CROSS-LINGUAL TRANSFER LEARNING FOR LOW-RESOURCE SPEECH TRANSLATION

Sameer Khurana[†], *Nauman Dawalatabad*[♣], *Antoine Laurent*[⋆],
Luis Vicente[ς], *Pablo Gimeno*[ς], *Victoria Mingote*[ς], *James Glass*[♣]
[†]Mitsubishi Electric Research Lab
[♣]MIT Computer Science and Artificial Intelligence Laboratory
[⋆]LIUM University
[ς]University of Zaragoza

ABSTRACT

The paper presents a novel three-step transfer learning framework for enhancing cross-lingual transfer from high- to low-resource languages in the downstream application of Automatic Speech Translation. The approach integrates a semantic knowledge-distillation step into the existing two-step cross-lingual transfer learning framework XLS-R. This extra step aims to encode semantic knowledge in the multilingual speech encoder pre-trained via Self-Supervised Learning using unlabeled speech. Our proposed three-step cross-lingual transfer learning framework addresses the large cross-lingual transfer gap (TRFGap) observed in the XLS-R framework between high-resource and low-resource languages. We validate our proposal through extensive experiments and comparisons on the CoVoST-2 benchmark, showing significant improvements in translation performance, especially for low-resource languages, and a notable reduction in the TRFGap.

***Index Terms*—** Cross-lingual transfer Learning, Automatic Speech Translation

1. INTRODUCTION

End-to-end multilingual speech translation technology has recently seen dramatic improvements owing to the widely used two-step Transfer Learning (TL) framework, self-supervised pre-training, followed by supervised fine-tuning. A large transformer encoder is pre-trained using self-supervised learning on massive amounts of unlabeled multilingual speech data. This is followed by multi-task supervised fine-tuning of the pre-trained encoder on several speech-to-text translation tasks. A popular two-step transfer learning framework is the Cross-Lingual Speech Representation (XLS-R) framework [1].

Cross-Lingual Transfer Gap. XLS-R consists of pre-training a transformer encoder using Self-Supervised Learning (SSL) on 400K hours of unlabeled speech in 128 languages collected from diverse speech datasets. The pre-trained encoder is then fine-tuned using multi-task supervised learning on 21 speech-to-text translation tasks of the form X→EN. X refers to a source language, and the learning task is to translate speech in X to text in English. The paired data for the 21 X→EN translation tasks comes from the Common Voice Speech Translation (CoVoST) corpus [2]. Depending on the amount of labeled data for each task, we categorize them into high, mid, and low-resource. High-resource tasks consist of more than 100 hours, mid-resource between 10 and 100 hours, and low-resource less than 10 hours of paired speech (X) and text (EN) translation data for fine-tuning. To set the problem statement, we show the performance of the two-step XLS-R cross-lingual TL framework described above on the CoVoST X→EN benchmark (Table 1).

Model	High	Mid	Low	**TRFGap**
XLS-R-0.3B	30.6	18.9	5.1	**25.1**
XLS-R-1B	34.3	25.5	11.7	**22.6**
XLS-R-2B	36.1	27.7	15.1	**21**

Table 1. Problem Statement (BLEU-4 scores).

Notice the sizeable cross-lingual transfer gap (**TRFGap**), defined as the performance difference between high- and low-resource tasks. The substantial TRFGap implies that the knowledge acquired by the translation model while learning to perform high-resource translation tasks does not help learn the low-resource translation tasks well. Since the translation model is built on the knowledge acquired by the speech encoder during the SSL pre-training step, we hypothesize that the XLS-R framework's pre-training step should be improved to facilitate better cross-lingual knowledge transfer during fine-tuning for multilingual translation, thus reducing the TRFGap. To that end, we propose a novel three-step TL framework.

Proposed Solution. We propose a **three-step TL framework** to reduce the abovementioned TRFGap. A semantic

SK did this work when he was at MIT. This work was partially performed using HPC resources from GENCI–IDRIS (Grant AD011012527) and has received funding from the H2020 Marie Skłodowska-Curie agreement No 101007666 and from the DGA/AID RAPID COMMUTE project.

knowledge-distillation (KD) step, SAMU-XLS-R, proposed in [3], is inserted between the SSL pre-training and fine-tuning steps of the XLS-R framework. We hypothesize that the XLS-R's pre-training step does not encode semantic knowledge in the speech encoder, and by injecting semantic knowledge, we can reduce the TRFGap.

Our proposed novel three-step cross-lingual TL framework (§2) consists of: **1) SSL pre-training** of speech encoder similar to XLS-R, **2) Semantic KD** borrowed from SAMU-XLS-R [3], to encode semantic knowledge in the pre-trained encoder, and **3) Adapter [4] based Multi-task fine-tuning** of the encoder on several speech-to-text translation tasks.

2. PROPOSED CROSS-LINGUAL TL FRAMEWORK

SSL Pre-training (xlsr-0.3B). This step is borrowed from the XLS-R contrastive pre-training method presented in [1], originally proposed in [5]. A transformer encoder is pre-trained using unlabeled speech in 128 languages. See [1] for exact pre-training details and transformer architecture. We use the 300M (0.3B) parameter pre-trained encoder checkpoint[1]. From now on, we refer to this pre-trained SSL checkpoint xlsr-0.3B.

Semantic KD (samu-xlsr-0.3B). We fine-tune xlsr-0.3B via semantic KD as proposed in SAMU-XLS-R [3]. This framework uses paired tuples $(\mathbf{x}, \mathbf{y})^l$ for training, where $\mathbf{x}$ is a speech waveform, $\mathbf{y}$ is its corresponding text transcript, and l refers to the language of speech and text. The training framework consists of the speech and text encoding branches.

The speech branch transforms the sample sequence $\mathbf{x} \in \mathbb{R}^{(S\times 1)}$ into a vector embedding $\mathbf{e}_{\text{speech}} \in \mathbb{R}^{1\times D}$. The text branch transforms the corresponding transcript $\mathbf{y}$ into a vector embedding $\mathbf{e}_{\text{text}} \in \mathbb{R}^{1\times D}$. The parameters of the speech branch are fine-tuned, while the text branch remains frozen during training. The cosine distance between the speech and the text embeddings gives the learning objective. The speech branch consists of the pre-trained xlsr-0.3B (from the previous step) that transforms $\mathbf{x}$ into an embedding sequence $\mathbf{Z} \in \mathbb{R}^{N\times D}$, followed by a self-attention-based temporal pooling mechanism with a single learnable query vector [6], that outputs the speech embedding $\mathbf{e}_{\text{speech}}$. The text branch comprises the pre-trained Language-Agnostic BERT Sentence Encoder (LaBSE) [7] that transforms the transcript into an embedding sequence $\mathbf{H} \in \mathbb{R}^{M\times D}$. The first embedding in the sequence $\mathbf{H}$ is the [CLS] token embedding, which we use as the text embedding $\mathbf{e}_{\text{text}}$.

xlsr-0.3B can encode speech in 128 languages, while LaBSE can encode text in 109 languages. The paired $(\mathbf{x}, \mathbf{y})^l$ tuples for training are derived from the CommonVoice-version8 dataset [8] using the intersection of the sets of languages supported by xlsr-0.3B, and LaBSE. This amounts to 13K hours of training data in 53 languages. Initially, semantic KD was performed in [3] using paired data in 25 languages. Also, unlike the original work, we use speed perturbation [9] with factors $\{0.9, 1.0, 1.1\}$ to increase the size of the training data threefold. See [3] for a detailed explanation of the semantic KD learning framework. Moving forward, we refer to the xlsr-0.3B fine-tuned using the abovementioned semantic KD as samu-xlsr-0.3B.

Adapter-based Multi-task fine-tuning. Finally, we fine-tune samu-xlsr-0.3B for speech-to-text translation. The translation model is a transformer comprising samu-xlsr-0.3B (from the previous step) as the encoder and pre-trained MBART as the decoder. MBART [10] is a multilingual text-to-text translation model trained to translate text in 50 languages to English. We use the autoregressive transformer decoder of the MBART checkpoint[2] to initialize the decoder of our speech-to-text translation model. The translation model's training data comprises tuples $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ is the speech waveform in a source language, and $\mathbf{y}$ is its text translation in a different target language. The samu-xlsr-0.3B encoder transforms the speech waveform $\mathbf{x}$ into an embedding sequence $\mathbf{H}$. The MBART decoder models the likelihood function autoregressively $p(\mathbf{y}|\mathbf{x}) = p(y_1|\mathbf{H})p(y_2|y_1, \mathbf{H}) \dots p(y_n|y_{1:n-1}, \mathbf{H})$. The model is trained to maximize the log-likelihood function. We use teacher forcing during training. The model generates text translation via beam search during inference.

The translation model consists of 700M trainable parameters. We only fine-tune 75M. Following [11], in the MBART decoder, we fine-tune the parameters of cross-attention (CA) and layernorm [12] modules while keeping self-attention frozen. Since CA of the MBART decoder was previously trained using input from the MBART text encoder, it has to be retrained for the input from samu-xlsr-0.3B encoder. Layernorm is task and data-dependent; hence, it's retrained. In the samu-xlsr-0.3B encoder, we keep all the parameters frozen to their pre-trained values and insert new parameters as adapter layers [4]. Two adapter layers are inserted in each layer of samu-xlsr-0.3B transformer encoder, one after the self-attention and the other after the feedforward layers. An adapter layer has an hourglass architecture. The input and output layers have the same size, while the hidden layer size is a fraction of the input layer. We found the optimal size (according to a dev set) of the hidden layer is one-fourth of the input.

The motivation for using adapters in samu-xlsr-0.3B is to avoid *catastrophic forgetting* [13] of semantic knowledge acquired via the semantic KD step (previous step) of our TL framework. We show (Table 6) that adapter-based fine-tuning

[1] https://dl.fbaipublicfiles.com/fairseq/wav2vec/xlsr2_300m.pt

[2] https://dl.fbaipublicfiles.com/fairseq/models/mbart50/mbart50.ft.n1.tar.gz

strategy is essential to achieve good cross-lingual transfer from high to low-resource language translation tasks.

The translation model training data is derived from the CommonVoice Speech Translation-2 (CoVoST-2) dataset [2]. CoVoST-2 consists of 21 X→English speech→text translation tasks. We train our above mentioned transformer on all 21 translation tasks simultaneously. The decoder is conditioned with a language ID to distinguish between translation tasks. All the model parameters are shared across the tasks. The model is trained using an Adam optimizer with a maximum learning rate of 5e-4 and a three-phase learning rate scheduler, similar to the semantic KD step (previous step in our TL framework). The model is trained on 8 A100 Nvidia GPUs, with a batch size of 10 minutes of speech and corresponding text translations.

3. EXPERIMENTAL RESULTS

Training & Evaluation. Translation models are trained and evaluated on the 21 X→English speech→text translation tasks of the CoVoST-2 benchmark. The 21 tasks are categorized into high (more than 100 hours of training data), mid (between 10 and 100 hours of training data), and low-resource tasks (less than 10 hours of training data). There are four high, five mid, and 12 low-resource tasks. We report the average BLEU-4 score on the three categories.

Baseline models. We compare our proposed translation model (§2; Adapter-based Multi-task fine-tuning) against several other translation models. All translation models are transformers with MBART decoder initialization. The models differ in encoder initialization and training. Adapter-based encoder fine-tuning is only performed for the samu-xlsr-0.3B translation model, while all the encoder parameters are fine-tuned for other models. We later show that adapter-based fine-tuning brings gains only for the samu-xlsr-0.3B translation model. For decoder fine-tuning, only cross-attention and layernorm parameters are fine-tuned for all the models. 1) **xlsr-[0.3B, 1B, 2B]**: refers to different-sized transformer encoders trained using unlabeled speech via SSL. Note that samu-xlsr-0.3B is built on top of xlsr-0.3B via semantic KD. 2) **mslam-[0.6B, 2B]**: the mslam encoders [14] are trained using a mix of SSL and supervised learning using paired speech-text tuples. Unlike samu-xlsr-0.3B, which is trained using semantic KD, mslam is not trained with explicit semantic supervision from the text modality.

Multilingual Translation Results. **Table 2** compares samu-xlsr-0.3B translation model with several xlsr based translation models. We make the following observations: 1) On high resource tasks, the xlsr-2B translation model performs the best, with samu-xlsr-0.3B lagging a couple of points behind. Compared to the similar-sized xlsr-0.3B model, samu-xlsr-0.3B performs 4 BLEU points better. 2) On mid-resource tasks, samu-xlsr-0.3B outperforms all the models, achieving a BLEU score of 31.1, significantly better than the xlsr-0.3B model's BLEU score of 18.9. Our model also outperforms the larger xlsr-2B speech encoder by 3.4 BLEU points. 3) On low-resource tasks, samu-xlsr-0.3B performs the best. Compared to the xlsr-0.3B model, it does better by 15 BLEU points. It also outperforms the much larger xlsr-2B by 5.2 BLEU points. The cross-lingual transfer gap (TRFGap), which is the difference in performance between high and low-resource task groups, is significantly less (14.1 BLEU) for the samu-xlsr-0.3B model than other models. Second to samu-xlsr-0.3B is xlsr-2B, which has a TRFGap of 21 BLEU points while having 500% more parameters.

Model	High	Mid	Low	TRFGap
xlsr-0.3B	30.6	18.9	5.1	25.1
xlsr-1B	34.3	25.5	11.7	22.6
xlsr-2B	**36.1**	27.7	15.1	21
samu-xlsr-0.3B	34.4	**31.1**	**20.3**	**14.1**

Table 2. We compare our proposed `samu-xlsr` based translation model with `xlsr` based translation models. The numbers are the average BLEU-4 scores.

Model	High	Mid	Low	TRFGap
mslam-0.6B	37.6	27.8	15.1	22.5
mslam-2B	**37.8**	29.6	18.5	19.3
samu-xlsr-0.3B	34.4	**31.1**	**20.3**	**14.1**

Table 3. We compare our proposed `samu-xlsr` based translation model with `mslam` based translation models.

Table 3 compares samu-xlsr-0.3B translation model with mslam based translation models. samu-xlsr-0.3B performs better on mid- and low-resource translation tasks. Importantly, samu-xlsr-0.3B has a lower cross-lingual transfer gap (TRFGap) between high and low resource groups of 14.1 BLEU points compared to 22.5 for mslam-0.6B and 19.3 for mslam-2B. The BLEU scores for mslam models are lifted from the paper [14] since these models are not publicly available.

Cascade vs. End-to-end Translation. We fine-tune xlsr-0.3B for Automatic Speech Recognition (ASR) using CTC framework [15]. The ASR model is used to transcribe speech into text. The transcription is translated into text in English using the pre-trained MBART checkpoint[3]. To train ASR, we use the same set of transcribed speech data used for the semantic KD of xlsr-0.3B. **Table 4** shows that samu-xlsr-0.3B end-to-end model is significantly better than the cascade model.

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

Model	High	Mid	Low	TRFGap
cascade	33.1	22.4	12.2	20.9
samu-xlsr-0.3B	34.4	**31.1**	**20.3**	**14.1**

Table 4. We compare samu-xlsr-0.3B end-to-end with a cascade translation model.

ASR Fine-tuning vs. Semantic KD. Here, we perform ablation over the second step in our three-step transfer learning framework. To get the samu-xlsr-0.3B speech encoder, we fine-tune the pre-trained xlsr-0.3B via semantic KD task (§2). Here, we fine-tune the pre-trained xlsr-0.3B encoder via ASR task. The ASR fine-tuning uses the same multilingual labeled speech data that was used for the semantic KD step. ASR fine-tuning is performed using the Connectionist Temporal Classification (CTC) framework [15]. The speech encoder we get after ASR fine-tuning is referred to as ctc-xlsr-0.3B. **Table 5** shows that CTC fine-tuning does not lead to significant TRFGap reduction, which enforces the importance of the SAMU-XLS-R semantic KD step. This result is not surprising since SSL pre-trained xlsr encoder is already good at few-shot ASR [1], implying that the information necessary for ASR is already encoded in xlsr's internal representations, and fine-tuning on ASR task does not add any new information.

Model	High	Mid	Low	TRFGap
xlsr-0.3B	30.6	18.9	5.1	25.1
ctc-xlsr-0.3B	31.6	20.9	8.5	23.1
samu-xlsr-0.3B	**34.4**	**31.1**	**20.3**	**14.1**

Table 5. Ablation-I: ASR Fine-tuning vs. Semantic KD as the second step in our proposed three-step TL framework.

Adapter vs. full encoder fine-tuning. We perform Adapter-based fine-tuning of the samu-xlsr-0.3B translation model's encoder by inserting adapter layers in each encoder layer. Meanwhile, for xlsr-based translation models, we fine-tune all the encoder parameters. Hence, it's natural to ask whether the gains in cross-lingual task transfer from high to low-resource translation tasks come from using adapters during multi-task fine-tuning or by the semantic KD step. Table 6 compares the translation model's performance when using Adapter-based fine-tuning vs. fine-tuning all the encoder parameters. Adapter-based multi-task translation fine-tuning of xlsr-0.3B encoder (xlsr-0.3B-A in table) slightly decreases the performance on high-resource translation tasks and slightly increases on low-resource languages. Although the TRFGap is reduced slightly, it is still substantially larger than the adapter-based fine-tuning of samu-xslr-0.3B (samu-xlsr-0.3B-A). Interestingly, full fine-tuning of samu-xlsr-0.3B encoder (samu-xlsr-0.3B-F in table) has a drastically larger TRFGap than adapter-based fine-tuning. This result implies that preserving semantic knowledge acquired by the samu-xlsr-0.3B encoder due to the semantic KD step is essential for excellent cross-lingual transfer from high- to low-resource languages.

Model	High	Mid	Low	TRFGap
xlsr-0.3B-F	30.6	18.9	5.1	25.1
xlsr-0.3B-A	28.6	17.9	7.2	21.4
samu-xlsr-0.3B-F	32.4	18.1	8.2	24.2
samu-xlsr-0.3B-A	**34.4**	**31.1**	**20.3**	**14.1**

Table 6. Ablation II: Adapter vs. full encoder fine-tuning.

Zero-Shot Translation Results. For zero-shot translation, we train the translation models on the four high-resource translation tasks out of the 21 X→English translation tasks in the CoVoST-2 benchmark. The translation models do not see training data for the five mid and 12 low-resource tasks. We evaluate the X→EN translation models on all three task groups to test for zero-shot cross-lingual transfer capability of samu-xlsr-0.3B translation model from high to mid and low-resource tasks. We observe, in **Table 7**, that samu-xlsr-0.3B, compared to xlsr-0.3B, performs on average 18.8 BLEU points better in the mid-resource and 11.9 BLEU points in the low-resource group. These results strengthen our claims that our three-step cross-lingual TL framework, with the crucial semantic KD step inspired by SAMU-XLS-R framework [3], improves cross-lingual transfer from high to low-resource languages.

Model	High	Mid	Low	TRFGap
xlsr-0.3B	31.0	5.8	0.9	30.1
samu-xlsr-0.3B	**33.6**	**24.6**	**12.8**	**20.8**

Table 7. Zero-shot X→EN translation performance. Mid, and low-resource tasks are unseen during training.

4. CONCLUSIONS

This paper addresses the central question of cross-lingual transfer learning in Natural Language Processing. We focus on the problem of multilingual spoken language translation, which we model using the transformer model. We analyze the impact of different encoder initializations on the downstream translation task performance. We show that by initializing the model's encoder with samu-xlsr-0.3B that is trained using the recently introduced semantic knowledge distillation framework SAMU-XLS-R presented in [3], we achieve significantly better cross-lingual transfer from high to low resource languages in the X→English translation tasks in the CoVoST-2 speech-to-text translation benchmark.

5. REFERENCES

[1] Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli, "Xls-r: Self-supervised cross-lingual speech representation learning at scale," *arXiv:2111.09296*, 2021.

[2] Changhan Wang, Juan Pino, Anne Wu, and Jiatao Gu, "Covost: A diverse multilingual speech-to-text translation corpus," *arXiv:2002.01320*, 2020.

[3] Sameer Khurana, Antoine Laurent, and James Glass, "Samu-xlsr: Semantically-aligned multimodal utterance-level cross-lingual speech representation," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–13, 2022.

[4] Jonas Pfeiffer, Ivan Vuli'c, Iryna Gurevych, and Sebastian Ruder, "Mad-x: An adapter-based framework for multi-task cross-lingual transfer," in *EMNLP*, 2020.

[5] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv:2006.11477*, 2020.

[6] Pooyan Safari, Miquel India, and Javier Hernando, "Self-attention encoding and pooling for speaker recognition," *arXiv:2008.01077*, 2020.

[7] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang, "Language-agnostic bert sentence embedding," *arXiv:2007.01852*, 2020.

[8] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber, "Common Voice: A massively-multilingual speech corpus," *arXiv:1912.06670*, 2020.

[9] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech 2015*, 2015, pp. 3586–3589.

[10] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer, "Multilingual denoising pre-training for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020.

[11] Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli, "Multilingual speech translation with efficient finetuning of pretrained models," *arXiv:2010.12829*, 2020.

[12] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton, "Layer normalization," *arXiv:1607.06450*, 2016.

[13] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[14] Ankur Bapna, Colin Cherry, Yu Zhang, Ye Jia, Melvin Johnson, Yong Cheng, Simran Khanuja, Jason Riesa, and Alexis Conneau, "mslam: Massively multilingual joint pre-training for speech and text," *arXiv preprint arXiv:2202.01374*, 2022.

[15] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, June 2006.