data is decreased to 10 hrs, both multitask learning and pre-training cannot work.

4.5.2. Model variations

First, we examine Conformer vs. Transformer wav2vec 2.0. We perform finetuning with the pre-trained encoder and a randomly initialized unit decoder using Es-En "*S2ST-syn*" data and see that Conformer wav2vec 2.0-LARGE model gives an average 4.6 BLEU gain compared with Transformer LARGE model.

Next, we study how λ and p affect unit mBART by training the model for 300k updates and finetuning on a unit-to-unit translation task, where both source and target speech are converted to *reduced* discrete unit sequences. From Table 5, we do not see large difference except when $p = 0.3$ and $\lambda = 5$.

Table 5: *Average BLEU on "S2ST-syn" En-Es dev sets with respect to hyper-parameters used in unit mBART training.*

		λ, *span_size*		
En-Es dev BLEU		5	10	15
p, *mask_ratio*	0.3	22.7	23.6	23.2
	0.5	23.2	23.1	23.5
	0.7	23.4	23.5	23.3

5. Conclusions

In this work, we study self-supervised pre-training and data augmentation for direct S2ST models. We take advantage of an S2UT framework that encodes target speech into discrete representations, apply wav2vec 2.0 speech encoder and unit mBART decoder pre-training and perform partial finetuning. Experiments under various setups including synthetic and real target speech and low-resource all verify the effectiveness of the approach. We also show that applying MT to create weakly supervised data from speech in the source language can be further combined with pre-training to improve model performance.

6. Acknowledgements

We would like to thank Justine Kao and Brian Bui for the help on MOS evaluation.

7. References

[1] Y. Jia, R. J. Weiss *et al.*, "Direct speech-to-speech translation with a sequence-to-sequence model," *Proc. Interspeech*, 2019.

[2] A. Tjandra, S. Sakti *et al.*, "Speech-to-speech translation between untranscribed unknown languages," in *ASRU*, 2019.

[3] C. Zhang, X. Tan *et al.*, "UWSpeech: Speech to speech translation for unwritten languages," *arXiv:2006.07926*, 2020.

[4] A. Lee, P.-J. Chen *et al.*, "Direct speech-to-speech translation with discrete units," *arXiv:2107.05604*, 2021.

[5] Y. Jia, M. T. Ramanovich *et al.*, "Translatotron 2: Robust direct speech-to-speech translation," *arXiv:2107.08661*, 2021.

[6] A. Lee, H. Gong *et al.*, "Textless speech-to-speech translation on real data," *arXiv:2112.08352*, 2021.

[7] A. Lavie, A. Waibel *et al.*, "JANUS-III: Speech-to-speech translation in multiple languages," in *ICASSP*, 1997.

[8] S. Nakamura, K. Markov *et al.*, "The ATR multilingual speech-to-speech translation system," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 365–376, 2006.

[9] K. Lakhotia, E. Kharitonov *et al.*, "Generative spoken language modeling from raw audio," *arXiv:2102.01192*, 2021.

[10] A. Polyak, Y. Adi *et al.*, "Speech resynthesis from discrete disentangled self-supervised representations," *arXiv:2104.00355*, 2021.

[11] A. Bérard, L. Besacier *et al.*, "End-to-end automatic speech translation of audiobooks," in *ICASSP*, 2018.

[12] P. Bahar, T. Bieschke *et al.*, "A comparative study on end-to-end speech to text translation," in *ASRU*, 2019.

[13] M. C. Stoian, S. Bansal *et al.*, "Analyzing asr pretraining for low-resource speech-to-text translation," in *ICASSP*, 2020.

[14] X. Li, C. Wang *et al.*, "Multilingual speech translation from efficient finetuning of pretrained models," in *ACL*, 2021.

[15] R. J. Weiss, J. Chorowski *et al.*, "Sequence-to-sequence models can directly translate foreign speech," *Proc. Interspeech*, 2017.

[16] Y. Jia, M. Johnson *et al.*, "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," in *ICASSP*, 2019.

[17] J. Pino, Q. Xu *et al.*, "Self-training for end-to-end speech translation," *Proc. Interspeech*, 2020.

[18] C. Wang, A. Wu *et al.*, "Large-scale self-and semi-supervised learning for speech translation," *arXiv:2104.06678*, 2021.

[19] T. Kano, S. Sakti *et al.*, "Transformer-based direct speech-to-speech translation with transcoder," in *SLT*, 2021.

[20] A. Baevski, Y. Zhou *et al.*, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Neurips*, vol. 33, pp. 12 449–12 460, 2020.

[21] Y. Liu, J. Gu *et al.*, "Multilingual denoising pre-training for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020.

[22] S.-w. Yang, P.-H. Chi *et al.*, "SUPERB: Speech processing universal performance benchmark," *arXiv:2105.01051*, 2021.

[23] W.-N. Hsu, B. Bolte *et al.*, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *arXiv:2106.07447*, 2021.

[24] S. Chen, C. Wang *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *arXiv:2110.13900*, 2021.

[25] J. Ao, R. Wang *et al.*, "SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing," *arXiv:2110.07205*, 2021.

[26] E. Kharitonov, A. Lee *et al.*, "Text-free prosody-aware generative spoken language modeling," *arXiv:2109.03264*, 2021.

[27] F. Kreuk, A. Polyak *et al.*, "Textless speech emotion conversion using decomposed and discrete representations," *arXiv:2111.07402*, 2021.

[28] A. Tjandra, S. Sakti *et al.*, "Machine speech chain with one-shot speaker adaptation," *Proc. Interspeech*, 2018.

[29] J. Kahn, A. Lee *et al.*, "Self-training for end-to-end speech recognition," in *ICASSP*, 2020.

[30] T. Hayashi, S. Watanabe *et al.*, "Back-translation-style data augmentation for end-to-end asr," in *SLT*, 2018.

[31] A. Vaswani, N. Shazeer *et al.*, "Attention is all you need," in *Neurips*, 2017, pp. 5998–6008.

[32] A. Gulati, J. Qin *et al.*, "Conformer: Convolution-augmented Transformer for speech recognition," *Proc. Interspeech*, 2020.

[33] M. Ott, S. Edunov *et al.*, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proc. NAACL-HLT 2019: Demonstrations*, 2019.

[34] C. Wang, Y. Tang *et al.*, "fairseq S2T: Fast speech-to-text modeling with fairseq," in *AACL: System Demonstrations*, 2020.

[35] C. Wang, A. Wu *et al.*, "Covost 2 and massively multilingual speech-to-text translation," *arXiv:2007.10310*, 2020.

[36] J. Iranzo-Sánchez, J. A. Silvestre-Cerdà *et al.*, "Europarl-st: A multilingual corpus for speech translation of parliamentary debates," in *ICASSP*, 2020.

[37] E. Salesky, M. Wiesner *et al.*, "Multilingual tedx corpus for speech recognition and translation," in *Proc. of Interspeech*, 2021.

[38] R. Cattoni, M. A. Di Gangi *et al.*, "Must-c: A multilingual corpus for end-to-end speech translation," *Computer Speech & Language*, vol. 66, p. 101155, 2021.

[39] C. Wang, M. Riviere *et al.*, "Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," *arXiv:2101.00390*, 2021.

[40] V. Pratap, Q. Xu *et al.*, "MLS: A large-scale multilingual dataset for speech research," *arXiv:2012.03411*, 2020.

[41] R. Ardila, M. Branson *et al.*, "Common voice: A massively-multilingual speech corpus," *arXiv:1912.06670*, 2019.

[42] V. Panayotov, G. Chen *et al.*, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015.

[43] F. Hernandez, V. Nguyen *et al.*, "TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *International conference on speech and computer*. Springer, 2018, pp. 198–208.

[44] J. Kahn, M. Riviere *et al.*, "Libri-light: A benchmark for ASR with limited or no supervision," in *ICASSP*, 2020.

[45] P. Lison, J. Tiedemann *et al.*, "OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora," in *LREC*, 2018.

[46] M. Ziemski, M. Junczys-Dowmunt *et al.*, "The united nations parallel corpus v1. 0," in *LREC*, 2016.

[47] R. Skadiņš, J. Tiedemann *et al.*, "Billions of parallel words for free: Building and using the EU bookshop corpus," in *LREC*, 2014.

[48] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proc. machine translation summit x: papers*, 2005.

[49] K. Wołk and K. Marasek, "Building subject-aligned comparable corpora and mining it for truly parallel sentence pairs," *Procedia Technology*, vol. 18, pp. 126–132, 2014.

[50] N. Reimers and I. Gurevych, "Making monolingual sentence embeddings multilingual using knowledge distillation," in *EMNLP*, 2020.

[51] T. Likhomanenko, Q. Xu *et al.*, "Rethinking evaluation in asr: Are our models robust enough?" *arXiv:2010.11745*, 2020.

[52] W. Chan, D. Park *et al.*, "Speechstew: Simply mix all available speech recognition data to train one large neural network," *arXiv:2104.02133*, 2021.

[53] C. Wang, M. Riviere *et al.*, "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," in *ACL*, 2021.

[54] M. Post, "A call for clarity in reporting BLEU scores," in *Proc. the Third Conference on Machine Translation: Research Papers*, 2018, pp. 186–191.

A. Model Training Hyper-parameters

A.1. Wav2vec 2.0

To train the wav2vec 2.0 models, we use the `wav2vec2_conformer_large_librivox` configuration defined in FAIRSEQ. The model contains 24 Conformer blocks with model dimension 1024, inner dimension 4096, 16 attention heads and a total of 620.5M parameters. We use a total batch size of 14.7-hr for En and 19.4-hr for Es, and dropout 0.1. We use Adam with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-6}$, weight decay of 0.1 and learning rate 0.005, and apply polynomial decay learning rate schedule with 32k warmup steps. We started the training with fp16 initially but switched to fp32 when we encountered NaNs in the back propagation. Rest of the hyper-parameters are listed in Table 6.

Table 6: *Hyper-parameters used in wav2vec 2.0 training*

hyper-parameter	En	Es
max tokens per GPU	550,000	700,000
#GPUs	256	400
max positions	5000	5000
update freq	6	4
max updates	220,000	188,000

A.2. Unit mBART

We use the `mbart_large` architecture defined in FAIRSEQ to train the unit mBART. The model contains 12 Transformer layers in both encoder and decoder, embedding size 1024, feed-forward network (FFN) dimension 4096, 16 attention heads and a total of 353M parameters. Different from original text mBART, we do not apply sentence permutation noise in the denoising task, and we only apply masking noise. We use Adam with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-6}$ and learning rate 0.0003, and apply polynomial decay learning rate schedule with 10000 warmup steps. The model is trained with dropout 0.1, mask probability 0.3, $\lambda = 10$ for 500k steps. We use 1024 max tokens with 64 GPUs and update frequency 9.

A.3. Supervised S2UT baselines

To train the supervised S2UT models with multitasks, we use the `s2ut_transformer_fisher` configuration defined in FAIRSEQ. We use Adam with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-8}$, weight decay of 0.1 and learning rate 0.0005, and apply inverse square root learning rate schedule with 10k warmup steps. For En-Es, we use a max tokens value of 3000, dropout value of 0.1 and 16 GPUs. For Es-En, we use 4500 max tokens value, dropout 0.3 and 8 GPUs.

For auxiliary tasks, we have a Transformer decoder on the sixth layer of the encoder for source character prediction and a Transformer decoder on the eighth layer of the encoder for target character prediction. Each of the multitasks has a loss weight of 8. The decoders have 2 Transformer layers with 256 embedding size and FFN dimension of 2048.

A.4. Finetuning

We use a learning rate of 0.0005, a dropout value of 0.1, encoder layerdrop value of 0.1 and label smoothing 0.2 for both language directions. We set the mask prob value to 0.3 and mask channel prob value to 0.25 in the wav2vec 2.0 encoder. For the decoder, we use the same embedding parameters for the input and output. We use Adam with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-6}$, weight decay of 0.1 and apply inverse square root learning rate schedule with 10k warmup steps. We finetune the models for 15k updates with "*S2ST-syn*" data and 25k updates when incorporating the weakly supervised data. We use max tokens of 3500 for En-Es and 4500 for Es-En, set the update frequency to 15 and train on 16 GPUs. The mask channel length and the number of updates we freeze the wav2vec 2.0 encoder in the beginning for each of the setups is listed in the Table 7.

Table 7: *Hyper-parameters used in wav2vec 2.0 and mBART finetuning*

	#Parameters Finetuned (Millions)	En-Es mask channel length	En-Es encoder freeze steps	Es-En mask channel length	Es-En encoder freeze steps
w2v2-L	628.9	64	5000	32	5000
w2v2-L + mBART (LNA-E)	335.1	64	0	32	0
w2v2-L + mBART (LNA-D)	725.7	32	5000	64	0
w2v2-L + mBART (LNA-E,D)	233.3	64	0	32	0
w2v2-L + mBART (full)	827.4	64	5000	32	0
Data augmentation:					
w2v2-L + mBART (LNA-D)	725.7	32	5000	32	0

A.5. Low-resource setup

Table 8 lists the statistics of the training data used in the low-resource setup. We use the same set of hyper-parameters for the wav2vec 2.0 and mBART finetuning on the low-resource setup as described in Sec. A.4 except for the ones listed in Table 9. Finally, Table 10 lists the BLEU scores on each dataset under the low-resource setup, whose average values are presented in Table 4.

Table 8: *Duration (hours) of the training data sampled from each dataset in the low-resource setup*

	En-Es		Es-En		
total hours	Europarl-ST	MuST-C	CoVoST-2	Europarl-ST	mTEDx
10	1.4	8.6	5.8	1.0	3.2
30	4.0	26.0	17.2	3.0	9.9
50	6.7	43.4	28.7	5.2	16.1
100	13.5	86.5	57.4	10.2	32.3

Table 9: *Hyper-parameters used in low-resource LNA-D finetuning experiments*

	En-Es			Es-En		
hours	learning rate	mask channel length	encoder freeze steps	learning rate	mask channel length	encoder freeze steps
10	0.0005	32	5000	0.0005	64	0
30	0.001	64	0	0.0005	32	0
50	0.0005	32	5000	0.0005	32	0
100	0.0005	32	5000	0.0005	64	0

Table 10: *Dev / test BLEU on all the datasets included in the "S2ST-syn" data in the low-resource setup.*

		En-Es BLEU		Es-En BLEU		
	hours	Europarl-ST	MuST-C	CoVoST-2	Europarl-ST	mTEDx
w/ multitask [4]	10	0.6 / 0.7	0.3 / 0.3	1.2 / 1.0	1.0 / 0.9	0.3 / 0.3
w2v2-L + mBART (LNA-D)	10	0.3 / 0.4	0.2 / 0.1	0.3 / 0.8	0.7 / 0.7	0.3 / 0.2
w/ multitask [4]	30	8.1 / 7.9	7.0 / 7.8	9.2 / 9.5	8.0 / 6.8	7.4 / 6.0
w2v2-L + mBART (LNA-D)	30	10.9 / 11.2	9.9 / 10.2	15.1 / 17.1	11.8 / 10.3	12.5 / 9.7
w/ multitask [4]	50	10.4 / 10.4	8.9 / 9.8	12.5 / 13.6	11.7 / 10.2	11.5 / 9.4
w2v2-L + mBART (LNA-D)	50	14.7 / 15	14.9 / 14.5	22.3 / 24.7	20.3 / 18.2	18.2 / 15.4
w/ multitask [4]	100	13.3 / 13.6	11.0 / 12.6	16.6 / 18.2	15.1 / 13.1	15.1 / 12.6
w2v2-L + mBART (LNA-D)	100	25.6 / 25.8	26.5 / 24.4	26 / 28.6	25.8 / 23.1	22.6 / 19.6