Transfer Learning For Spoken Language Processing

by

Sameer Khurana

BNG., Delhi College of Engineering, India (2012)

S.M., University of Edinburgh, Scotland (2015)

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored By: Sameer Khurana
Department of Electrical Engineering and Computer Science
May 17, 2023

Certified By: James R. Glass
Senior Research Scientist, Computer Science and Artificial Intelligence Lab
Thesis Supervisor

Accepted By: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Transfer Learning For Spoken Language Processing

by

Sameer Khurana

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis develops transfer learning paradigms for spoken language processing applications. In particular, we tackle domain adaptation in the context of Automatic Speech Recognition (ASR) and Cross-Lingual Learning in Automatic Speech Translation (AST).

The first part of the thesis develops an algorithm for unsupervised domain adaptation of End-to-End ASR models. In recent years, ASR performance has improved dramatically owing to the availability of large annotated corpora and novel neural network architectures. However, the ASR performance drops considerably when the training data distribution does not match the distribution that the model encounters during deployment (target domain). A straightforward remedy is collecting labeled data in the target domain and re-training the source domain ASR model. However, it is often expensive to collect labeled examples, while unlabeled data is more accessible. Hence, there is a need for unsupervised domain adaptation methods. To that end, we develop a simple but effective adaptation algorithm called the Dropout Uncertainty-Driven Self-Training (DUST). DUST repurposes the classic Self-Training (ST) algorithm to make it suitable for the domain adaptation problem.

The second part of the thesis develops a transformer neural network encoder that embeds speech from several languages into a shared semantically aligned joint speech-text embedding space. To learn the multimodal semantic embedding space, we propose a teacher/student learning framework where we fine-tune a pre-trained multilingual speech encoder (student) using semantic supervision from a pre-trained multilingual semantic text encoder (teacher). We show that by building multilingual speech-to-text translation technology using the semantic representations learned by our speech encoder, we could achieve a significant *zero-shot* cross-lingual task transfer from seen (during training) high-resource spoken languages to unseen (during training) low-resource spoken languages.

Thesis Supervisor: James R. Glass
Title: Department of Electrical Engineering and Computer Science

Acknowledgments

Thanks to my supervisor Jim Glass for allowing me to pursue Ph.D. in his lab, for giving me tremendous freedom to explore topics of my interest, and for being extremely patient.

I am grateful to Yoon Kim and Jacob Andreas for agreeing to be part of my thesis committee and providing helpful feedback.

Thanks to my wife, Tulika Roy Choudhury, for sticking with me through the tough times during this long journey. This Ph.D. thesis would not have been possible without her. And I am grateful to my mother, Sherry Khurana, who has always supported me, and without her support, I would not be here.

I thank my academic collaborators Antoine Laurent, Niko Moritz, Takaaki Hori, and Jonathan Le Roux. Finally, I am grateful to Defence Science and Technology Agency, Singapore, for funding part of my research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This work proposes several methods for tackling the transfer learning problem in speech processing applications. Transfer learning refers to the problem of transferring knowledge gained from solving a source task in a source domain to a different but related target task in a target domain. Several learning frameworks can be seen as an instance of transfer learning. For example, learning from large amounts of unlabeled data to facilitate few-shot learning with labeled examples is an instance of sequential transfer learning, where we first pre-train a model on unlabeled data using self-supervised learning and then fine-tune it on a small quantity of manually annotated labeled data. Famous examples of the above-mentioned sequential transfer learning are BERT (Devlin et al., 2019) in Natural Language Processing and Wav2vec2.0 (Baevski et al., 2020) in Speech Processing (Zoph et al., 2020).

The need for efficient transfer learning is regularly observed when deploying speech processing models in the real world. Often, there is a mismatch between the training (source domain) and the data distribution that the model encounters during deployment (target domain). A straightforward remedy is collecting labeled examples in the target domain to re-train the ASR model. However, collecting manually annotated data is highly time-consuming and expensive. In contrast, collecting unlabeled data is relatively inexpensive. Hence, there is a need for unsupervised domain adaptation methods. The first part of this thesis (Chapters 3-4) focuses on the problem of unsupervised domain adaptation of End-to-End Automatic Speech Recognition models.

We focus on an old algorithm, Self-Training (ST), which has recently seen a resurgence in machine translation (He et al., 2019), speech recognition (Q. Xu et al., 2020; Jacob Kahn, A. Lee, and Hannun, 2020), speech translation (Pino et al., 2020a), and visual object detection. We repurpose ST to make it more suitable for the domain adaptation problem (Chapter 3). Chapter 4 shows an interesting application of our modified ST algorithm for efficiently building ASR models for low-resource languages.

The second part of our work (Chapters 5-6) focuses on cross-lingual transfer learning. Chapter 5 proposes a novel multilingual speech encoder, the Semantically Aligned Multimodal Cross-Lingual Speech Representations (SAMU-XLS-R). SAMU-XLS-R is a joint speech-text embedding learning framework that encodes semantic information in its learned speech representations, unlike other multilingual speech representation learning frameworks (Conneau, Baevski, et al., 2020; Babu et al., 2021; Bapna, Cherry, et al., 2022a), which often encode the less transferable low-level linguistic knowledge. Since semantic knowledge is language agnostic, building multilingual speech processing models on top of learned semantic representations should improve the models' cross-lingual portability. We show (Chapter 6) that building multilingual speech translation models with the SAMU-XLS-R speech encoder leads to better task-specific knowledge transfer from high to low-resource languages than the other non-semantic speech encoders. Thus, we significantly improve multilingual speech-to-text translation on several public benchmarks.

The work presented in this thesis is based on the following **papers**:

- Chapter 3: Sameer Khurana, Niko Moritz, Takaaki Hori, and Jonathan Le Roux. "Unsupervised Domain Adaptation for Speech Recognition via Uncertainty Driven Self-Training." In ICASSP 2021.

- Chapter 4: Sameer Khurana, Antoine Laurent, and James Glass. "Magic Dust for Cross-Lingual Adaptation of Monolingual wav2vec-2.0." In ICASSP 2022.

- Chapter 5: 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, 2022.

- Chapter 6: Sameer Khurana, Antoine Laurent, and James Glass. "Improving Cross-Lingual Transfer in Multilingual Speech Translation." Under Revision (Journal).

The **outline** of the thesis is as follows. Chapter 2 presents some preliminary information that might be useful for understanding the technical details in the subsequent chapters. It also presents statistics on the corpora used in the thesis. Chapter 3 presents Dropout Uncertainty-Driven Self-Training, our algorithm for domain adaptation of speech recognition models. Chapter 4 presents an exciting application of DUST for building speech recognition models for low-resource languages. Chapter 5 presents our multilingual semantic speech encoder SAMU-XLS-R, and Chapter 6 applies SAMU-XLS-R to multilingual speech-to-text translation. Finally, we summarize the key findings and present ideas about future work in Chapter 7.

Chapter 2

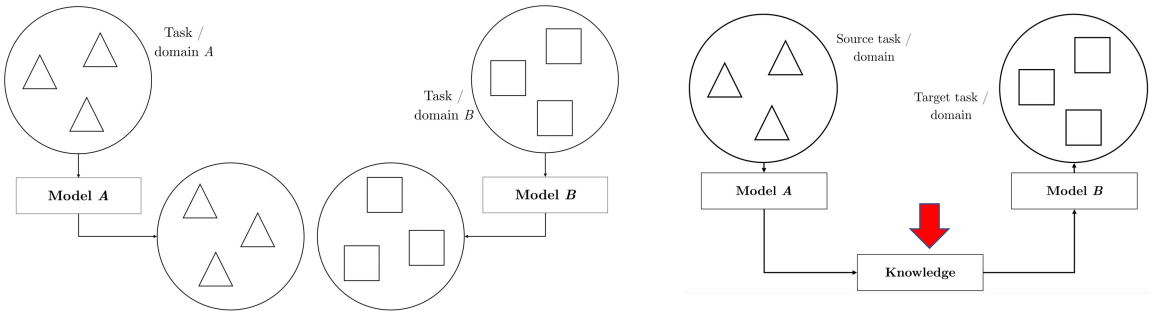
Background

This chapter introduces transfer learning and four transfer learning scenarios that arise in practice. We give examples of popular transfer learning frameworks in speech and natural language processing. We discuss the two-step *sequential transfer learning framework* Wav2Vec-2.0 for speech. Wav2Vec-2.0 and its multilingual extension XLS-R are used throughout this thesis. Also, we give background on different methods of *domain adaptation*. In particular, we focus on data augmentation and self-training methods for domain adaptation, which forms the bulk of our work on this topic. We end the chapter by discussing several datasets used in our work.

2.1 Transfer Learning

This section discusses transfer learning (TL). How does TL differ from traditional Machine Learning (ML)? Suppose we have a learning task A in domain A and task B in domain B. In traditional ML, we collect data for Task A and train a model. We repeat the same process for task B, independent of what was learned in task A. But, in TL, we have a source task in the source domain and a target task in the target domain; the goal is to learn a classifier for the target task in the target domain using *knowledge* acquired by performing the source task in the source domain. Figure 2-1 illustrates the difference between traditional ML and TL. Understanding transfer learning requires us to define the concept of a task and a domain. We follow the

Figure 2-1: An illustration of the difference between traditional machine learning and transfer learning. The key difference is knowledge sharing between the source and target tasks.



survey in Pan and Q. Yang (2010) to explain the necessary concepts.

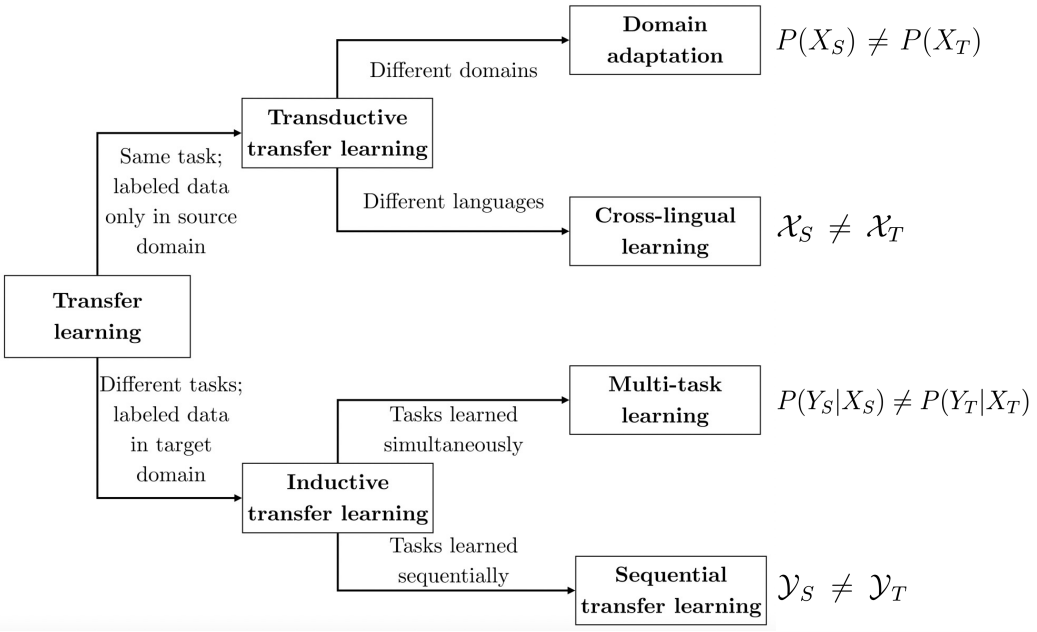
Domain. We define domain as a tuple $\mathcal{D} = \{\mathcal{X}, P_{\mathcal{X}}(X)\}$, where $\mathcal{X}$ denotes the feature space, $P_{\mathcal{X}}$ denotes the probability over the feature space, and $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ is the set of observed samples $\mathbf{x}_i \in \mathcal{X}$. For example, in our case, $\mathbf{x}_i$ is the feature sequence representing a speech waveform.

Task. A task is defined as a tuple $\mathcal{T} = \{\mathcal{Y}, P_{\mathcal{Y}|\mathcal{X}}(Y|X)\}$, where $\mathcal{Y}$ is the label space, and $P_{\mathcal{Y}|\mathcal{X}}$ is the classifier that is learned using paired examples $\mathbf{x}_i \in \mathcal{X}$, $\mathbf{y}_i \in \mathcal{Y}$ using a training set. For example, $\mathbf{y}_i$ could be the text transcript corresponding to a speech waveform $\mathbf{x}_i$.

Transfer Learning. The objective of transfer learning is to build a classifier $P(Y_T|X_T)$ for a target task $\mathcal{T}_T$ in some target domain $\mathcal{D}_T$ using knowledge gained from solving the source task $\mathcal{T}_S$ in a different but related source domain $\mathcal{D}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$. The above inequalities lead to the following four **transfer learning scenarios** as shown in Fig. 2-2 and discussed below.

- $P(X_S) \neq P(X_T)$: The source and the target feature distributions differ. This scenario is commonly referred to as *domain adaptation*. For example, a source speech sample set X_S drawn from recordings of speakers reading audiobooks in English has a different feature distribution from a target sample set X_T drawn

Figure 2-2: An illustration of the four transfer learning scenarios arising from the source and target mismatch, as proposed in Ruder (2019).



from English TED talks or conversations recorded in a meeting room. The first part of our work focuses on the problem of domain adaptation in the context of Automatic Speech Recognition.

- $\mathcal{X}_S \neq \mathcal{X}_T$: The source and the target features are different. For example, speech recordings in different languages. This scenario is referred to as *cross-lingual transfer learning*. For example, consider the practical application of intent classification from speech. Training a classifier $P(Y_S|X_S)$ on labeled data in the source language S and being able to use it for a different target language T is the problem of cross-lingual learning. This transfer is feasible because the label space (intent) is shared across languages.

- $P(Y_S|X_S) \neq P(Y_T|X_T)$: The label distribution is not the same. An example of this scenario is that the speech recordings are from speakers discussing completely different topics. The source recordings could be about business news, while the target recordings are about sports. In such a scenario, the label distributions could vary.

- $\mathcal{Y}_S \neq \mathcal{Y}_T$: The label spaces are not same. For example, if the language of the source and target speech recordings differs, the goal is to generate their corresponding text transcripts. The label spaces are different in this case, but the label distribution is also mismatched. Hence, usually. $\mathcal{Y}_S \neq \mathcal{Y}_T$ implies $P(Y_T|X_T) \neq P(Y_S|X_S)$.

2.2 Transfer Learning Methods

2.2.1 Sequential Transfer Learning

Self-Supervised Learning (**SSL**) is a widely used *sequential transfer learning paradigm* for transfer learning. SSL follows a two-step transfer learning formula.

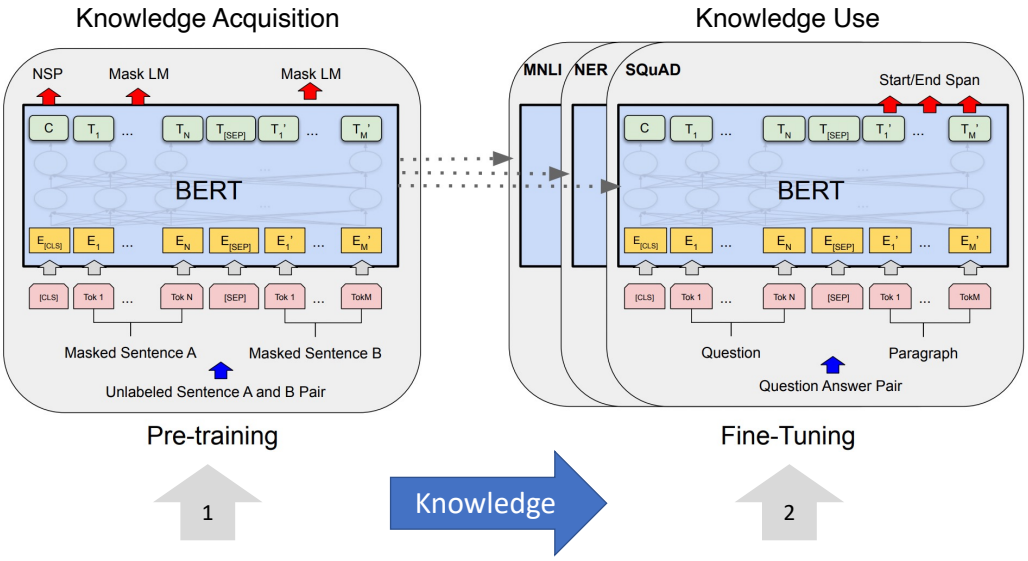
- *Pre-Training* (Knowledge Acquisition): First, a neural network is used to learn abstract representations of our signal of interest, such as speech, text, images, or protein sequences. This step is usually carried out using unlabeled data, but labeled data is sometimes used. The goal is to embed the raw input signal into structured manifolds constructed by several hidden layers of a deep neural network. The structure of these hidden manifolds is a function of the loss functions, datasets, and the neural network architectures used for modeling.

- *Fine-Tuning* (Knowledge use): Second, the pre-trained neural network is fine-tuned on task-specific labeled data. Often, the pre-trained network leads to data-efficient task fine-tuning compared with learning the task from scratch.

Examples of SSL

SSL in Natural Language Processing. A famous example of this two-step sequential TL in the field of *Natural Language Processing* is BERT (Devlin et al., 2019) shown in Figure 2-3. In the pre-training step, a transformer encoder (Vaswani et al., 2017) is trained to predict the identities of the masked tokens in the input text sequence and the following sentence in the document. Then, the same pre-trained

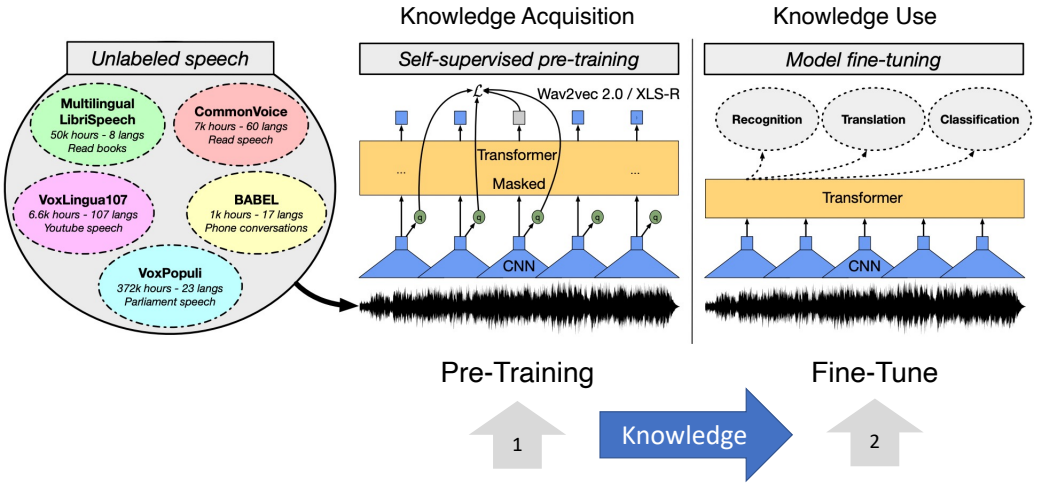
Figure 2-3: BERT: Example of the two-step sequential transfer learning paradigm for text modeling. The first pre-trains a transformer encoder on unlabeled text sentences using masked self-prediction. The second step fine-tunes the encoder on several natural language processing tasks.



transformer encoder is fine-tuned to perform several *Natural Language Understanding* tasks such as named entity recognition, sentiment classification, question answering, etc. Several works extended BERT-style self-supervised learning for languages beyond English (Conneau, Khandelwal, et al., 2019a; Ruder, Søgaard, and Vulić, 2019). Not all SSL frameworks train the transformer encoder using masked self-prediction, such as GPT (Radford and Narasimhan, 2018; Brown et al., 2020) where the encoder is trained using next token prediction.

SSL in Speech. A famous example of self-supervised learning in speech processing is Wav2Vec-2.0 (Baevski et al., 2020) and its multilingual extension XLS-R (Conneau, Baevski, et al., 2020; Babu et al., 2021) shown in Fig. 2-4. Similar to BERT, Wav2Vec-2.0 is a two-step sequential transfer learning paradigm. A transformer encoder is trained using unlabeled speech data in the first step. The pre-trained encoder is then fine-tuned for speech recognition, translation, or classification. The pre-training step in Wav2Vec-2.0 uses only speech data. HuBERT (Hsu, Bolte, et al., 2021), and WavLM (S. Chen et al., 2021) are some other works that are competitive with Wav2Vec-2.0/XLS-R and have a unimodal (speech only) pre-training step. Re-

Figure 2-4: Wav2Vec-2.0 / XLS-R: Example of a popular two-step sequential transfer learning framework in speech. The first step trains a transformer encoder on large amounts of unlabeled speech data collected from several languages. The second step fine-tunes the pre-trained encoder on several downstream tasks, such as speech recognition, translation, and classification.



cently, transfer learning methods that pre-train the transformer encoder using both speech and text data have emerged, such as MSLAM (Bapna, Y.-a. Chung, et al., 2021; Bapna, Cherry, et al., 2022a; Cheng et al., 2022), SpeechT5 (Ao et al., 2021), and MAESTRO (Rosenberg et al., 2022), which perform better than the unimodal models mentioned above.

Other forms of multimodal SSL are the joint audio-visual embedding frameworks such as Davenet (Harwath, Torralba, and J. Glass, 2016; Harwath, Hsu, and J. Glass, 2020) that pre-train a speech encoder using semantic supervision from the visual modality. But, Audio-Visual SSL is not competitive with the other SSL frameworks on any speech processing tasks of interest to us and hence, not considered in this thesis. Below we give some details about the Wav2Vec-2.0 framework.

Wav2Vec-2.0

Introduced in (Baevski et al., 2020), Wav2Vec-2.0 is a Self-Supervised learning framework for training a large speech transformer encoder using unlabeled speech data. The Wav2Vec-2.0 (transformer) encoder consists of a seven-layer CNN that transforms the raw speech waveform $\mathbf{a}_{1:S} \in \mathbb{R}^S$ into an embedding sequence $\mathbf{f}_{1:T} \in \mathbb{R}^{T \times d}$,

which is masked using a two-dimensional masking function similar to SpecAugment (Section 2.2.2) (Park, Chan, et al., 2019) to give the masked sequence $\tilde{\mathbf{f}}_{1:T}$. A multi-layered transformer encoder transforms the masked sequence $\tilde{\mathbf{f}}_{1:T}$ into a contextual embedding sequence $\mathbf{c}_{1:T}$.

The transformer encoder is trained to reconstruct the identities of the masked time steps in its input sequence $\tilde{\mathbf{f}}_{1:T}$. The labels for the masked time steps are derived by quantizing the unmasked sequence $\mathbf{f}_{1:T}$ using an online K-means Vector Quantizer, which outputs a sequence of quantized embeddings $\mathbf{q}_{1:T}$. The contrastive training loss is computed as follows:

$$\mathcal{L}_t = \text{score}(\mathbf{c}_t, \mathbf{q}_t) - \log \sum_{k=1:k\neq t}^{K} \exp(\text{score}(\mathbf{c}_t, \mathbf{q}_k))$$

where $\text{score}(\mathbf{c}_t, \mathbf{q}_t)$ is the cosine similarity between the contextual embedding of the masked time step outputted by the transformer encoder, and $\mathbf{q}_t$ is the quantized embedding of the masked time step in the masked input embedding sequence $\tilde{\mathbf{f}}_{1:T}$ to the transformer.

The Wav2Vec-2.0 encoder trained using about 60K hours of unlabeled speech (J. Kahn et al., 2020) via self-supervised contrastive loss mentioned above is an excellent *few-shot learner* of automatic speech recognition learning task. In (Baevski et al., 2020), a pre-trained Wav2Vec-2.0 fine-tuned using just 10 minutes of transcribed speech can achieve single-digit error rates that previously were possible only by training ASR models using thousands of hours of labeled data on the Librispeech benchmark (Panayotov et al., 2015). Our work uses several variations of Wav2Vec-2.0 listed below.

Wav2Vec-2.0 Base. The smallest of the Wav2Vec-2.0 model series. It consists of 12 transformer layers and 100 million parameters. The model dimension is 768. The encoder is trained on 960 hours of unlabeled English speech collected from the audiobooks corpus Librispeech (Panayotov et al., 2015). The transformer encoder has a similar architecture as the BERT base model (Devlin et al., 2019). The encoder is introduced in (Baevski et al., 2020).

Wav2Vec-2.0 Large. Consists of 24 transformer layers and a model size of 1024. It has 300 million trainable parameters. The encoder is introduced in Baevski et al. (2020). It is trained using 60K hours of unlabeled English speech collected from LibriLight audiobooks corpus (J. Kahn et al., 2020).

Wav2Vec-2.0 Robust. Proposed in Hsu, Sriram, et al. (2021), this encoder has the same architecture as the large Wav2Vec-2.0 encoder. Unlike Wav2Vec-2.0 large, which is trained on the audiobooks domain, the robust Wav2Vec-2.0 is trained using unlabeled English speech collected from several English speech corpora, such as CoVo-English (read noisy speech) (Ardila et al., 2020), SWBD (conversational) (Godfrey, Holliman, and McDaniel, 1992), LibriLight (read audiobooks) (J. Kahn et al., 2020). The aim is to make the pre-trained encoder robust to domain variations.

XLS-R. XLS-R encoder (Babu et al., 2021) has the same architecture as the Wav2Vec-2.0 large. Unlike Wav2Vec-2.0, XLS-R is trained using 400K hours of multilingual unlabeled speech collected from 128 languages. See (Babu et al., 2021) for more details.

All the above pre-trained speech encoders are publicly available[1].

Fine-Tuning of Pre-Trained Model

Above, we discussed the pre-training step in the two-step sequential TL framework. But, research has also focused on determining the best ways to fine-tune the pre-trained models. In particular, Houlsby et al. (2019) proposes an adapter-based fine-tuning method. In this method, a few new task-specific parameters are added to the pre-trained transformer encoder, and during fine-tuning, the new parameters are tuned for the downstream task. And the rest of the model's parameters are kept fixed to their pre-trained values. In Houlsby et al., 2019, an adapter layer is added to each pre-trained BERT transformer encoder block. Two adapter layers are inserted in each block, one after the self-attention module and the other after the feed-forward block. The equations below show the computation that takes place in a single transformer

[1] https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec

encoder block augmented with adapter layers (AdaptL):

$$
\begin{aligned}
x &= \texttt{AdaptL}(\texttt{MHSA}(\texttt{LN}(x))) + x \\
x &= \texttt{AdaptL}(\texttt{FC}(\texttt{LN}(x))) + x
\end{aligned}
$$

where, (MHSA) is the Multi-Headed Self-Attention layer, FC denotes a feed-forward module, LN is the layer normalization layer (J. L. Ba, Kiros, and G. E. Hinton, 2016), and AdaptL is the adapter layer. See (Vaswani et al., 2017) for details about MHSA, and FC modules. The adapter layer is a feed-forward neural network with a single hidden layer. The adapter layer has a bottleneck architecture; the input and output layers have the same size, and the hidden layer is a fraction of the size of the input layer. The adapter layer computation can be described with the following equations:

$$
\begin{aligned}
x_{\text{down}} &= \texttt{ACTFn}(V @ x_{\text{in}} + b) \\
x_{\text{up}} &= U @ x_{\text{down}}
\end{aligned}
$$

where $x_{\text{in}} \in \mathbb{R}^d$ is the input embedding to the adapter layer. $V \in \mathbb{R}^{d \times d/r}$ is the weight matrix that downsamples the embedding to dimension d/r, where r is the downsampling rate. Commonly used values of r are 4, 8, or 16. $U \in \mathbb{R}^{d/r \times d}$ is a weight matrix that upsamples the downsampled embedding to its original size. ACTFn refers to a non-linearity such as ReLU, @ refers to a matrix vector multiplication, and b is the bias vector. Some follow-up works have looked at the placement of adapters Pfeiffer, Vulić, et al., 2020 and fusing multiple parallel adapters (Pfeiffer, Kamath, et al., 2021). This work uses the adapter setup proposed by Houlsby et al. (2019) (explained above) for fine-tuning a pre-trained speech transformer encoder for the task of multilingual speech translation.

2.2.2 Domain Adaptation

We work with the following domain adaptation scenario: labeled data in the source domain and unlabeled data in the target domain for source-to-target domain adap-

Figure 2-5: A clean speech waveform sampled from the Librispeech corpus. We apply different data augmentation methods to this waveform and visualize the impact. Listen to clean speech.

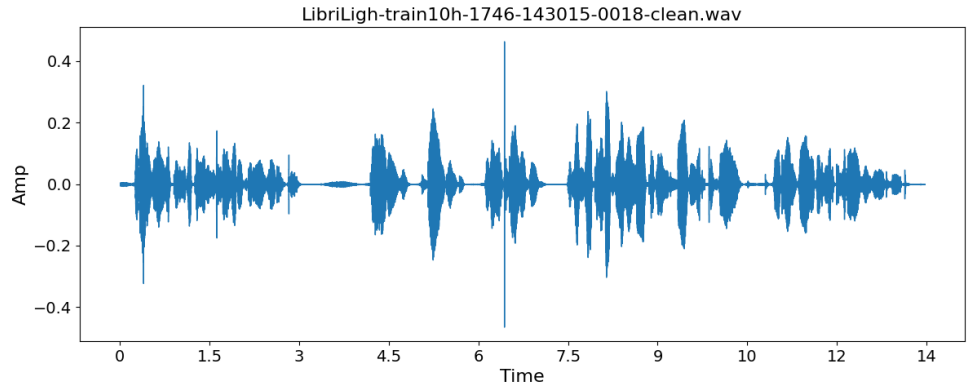
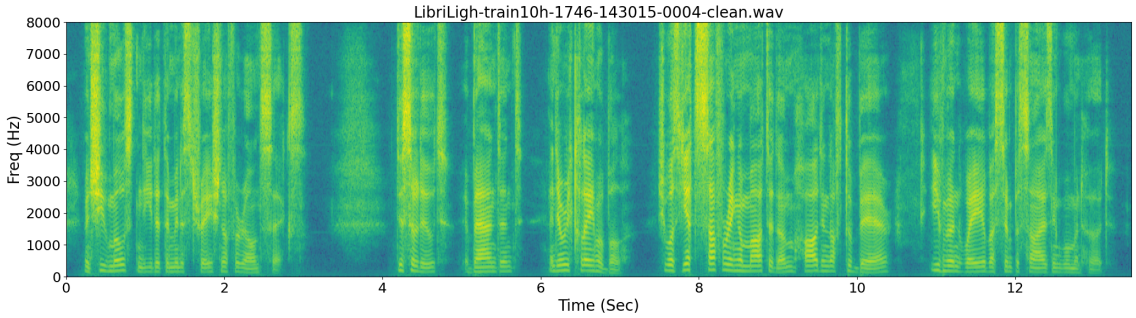


Figure 2-6: A Mel Spectrogram corresponding to a clean speech waveform considered for augmentation. We will observe how this Mel Spectrogram changes after applying different data augmentation methods to the clean speech waveform. Listen to clean speech.



tation. The source and target tasks are the same $\mathcal{T}_S = \mathcal{T}_T$, but the data distributions are mismatched $P(X_S) \neq P(X_T)$. The goal is to build a classifier for the target task in the target domain, using source domain labeled data and target domain unlabeled data. Below, we explain some of the methods commonly used for domain adaptation.

Data Augmentation

A straightforward approach is to train the model on source domain labeled data using strong data augmentation to generalize better to the unseen target domain. Commonly used offline data augmentation strategies for speech are reverberation (Ko, Peddinti, Povey, Michael L. Seltzer, et al., 2017b), speed perturbation (Ko,

Figure 2-7: An example of a Room Impulse Response used to corrupt a clean speech waveform in the Reverb data augmentation method.

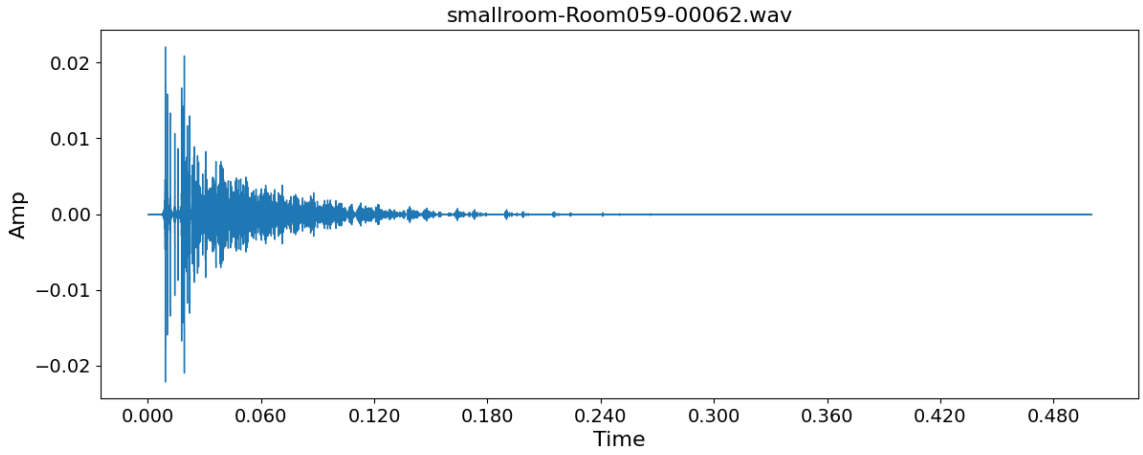
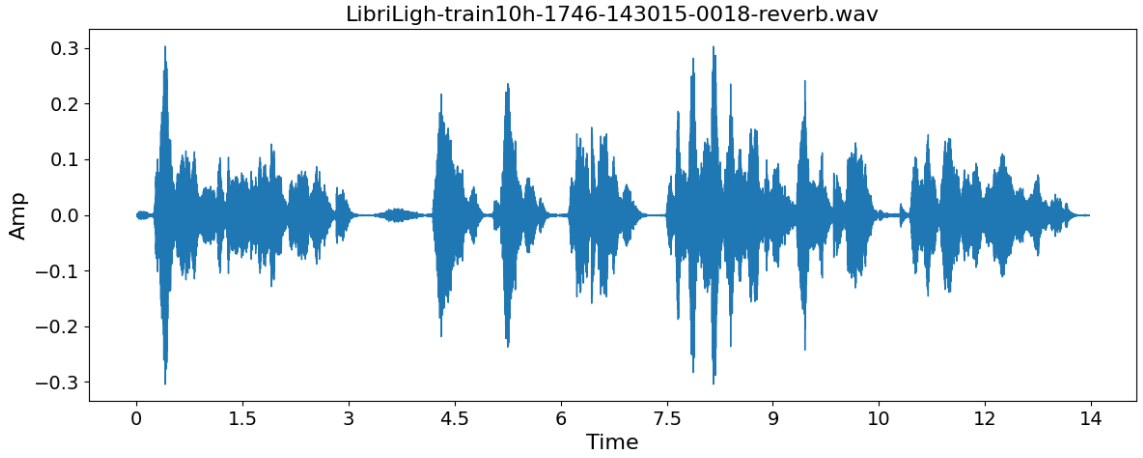


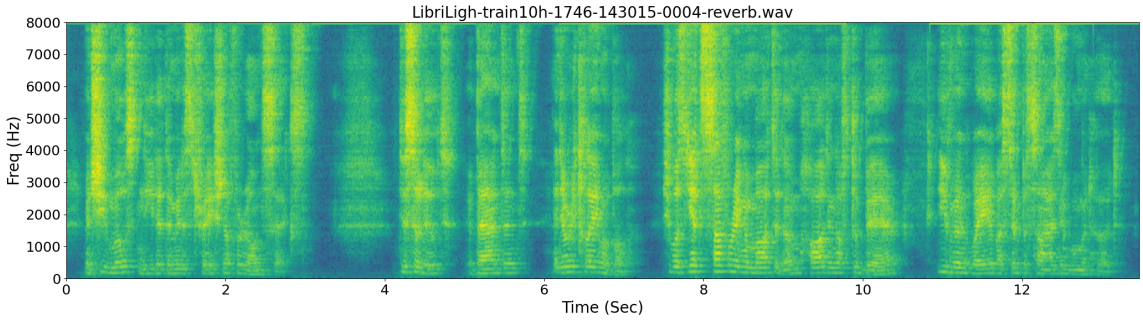
Figure 2-8: A speech waveform corrupted by the Reverb data augmentation method. Listen to the reverberated speech.



Peddinti, Povey, and Khudanpur, 2015), corrupting the speech waveform by adding Gaussian noise, music, speech (Snyder, G. Chen, and Povey, 2015b). An online data augmentation method is SpecAugment (Park, Chan, et al., 2019), where we apply a two-dimensional mask to the speech Mel Spectrogram. Different data augmentation methods are often used in tandem. The training loss for a classifier f trained with data augmentation A is described as follows:

$$L_B = \frac{1}{|B|} L_{\text{task}}(f(A(B)), \mathbf{y}) \tag{2.1}$$

Figure 2-9: An illustration of the Mel Spectrogram of a waveform corrupted by the Reverb data augmentation method. Listen to the reverberated speech.



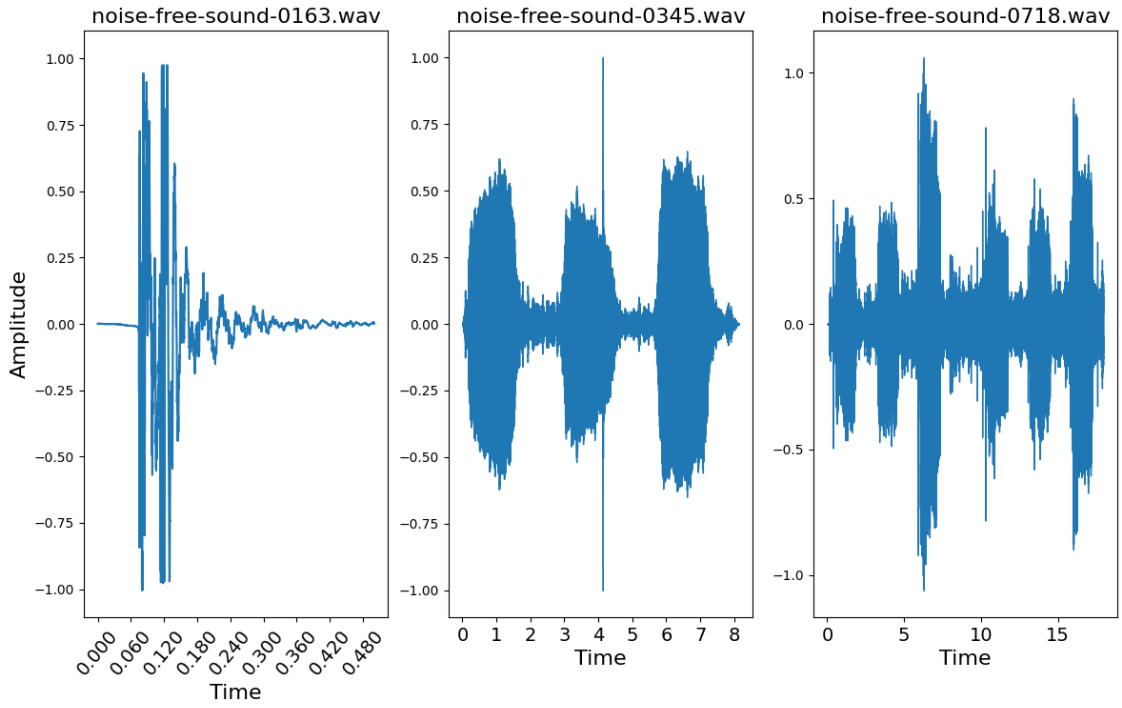
where L_{task} is the task-specific training loss, B is the batch of training examples with labels $\mathbf{y}$. B can be seen as the design matrix. The rows of B correspond to the feature representation of a training example. When modeling speech or text, the feature representation is a two-dimensional sequence of vectors. Below, we list the commonly used speech data augmentation methods.

Speed Perturbation. This augmentation aims to make the speech processing model invariant to different speaking rates that the model might encounter during deployment. Speed perturbation is performed by *resampling* a speech waveform to either increase or decrease its speed. Resampling is a standard operation in any audio processing library such as `sox`[2]. The standard recipe is to create three versions of the clean speech waveform using speed factors of 0.9, 1.0, and 1.1. See Ko, Peddinti, Povey, and Khudanpur (2015) for details on the effect of speed perturbation on the speech recognition model's performance. This augmentation method does not preserve the length of the original waveform.

Reverb. This augmentation aims to make the speech processing model invariant to different rooms in which the speaker might be situated. Different room dimensions induce different reverberation effects in the speech waveform. In this method, to simulate different reverberation effects, we sample impulse responses from the publicly available Room Impulse Response (RIR) dataset[3] introduced in (Ko, Peddinti, Povey, Michael L Seltzer, et al., 2017a) and convolve it with the original speech waveform to

[2] https://sox.sourceforge.net/
[3] https://www.openslr.org/28/

Figure 2-10: Three types of noise signals added to a clean speech waveform in the MUSAN data augmentation method.



generate a reverberated speech waveform. We show an example of an RIR in Fig. 2-7, which we convolve with the speech waveform in Fig 2-5 (Mel Spectrogram in Fig. 2-6) to output the reverberated waveform in Fig. 2-8 (Mel Spectrogram in Fig. 2-9). See Ko, Peddinti, Povey, Michael L Seltzer, et al. (2017a) for more details about this augmentation method and how it impacts speech recognition performance. We use this method in Chapter 3 to improve speech recognition model's generalization to the target domain.

MUSAN. This augmentation method aims to make the speech model robust to background (or foreground) noises such as music, speech (babble), and Gaussian noise (MUSAN). We sample noise from the MUSAN dataset[4] (Snyder, G. Chen, and Povey, 2015a) and add it to the speech waveform according to a pre-defined Signal-to-Noise Ratio (SNR). Fig. 2-10 shows three different noise waveforms sampled from MUSAN. We add the three noise signals to the speech waveform in Fig. 2-5 (Mel Spectrogram in Fig. 2-6) to output the noisy speech waveform in Fig. 2-11 (Mel Spectrogram in

[4] https://www.openslr.org/17/

Figure 2-11: A speech waveform corrupted with three types of foreground noise; music, speech, and gaussian noise. Listen to the noisy speech. Pay attention to three different noises, one at the beginning, one in the middle, and one towards the end.

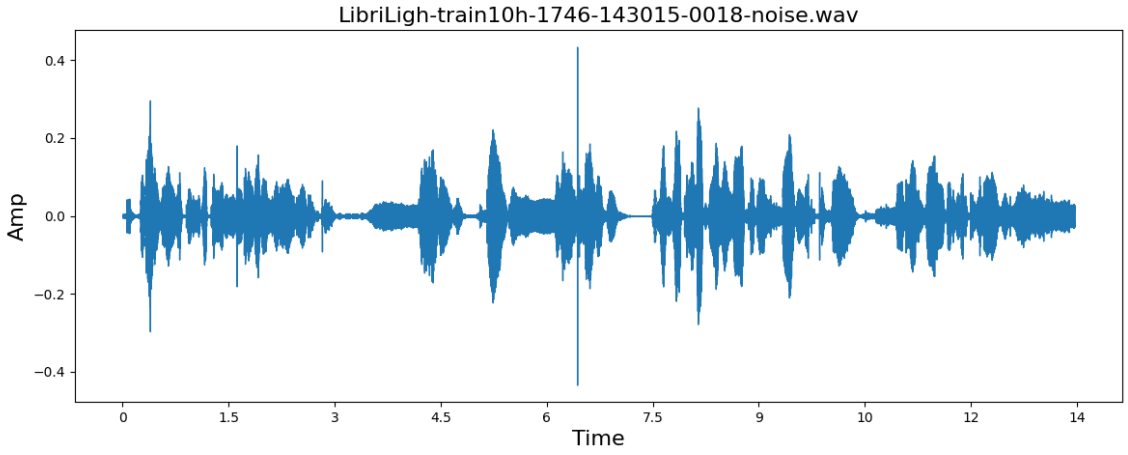


Figure 2-12: An illustration of the Mel Spectrogram of a speech waveform corrupted by music, speech, and gaussian foreground noises. Listen to the noisy speech.

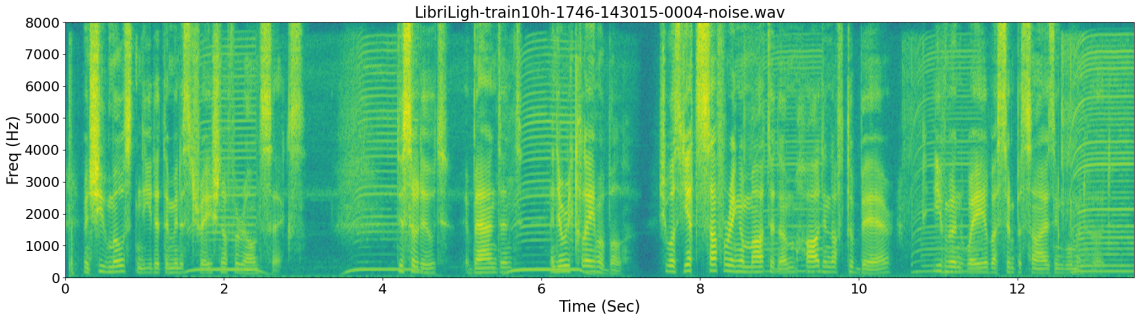
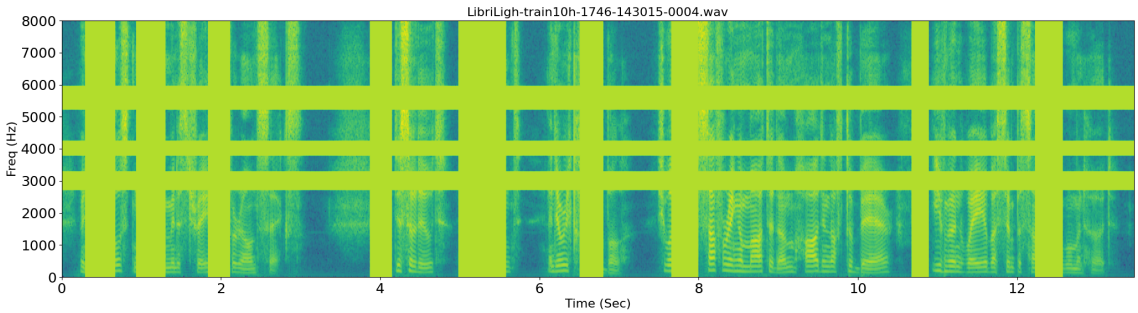


Fig. 2-12). The three noise signals (from left to right) in Fig. 2-10 are added with an SNR of 10, 5, and 15, respectively, and with an offset (start time of addition with the original speech waveform) of 0, 1.4, and 8.3 seconds respectively. We add the noises in the foreground. We use this method and reverberation to improve source-to-target domain generalization of speech recognition models.

SpecAugment. Unlike the other data augmentation methods mentioned above, SpecAugment (Park, Chan, et al., 2019) works on the two-dimensional Mel Spectrogram rather than the raw speech waveform. In SpecAugment, we apply a two-dimensional mask to the Mel Spectrogram as shown in Fig. 2-13. We choose the number of masks and the mask width range for the time and frequency dimensions. Although SpecAugment was introduced for masking the Mel Spectrogram, the same

Figure 2-13: An illustration of the SpecAugment data augmentation method. In SpecAugment, we apply a two-dimensional mask to the Mel Spectrogram.



method can be applied to any two-dimensional embedding sequence. See Baevski et al. (2020) for an example.

Domain Invariant Representations

This approach aims to project the speech signal into a domain-agnostic representation space, i.e., two speech utterances with similar content but drawn from two different data distributions should have a similar representation in the domain-invariant representation space. A popular approach for learning domain invariant representations is domain-adversarial training (Ganin et al., 2016), where the classifier is penalized if the source and target feature distributions are far apart. To understand domain-adversarial learning, consider a neural network classifier composed of a feature extractor neural net f that outputs features h for the input signal x. A classification head g maps h to the output label space. The task is to make f domain invariant. We describe the learning algorithm below:

1. Sample a batch of labeled data from the source domain $B_L \sim \mathcal{L}$, and a batch of unlabeled data from the target domain $B_U \sim \mathcal{U}$

2. Forward pass B_L, and B_U through the neural net feature extractor f:

$$H_U = f(A(B_U))$$
$$H_L = f(A(B_L))$$

H and B are matrices. Each row of H corresponds to a feature vector corresponding to a row of B. B contains raw signal representation, and H contains hidden representations given by the neural net f. A is the data augmentation function.

3. Compute the *domain discrimination* loss as follows:

$$\mathcal{L}_d = \frac{1}{|B_U|} L_{\text{BCE}}(g_d(H_U), \mathbf{y} = \mathbf{0}) + \frac{1}{|B_L|} L_{\text{BCE}}(g_d(H_L), \mathbf{y} = \mathbf{1})$$

g_d is the domain classifier neural network trained to classify examples into the source (label=1) or target domain (label=0). The domain classifier's loss is the standard binary cross-entropy loss (BCE).

4. Compute the training loss for the model $f \cdot g$ as follows:

$$\mathcal{L} = \frac{1}{|B_L|} L_{\text{task}}(g(H_L), \mathbf{y}_L) - \lambda \mathcal{L}_d \tag{2.2}$$

L_{task} is the task-specific loss function, g is the task classification head that maps the hidden representations H_L to the output label space, and $\mathbf{y}_L$ is the ground-truth label vector. λ is the strength of the penalty term and is tuned by hand.

The above algorithm has found its way into some speech recognition research. In particular, (Adams et al., 2019a) treated different languages as domains and attempted to design a language-agnostic feature extractor for speech using the above algorithm.

Self-Training

Self-Training (ST) (Scudder, 1965) is an easy-to-use but quite effective method for domain adaptation, and in general, for semi-supervised learning. As before, the adaptation scenario is we have access to labeled set $\mathcal{L}$ in the source domain and an unlabeled set $\mathcal{U}$ in the target domain. ST is a Teacher-Student learning framework. ST algorithm is sketched below:

1. **Teacher Training:** Train a teacher model f_{teacher} by minimizing a task-specific loss on the labeled set $\mathcal{L}$.

$$\frac{1}{|B|} L_{\text{task}}(f_{\text{teacher}}(A(X), \mathbf{y})) \tag{2.3}$$

where $B = (X, \mathbf{y}) \sim \mathcal{L}$ is a batch of labeled examples $(X, \mathbf{y})$. X is the feature matrix, and $\mathbf{y}$ is the label vector. Each row in the feature matrix corresponds to the feature representation for a particular data point in set $\mathcal{L}$. The length of the label vector is equal to the number of rows in the feature matrix.

2. **Inference:** Generate label for each unlabeled data point $x_u \sim \mathcal{U}$ using the teacher.

$$\hat{y}_u = \texttt{Inference}(f_{\text{teacher}}(x_u)) \tag{2.4}$$

Add $(x_u, \hat{y}_u) \in \mathcal{P}$. `Inference` process is task-specific. E.g., a sequence generation task such as speech recognition would involve using the *beam search* algorithm (Newell, 1973). A multi-label classification task would involve choosing the most likely label (`argmax`) using a class probability vector outputted by the teacher model.

3. **Student Training**: Train a student model f_{student} on combined $\mathcal{L} \cup \mathcal{P}$.

$$\frac{1}{|B|} L_{\text{task}}(f_{\text{student}}(A(X), \mathbf{y})) \tag{2.5}$$

where $B = (X, \mathbf{y}) \sim (\mathcal{L} \cup \mathcal{P})$. L_{task} is the same loss used to train the teacher in step 1.

4. **Iterate**: Go back to step 2, where $f_{\text{teacher}} = f_{\text{student}}$ the student now becomes the teacher.

As the student improves each iteration, the quality of pseudo-labels for the next iteration on the unlabeled data points improves. Hence, the subsequent student becomes better in the target domain. Recently, ST has been used in deep learning to improve object detection (Zoph et al., 2020), text-based machine translation (He

et al., 2019), speech recognition (Q. Xu et al., 2020), and speech translation (Pino et al., 2020b).

Noisy Student-Training

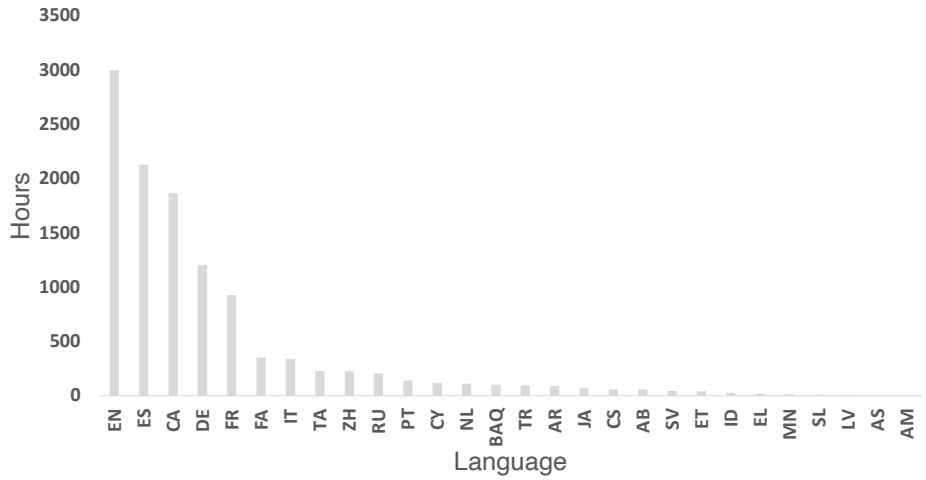
Noisy Student Training (Xie et al., 2020; Park, Zhang, et al., 2020) is similar to self-training, except that noise is injected into the student's training process. The student model is a neural network in our work. Several noise sources during neural network training exist, such as using *dropout* (Srivastava et al., 2014). Also, noise can be injected using different data augmentation techniques mentioned above. Another noise source is the pseudo-label generation process for student training, such as using *beam search* for inference naturally injects some randomness in the pseudo-label generation step. Noise in the self-training process is essential for achieving success. See He et al. (2019) for an in-depth analysis of the different noise sources in self-training and the impact of each on the final student model's performance.

2.2.3 Multi-Task Learning

Multi-task learning (MTL) refers to the learning framework where a model is trained simultaneously on several learning tasks, unlike sequential transfer learning, where we learn several tasks in a sequence. In the context of neural networks, there are two types of MTL. (i) *Hard Parameter Sharing*: Most model parameters are shared across learning tasks, while a few might be task-specific. A typical model specification for multi-task learning is $f_{\text{joint}} \cdot g_{\text{task}}$, a composition of a feature extractor (E.g., a deep neural network) f_{joint} shared across all tasks, and a task-specific classification head g_{task} that maps the shared representation $h_{\text{joint}} = f_{\text{joint}}(x)$ to the task-specific label space. This idea goes back to Caruana, 1997. (ii) *Soft-Parameter Sharing*: In this setup, each task has its model, but the parameters of all the models are encouraged to be close by using, for instance, l_2 regularization as in Duong, Cohn, et al. (2015), or instance norm as in Y. Yang and Hospedales (2016).

MTL (i) *Avoids Overfitting*: Training a model on combined data from several

Figure 2-14: An illustration of the motivation for cross-lingual transfer learning. In the real world, we often have resources for building language technology for high-resource languages, and there is a long tail of low-resource languages for which we have limited resources.



tasks in the hard parameter sharing setup increases the training sample size and naturally leads to a more robust model, acting as a form of data augmentation. (ii) *Learns Robust Representations*: Since the model is trained to perform multiple tasks simultaneously, it is expected to generalize better to novel tasks than a model trained on only one task (Baxter, 2000).

We develop a single speech translation model capable of handling several translation tasks. The translation model is trained using multi-task learning.

2.2.4 Cross-Lingual Learning

Cross-lingual learning forms the majority of this thesis. The motivation for cross-lingual learning is depicted in Fig. 2-14. A scenario that often arises in the real world is we have enough data for building language technology in high-resource languages, and we have a long tail of low-resource languages for which we do not have enough resources. But, all the languages share some common underlying linguistic structures. Cross-lingual learning aims to extract these shared structures, build language technology on top of these common structures, and thus maximize cross-lingual task transfer from high to low-resource languages. A popular example of cross-lingual

learning in speech is the two-step transfer learning formula XLS-R discussed above (Section 2.2.1). In the first step, a neural net encoder is trained using unlabeled speech in several languages. This step aims to learn a shared cross-lingual representation space for speech (ideally, a language-agnostic space). The second step involves building speech technology, such as speech-to-text translation, on top of the acquired shared cross-lingual representation space. Babu et al. (2021) show that this two-step method improves performance on multilingual speech-to-text translation over several previous methods by improving cross-lingual transfer from high to low-resource translation tasks. Our work aims to improve the cross-lingual transfer further by learning a shared semantically aligned cross-lingual representation space in the first step of the two-step method.

2.3 Sequence Generation Tasks

This work focuses on two sequence generation tasks, Automatic Speech Recognition (ASR), and Automatic Speech Translation (AST). Below, we give details about the two tasks.

Automatic Speech Recognition. Automatic Speech Recognition (ASR) is a sequence-to-sequence mapping problem. Given a labeled set $\mathcal{L} = \{\mathbf{x}, \mathbf{y}_i\}_{i=1}^{l}$, The goal is to learn a mapping from input speech waveform $\mathbf{x}$ to its corresponding output text transcript $\mathbf{y}$. The ASR mapping problem consists of a learning and inference step.

The learning phase involves tuning the parameters θ of a classifier $\mathcal{M}_\theta$ to maximize the the conditional probability $p_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}|\mathbf{x}, \mathcal{M}_\theta)$ of observing the transcript $\mathbf{y}$ for a given speech waveform $\mathbf{x}$. Traditionally, the above-mentioned conditional probability has been estimated using a complicated generative modeling framework (Schwartz et al., 1985; L. Rabiner and B. Juang, 1986; Lawrence Rabiner and B.-H. Juang, 1993; K.-F. Lee, 1990; Bellegarda and Nahamoo, 1990; Bahl et al., 1991; Renals et al., 1994; Morgan and Bourlard, 1995; Young, Odell, and Woodland, 1994; Neto et al., 1995; Robinson, Hochberg, and Renals, 1996; Ortmanns and H. Ney, 2000; J. R. Glass,

2003; Mohri, Pereira, and Riley, 2008; Povey, Ghoshal, et al., 2011; Rybach et al., 2011; Swietojanski, Ghoshal, and Renals, 2013; Graves, Jaitly, and A.-r. Mohamed, 2013) due to the intractability of computing $p_{\mathcal{Y}|\mathcal{X}}$ directly. However, recently, due to the advancements in neural network based "end-to-end" learning frameworks for ASR (Graves and Jaitly, 2014; Hannun, Case, et al., 2014; Chorowski et al., 2015a; Maas et al., 2015; Chan et al., 2016; Collobert, Puhrsch, and Synnaeve, 2016; Hannun, 2017; Watanabe, Hori, S. Kim, et al., 2017a; Liptchinsky, Synnaeve, and Collobert, 2017; Chiu et al., 2018; Zeghidour et al., 2018; Sperber et al., 2018; S. Zhou et al., 2018; Watanabe, Hori, Karita, Hayashi, Nishitoba, Unno, Soplin, et al., 2018; Salazar, Kirchhoff, and Z. Huang, 2019; Hannun, A. Lee, et al., 2019; A. Mohamed, Okhonko, and Zettlemoyer, 2019), we can estimate the function $p_{\mathcal{Y}|\mathcal{X}}$ directly in a differentiable manner by using (i) the Connectionist Temporal Classification (CTC) framework of Graves, Fernández, et al. (2006), (ii) the encoder-decoder based Cross-Entropy (CE) minimization framework of Sutskever, Vinyals, and Le (2014) and Vaswani et al. (2017), or (iii) the CTC/CE hybrid learning framework of Watanabe, Hori, S. Kim, et al. (2017b). In the context of end-to-end neural network-based ASR, the learning phase can be formulated as the following optimization problem:

$$\theta^{\star} = \arg\max_{\theta} \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})}[\log p(\mathbf{y}|\mathbf{x}, \mathcal{M}_{\theta})] \tag{2.6}$$

where $p(\mathbf{x})$, and $p(\mathbf{y}|\mathbf{x})$ are the data generating distributions, and $\mathcal{M}_{\theta}$ is the model that forms the conditional probability of observing the sequence $\mathbf{y}$, given the sequence $\mathbf{x}$. An approximate solution of the above optimization problem is rendered by stochastic approximation methods (Robbins and Monro, 1951) such as stochastic gradient descent (SGD) using a labeled set of paired examples $(\mathbf{x}, \mathbf{y})$. Many SGD variants are available to find the optimal model parameters θ, such as Adam (Kingma and J. Ba, 2014).

After estimating the model, we use it for inference. **Inference** can be cast as the

following optimization problem:

$$\mathbf{y}_{\texttt{MPC}} = \arg\max_{\mathbf{y}} \left[\alpha \log p(\mathbf{y}|\mathbf{x}, \mathcal{M}_{\theta^\star}) + \beta \log p(\mathbf{y}|\mathcal{M}_{\texttt{LM}})\right]$$

Where $\theta^\star$ are the optimal model parameters obtained during the learning phase, $\mathcal{M}_{LM}$ is an external language model that is trained on a collection of text. The out of the above optimization is the *most probable configuration* (MPC), $\mathbf{y}_{\texttt{MPC}}$ under the acoustic model $\mathcal{M}_{\theta^\star}$, and an external language model $\mathcal{M}_{\texttt{LM}}$. Due to the independence assumptions inherent in the end-to-end ASR modeling framework, the inference could be cast as a dynamic programming problem that can be solved efficiently using algorithms such as Beam Search. We develop CTC-based ASR models in this thesis.

Automatic Speech Translation. Automatic speech translation (AST) refers to mapping a speech waveform in a source language to its text translation in a target language. Similar to ASR, AST is a sequence-to-sequence mapping problem. Earlier works in AST focused on cascading a speech recognition model that maps the source language speech to its text transcript, followed by a text-to-text translation system that maps the source text transcript to text in the desired target language (Hermann Ney, 1999; Nakamura et al., 2006). Recent works use the neural network-based end-to-end learning framework for AST. Sutskever, Vinyals, and Le (2014) introduced a Long-Short Term Memory Recurrent Neural Network (Hochreiter and Schmidhuber, 1997) encoder-decoder model for end-to-end text-based machine translation, later superseded by the transformer model (Vaswani et al., 2017). See Duong, Anastasopoulos, et al. (2016), Mattia A Di Gangi et al. (2019a), Inaguma et al. (2020), and X. Li et al. (2020b) for recent advancements in AST using end-to-end learning with neural networks.

In this thesis, we use a transformer encoder-decoder model for building speech-to-text translation models. The encoder embeds the raw speech waveform into a contextual embedding sequence, which is used to condition an autoregressive transformer decoder which generates the output text translation sequence. The model is

Table 2.1: Wall Street Journal corpus statistics. The corpus comprises 82 hours of transcribed read speech from Wall Street Journal news articles.

Characteristic		train_si284	test_dev93	test_eval92
Dur. (Hours)	Total	81.5	1.1	0.8
	Male	39.8	0.5	0.5
	Female	41.6	0.5	0.3
#Segments	Total	37416	503	333
	Male	18722	246	210
	Female	18694	257	123
#Speakers	Total	283	10	8
	Male	142	5	5
	Female	141	5	3
Avg. Segment Dur(s)		7.8	7.8	7.6

Table 2.2: Example of English transcripts corresponding to speech segments in the Wall Street Journal transcribed speech corpus.

the airline said the dispute involved minor repairs
basic engineering design and site preparation studies are to start this winter
mr. moreland who is now working through a company called continental trading international limited couldn't be reached for comment
the combined operation will serve more than forty thousand individual and institutional clients

trained using the standard cross-entropy loss (Baum and Wilczek, 1987) with label smoothing (Szegedy et al., 2016b).

2.4 Data

2.4.1 Transcription

Below, we list the transcribed speech datasets used in this thesis.

Table 2.3: TED-LIUM 3 corpus statistics. TED-LIUM 3 comprises around 350 hours of transcribed speech data collected from TED talks.

Characteristic		Train	Dev	Test
Dur. (Hours)	Total	346.2	3.7	3.8
	Male	242.2	2.3	2.4
	Female	104.0	1.4	1.4
#Speakers	Total	1938	16	16
	Male	1303	10	10
	Female	635	6	6
#Talks	Total	2281	16	16

Wall Street Journal

This work uses the Wall Street Journal corpus (WSJ) (Douglas B. Paul and J. M. Baker, 1992) in Chapter 3 as the source domain for the source-to-target domain adaptation of speech recognition models. The domain of the corpus is *read news speech*. Human transcribers are asked to read aloud text collected from wall street journal news articles. WSJ consists of approximately 81 hours of transcribed speech data for ASR model training, one hour for model selection, and 48 minutes for model evaluation. Table 2.1 shows the data distribution of the WSJ corpus. It consists of 37K transcribed speech segments in training, 503 in development, and 333 in the evaluation set. There are 283 speakers in the training set, with an equal representation of male and female speakers. The average duration of a speech segment in the WSJ corpus is 7 to 8 seconds. Table 2.2 shows a sample of text transcripts in the WSJ corpus.

TED-LIUM 3

This work uses TED-LIUM 3 (Hernandez et al., 2018a) corpus in Chapter 3 as one of the adaptation targets for source-to-target domain adaptation of speech recognition models. The domain of the corpus is *oratory speech*, which is different from the WSJ corpus mentioned above. TED consists of 346 hours of training, 4 of development, and

Table 2.4: Example of English transcripts corresponding to speech segments in the TED-LIUM 3 transcribed speech corpus of English TED talks.

well i finally found bill about a block away from our house at this public school playground it was a saturday and he was all by himself just kicking a ball against the side of a wall
our family was too strange and weird for even santa claus to come visit and my poor parents were trying to protect us from the embarrassment this humiliation of rejection by santa who was jolly but let's face it he was also very judgmental
this figure shows you the change in the lake level of lake mead that happened in the last fifteen years you can see starting around the year two thousand the lake level started

4 hours of evaluation transcribed English speech data. It has around 2000 speakers (1.3K males and 635 females). Table 2.3 presents the data distribution. Table 2.4 shows transcripts sampled from the TED-LIUM 3 corpus. In general, the transcript lengths are longer than WSJ. Unlike WSJ, TED transcripts are more colloquial.

Switchboard

This work uses Switchboard (SWBD) (Godfrey, Holliman, and McDaniel, 1992) in Chapter 3 as one of the adaptation targets for source-to-target domain adaptation of speech recognition models. SWBD[5] is an English conversational speech corpus. It comprises 260 hours of transcribed telephone conversations from 543 speakers (302 males, 241 females) for training and a small fraction as a development set. We use the standard 2000 *HUB5 English Evaluation Speech* corpus[6] for model evaluation. Table 2.5 shows some sampled transcripts from the SWBD corpus. Notice the speaking style is more conversational than WSJ or TED corpus.

[5] https://catalog.ldc.upenn.edu/LDC97S62
[6] https://catalog.ldc.upenn.edu/LDC2002S09

Table 2.5: Example of English transcripts corresponding to speech segments in the Switchboard conversational transcribed speech corpus.

is a person of empty promises you can't you know it's hard it's hard to do that so it's hard to change a different- a country especially in the Middle East which has different views on life
to go against the to exactly you can't go after the the leader you can't legally do that i mean then again you know there's people who say war is war everything's fair
yeah it's it's but it's it's it's it's very hard to deal with people that don't believe- don't believe in the same things and don't live the same way that you do you know it's like as we think that

Table 2.6: Example of multilingual transcripts corresponding to speech segments in the CommonVoice multilingual transcribed speech corpus.

he later studied sculpting in marble at pietrasanta in tuscany.
where are they now?
la nouvelle demeure est construite pour un montant de livres.
kurz nachdem die räuber die bank verlassen hatten, wurde die polizei verständigt.
لن أستسلم لأني لدي شيء يستحق أن أكافح من أجله.
astăzi aș dori să subliniez două aspecte.

CommonVoice

We use CommonVoice (CoVo) (Ardila et al., 2020) corpus in Chapter 5. CoVo is a *multilingual* transcribed *read speech* corpus. There are several versions of CoVo. We use Version eight. It comprises 14K hours (10M segments) of transcribed speech from 87 languages (26 language families). See Table 5.1 for the detailed summary statistics of the corpus. Also, Tables 2.15 and 2.16 present language-wise data distribution; we detail the number of training, development, and test transcribed speech segments for each language in the CoVo corpus. Also, we detail the total hours of transcribed speech, the average duration of a speech segment, and the number of speakers for each language. Table 2.17 presents the same statistics per language family. The average speech segment duration is around 5-6 seconds for each language in the corpus, which

Table 2.7: Example of Arabic transcripts corresponding to speech segments in the Multi-Genre Broadcast 2 Arabic transcribed speech corpus.

أهلا بكم مشاهدينا الكرام في حلقة جديدة من برنامج الاقتصاد والناس موضوع حلقتنا لهذا اليوم
يتحدث عن البيع بالتقسيط وهو نوع من أنواع البيوع الآجلة
يتفق بموجبه البائع والمشتري على كيفية سداد أثمان هذه السلع

can be considered relatively short segments compared to other speech corpora. See Table 2.7 for sampled text transcripts from the CoVo corpus. The first two transcripts are in English, followed by French, German, Arabic, and Romanian. Transcripts are smaller than the other datasets mentioned above.

Multi-Genre Broadcast News 2

Multi-Genre Broadcast 2 (MGB-2) (Ali et al., 2019) corpus consists of 1,200 hours of Arabic transcribed speech collected from the Al-Jazeera news channel. Table 2.9 presents the data distribution. For full details, see Ali et al. (2019). Table 2.6 shows some sampled transcripts from the MGB2 corpus. The speech domain is *news broadcast*. The transcript lengths are comparable to TED-LIUM 3, and longer than CoVo transcripts.

2.4.2 Translation

Below, we list the datasets used for building the translation models in this work. All the datasets are freely available.

CoVoST-2

CoVoST-2 (Changhan Wang, Pino, et al., 2020) is a public speech translation benchmark. We use this dataset for developing multilingual speech-to-text translation models. CoVoST consists of two types of translation tasks: (i) X→EN: where X denotes the language of the source speech utterances, and EN denotes the language of target text translations. The translation task is to generate EN text translations

Table 2.8: CommonVoice corpus statistics. CommonVoice is a multilingual transcribed speech corpus. It comprises 14K hours of transcribed speech (9.5M speech segments) in 87 languages (26 language families).

Characteristic		Train	Dev	Test
Dur. (Hours)	Total		14122	
	Male		7196.4	
	Female		2659.6	
	Other		81.1	
	Unknown		4221.6	
#Speakers	Total		207602	
	Male		102762	
	Female		32522	
	Other		2499	
	Unknown		71022	
#Segments	Total	9457044	420042	430676
	Male	4814401	201364	207518
	Female	1776869	80869	82413
	Other	54797	2007	2050
	Unknown	2810975	135802	138695
#Languages	Total		87	
#Lang. Families	Total		26	

corresponding to speech utterances in language X. There are 21 X→EN translation tasks in this benchmark. Table 2.18 presents the data statistics for each of the 21 translation tasks. For each task, we report number of segments in the training, development, and test splits. Also, we report the total number of hours of annotated data available for each task, average duration of a speech segment, and the number of speakers. The average speech segment duration is around 5 to 6 seconds, similar to the CoVo transcribed speech data, since CoVoST is a subset of CoVo. (ii) EN→X: where source speech is in English, and the target text translation is in some language X. The task is to translate EN speech into text in language X. CoVoST has 15 EN→X translation tasks. Table 2.19 shows data statistics for each of the 15 translation tasks.

Table 2.9: MGB-2 corpus data statistics. MGB-2 is an Arabic transcribed speech corpus that consists of around 1200 hours of annotated data.

Characteristic		Train	Dev	Test
Dur. (Hours)	Total	1200	10	10
#Segments	Total	500K	5K	5K

Table 2.10: Example of translation pairs in the CoVoST-2 X→EN translation corpus. We show the transcript corresponding to speech utterances in language X and their corresponding text translation in English.

X	Source(X)	Target(EN)
AR	من بحث وجد.	Anyone search will find.
AR	ما أسهل أن يكتسب المرء عادات سيئة!	Nothing is easier than getting a bad habit!
FR	La famille devra alors tout réapprendre.	So the family will need to relearn everything.
DE	Hast du dir ein Loch in die rote Mütze gerissen?	Did you tear a hole in the red hat?
ID	Aku sedang berbicara dengan muridku.	I'm talking with my student.
CY	Mae hi'n ddeng munud wedi un.	It's ten minutes past one.

Table 2.10 shows some translation pairs sampled from the CoVoST corpus for X→EN translation tasks. The table shows the transcript corresponding to the source speech utterances, and their corresponding EN text translations. Table 2.11 shows sampled translation pairs from EN→X translation tasks in CoVoST-2 corpus.

All state-of-the-art pre-trained multilingual speech encoders, such as XLS-R (Babu et al., 2021) and MSLAM (Bapna, Cherry, et al., 2022b; Bapna, Cherry, et al., 2022a; Rosenberg et al., 2022) evaluate the pre-trained encoder's translation capabilities on the CoVoST-2 benchmark. We show the English transcript for the source speech utterance, and its corresponding text translation in different target languages.

Europarl

Another translation benchmark we use to test our translation models is Europarl (Iranzo-Sánchez et al., 2019), created from the European parliament speech recordings and their corresponding transcripts in several European languages. Europarl consists of 72 translation tasks. There are nine spoken languages. Speech utterances in each language are paired with corresponding text translations in the eight remaining

Table 2.11: Example of translation pairs in the CoVoST-2 EN→X translation corpus. We show the transcript corresponding to English speech utterances and their corresponding text translation in language X.

X	Source(EN)	Target(X)
AR	She'll be all right.	ستكون بخير.
DE	He sat up abruptly.	Er setzte sich schlagartig auf.
FA	He sat up abruptly.	او ناگهان بلند شد.
SV-SE	Man in white shirt standing in a city street.	Man i vit skjorta står på en gata.

Table 2.12: Examples of translation pairs in the Europarl corpus. We show two examples from two of the 72 translation tasks in the Europarl corpus. We show the transcript corresponding to the source speech utterance and its corresponding text translation in the target language.

Source(DE)	Target(PT)
Herr Präsident! Ich begrüße den Ansatz sehr, den wir in dem Weißbuch beschreiben, dass wir den Ursachen von Fehlernährung und Fettleibigkeit und daraus folgenden Krankheiten auf die Spur kommen.	Senhor Presidente, muito me congratulo com a estratégia apresentada no Livro Branco, que nos permitirá tratar as causas de uma nutricão deficiente e da obesidade e doencas associadas.
Source(EN)	Target(PL)
Madam President, are only greed, euphoria and cheap money to be blamed for the whole mess?	Pani przewodniczaca! Czy wina za całe to zamieszanie należy obarczyć tylko chciwość, euforie i dostepność taniego pieniadza?

languages. Tables 2.20, 2.21, and 2.22 present the data statistics for each translation task. Unlike for CoVoST-2 X→EN translation tasks, the average speech segment duration is around 9 to 11 seconds, which is significantly longer than speech segments in the CoVoST-2 corpus. Table 2.12 shows two paired translation examples from two different translation tasks in the Europarl corpus. Notice the length of the translations is significantly longer than the CoVoST dataset. Hence, Europarl could be considered a much harder translation task than CoVo on European languages.

Table 2.13: Languages that exist in the CommonVoice Version 8 multilingual transcribed speech corpus.

Lang Code	Lang	Family	Script
ar	Arabic	Arabic	Arabic
as	Assamese	Indo-Aryan	Assamese
hi	Hindi	Indo-Aryan	Devanagari
mr	Marathi	Indo-Aryan	Devanagari
or	Oriya	Indo-Aryan	Odia
pa-IN	Panjabi	Indo-Aryan	Gurmukhi
ur	Urdu	Indo-Aryan	Arabic
az	Azerbaijani	Turkic	Latin-Cyrillic-Persian
kk	Kazakh	Turkic	Cyrillic
ky	Kyrgyz	Turkic	Kyrgyz
tr	Turkish	Turkic	Latin
tt	Tatar	Turkic	Arabic-Cyrillic
ug	Uighur	Turkic	Arabic
uz	Uzbek	Turkic	Latin-Cyrillic
be	Belarusian	Slavic	Cyrillic
bg	Bulgarian	Slavic	Cyrillic
cs	Czech	Slavic	Latin
mk	Macedonian	Slavic	Cyrillic
pl	Polish	Slavic	Latin
ru	Russian	Slavic	Cyrillic
sk	Slovak	Slavic	Latin
sl	Slovenian	Slavic	Latin
sr	Serbian	Slavic	Cyrillic-Latin
uk	Ukrainian	Slavic	Cyrillic
ca	Catalan	Romance	Latin
es	Spanish	Romance	Latin
fr	French	Romance	Latin
gl	Galician	Romance	Latin
it	Italian	Romance	Latin
pt	Portuguese	Romance	Latin
ro	Romanian	Romance	Latin
cy	Welsh	Celtic	Latin-Welsch
da	Danish	Germanic	Latin
de	German	Germanic	Latin
en	English	Germanic	Latin

Table 2.14: Languages that exist in the CommonVoice Version 8 multilingual transcribed speech corpus.

Lang Code	Lang	Family	Script
fy-NL	Western-Frisian	Germanic	Latin
nl	Dutch	Germanic	Latin
sv-SE	Swedish	Germanic	Latin
el	Greek	Hellenic	Greek
eo	Esperanto	Esperanto	Latin
et	Estonian	Uralic	Latin
fi	Finnish	Uralic	Latin
hu	Hungarian	Uralic	Latin
eu	Basque	Basque	Basque
fa	Farsi	Iranian	Arabic
ga-IE	Irish	Irish	Latin
ha	Hausa	Afro-Asiatic	Latin
hy-AM	Armenian	Armenian	Armenian
id	Indonesian	Malayo-Polyn	Latin
ig	Igbo	Niger-Congo	Latin
rw	Kinyarwanda	Niger-Congo	Latin
sw	Swahili	Niger-Congo	Latin
ja	Japanese	Japonic	Kanji-Kana
ka	Georgian	Kartvelian	Georgian
lt	Lithuanian	Baltic	Latin
lv	Latvian	Baltic	Latin
ml	Malayalam	Dravidian	Malayalam
ta	Tamil	Dravidian	Tamil
mn	Mongolian	Mongolic	Cyrillic
mt	Maltese	Semitic	Latin
th	Thai	Kra-Dai	Thai
vi	Vietnamese	Vietic	Latin
zh-CN	Chinese-Mandarin	Chinese	Chinese
zh-HK	Chinese-HK	Chinese	Chinese
zh-TW	Chinese-TW	Chinese	Chinese

Table 2.15: CommonVoice language-wise corpus statistics. We report the number of transcribed speech segments for each language, the total hours of transcribed speech, the average duration in seconds of a speech segment, and the number of speakers. The rows are arranged in decreasing order of speech segments in the train set.

	#Segments					
Lang Code	Train	Dev	Test	Total (Hrs)	Avg. Dur. (s)	#Spks
en	1497733	16326	16326	2185.8	5.1	79398
rw	1406052	15988	16213	2000.7	5.0	1055
eo	817861	14902	14915	1407.9	6.1	1415
de	714468	16007	16007	1062.8	5.1	16390
be	646332	15803	15801	903.9	4.8	6160
ca	575444	16077	16078	916.8	5.4	6665
fr	564816	16021	16021	826.1	5.0	16082
fa	264722	9728	9728	317.3	4.0	4016
es	258330	15440	15440	404.6	5.0	22741
it	179217	14905	14905	310.6	5.3	6576
ta	103042	11473	11499	217.7	6.2	679
pl	99367	7748	7749	142.2	4.5	3026
th	98442	10769	10769	142.1	4.3	7414
ru	93070	9415	9419	162.6	5.2	2452
sw	81987	8805	8941	146.8	5.3	288
pt	78826	8302	8301	112.0	4.2	2365
cy	76468	5131	5144	116.3	4.8	1695
zh-HK	73570	5563	5563	99.7	4.2	2738
nl	61311	10477	10477	98.0	4.3	1462
zh-TW	61109	4200	4200	62.6	3.2	1695
eu	55727	6463	6463	98.9	5.2	1192
ar	54336	10386	10388	85.2	4.2	1216
uz	50094	10849	11598	81.0	4.0	1355
tr	46787	8110	8339	65.1	3.7	1228
uk	35459	5802	5802	63.4	4.9	684
cs	31535	6950	7267	54.9	4.3	525
ug	30225	2742	2744	59.8	6.0	382
fy-NL	29832	3024	3024	49.6	5.0	1132
sv-SE	27695	4764	4843	40.8	3.9	718
zh-CN	27357	9688	9698	68.0	5.2	4013

Table 2.16: CommonVoice language-wise corpus statistics. We report the number of transcribed speech segments for each language, the total hours of transcribed speech, the average duration in seconds of a speech segment, and the number of speakers. The rows are arranged in decreasing order of speech segments in the train set.

	#Segments					
Lang Code	Train	Dev	Test	Total (Hrs)	Avg. Dur. (s)	#Spks
ky	26266	1613	1613	37.2	4.5	234
ja	22207	4124	4483	40.8	4.8	550
tt	20204	2812	5086	29.2	3.7	206
id	16059	3207	3608	25.8	4.1	394
et	12077	2613	2613	32.4	6.7	723
sk	11422	2290	2217	17.7	4	133
el	10479	1690	1681	15.9	4.1	312
ro	6855	3683	3843	15.8	4.0	332
hu	6736	3865	4020	19.9	4.9	197
sl	6553	1229	1193	9.6	3.9	125
lt	5146	3370	3647	17.4	5.2	249
mn	4469	1829	1882	12.4	5.5	451
hi	4019	2175	2693	11.7	4.7	276
lv	3565	1829	2148	7.1	3.4	115
fi	3530	1430	1739	8.5	4.5	171
ga-IE	3391	512	509	4.3	3.5	153
gl	3120	2240	2258	10.2	4.8	130
mt	3097	1596	1625	8.3	4.7	203
bg	3087	600	1700	8.2	5.5	60
vi	2946	0	1120	4.5	4.0	200
da	2811	1259	1390	6.6	4.4	137
ka	2407	1348	1345	7.6	5.4	127
ha	1941	0	892	3.4	4.3	25
sr	708	572	598	1.5	2.8	51
pa-IN	590	266	360	1.6	4.8	47
or	546	306	213	1.5	5.1	79
as	508	116	294	1.4	5.3	38
hy-AM	500	229	335	1.8	6.1	32
ur	469	341	341	1.3	4.2	48
mr	429	269	306	1.6	5.8	14
kk	406	316	336	1.5	5.0	75
az	39	15	18	0.1	5.4	10

Table 2.17: CommonVoice language-family-wise corpus statistics. We report the number of transcribed speech segments for each language family, the total hours of transcribed speech, the average duration in seconds of a speech segment, and the number of speakers. The rows are arranged in decreasing order of speech segments in the train set.

	#Segments						
Family	Train	Dev	Test	Total (Hrs)	Avg. Dur. (s)	#Spks	#Langs
Germanic	2333850	51857	52067	3443.6	4.6	99237	6
Romance	1666608	76668	76846	2596.1	4.8	54891	7
Niger-Congo	1488039	24793	25154	2147.5	5.6	1344	3
Slavic	927624	50409	51746	1364.1	4.5	13219	10
Esperanto	817861	14902	14915	1407.9	6.1	1415	1
Iranian	264722	9728	9728	317.3	4.0	4016	1
Turkic	174021	26457	29734	273.9	4.6	3490	7
Chinese	162036	19451	19461	230.3	4.2	8446	3
Dravidian	103344	11473	11499	218.0	5.2	689	2
Kra-Dai	98442	10769	10769	142.1	4.3	7414	1
Celtic	76468	5131	5144	116.3	4.8	1695	1
Basque	55727	6463	6463	98.9	5.2	1192	1
Arabic	54336	10386	10388	85.2	4.2	1216	1
Uralic	22343	7908	8372	60.8	5.4	1091	3
Japonic	22207	4124	4483	40.8	4.8	550	1
Malayo-Polyn	16059	3207	3608	25.8	4.1	394	1
Hellenic	10479	1690	1681	15.9	4.1	312	1
Baltic	8711	5199	5795	24.5	4.3	364	2
Indo-Aryan	6561	3473	4207	19.1	5.0	502	6
Mongolic	4469	1829	1882	12.4	5.5	451	1
Irish	3391	512	509	4.3	3.5	153	1
Semitic	3097	1596	1625	8.3	4.7	203	1
Vietic	2946	0	1120	4.5	4.0	200	1
Kartvelian	2407	1348	1345	7.6	5.4	127	1
Afro-Asiatic	1941	0	892	3.4	4.3	25	1
Armenian	500	229	335	1.8	6.1	32	1

Table 2.18: CoVoST-2 X→English speech-to-text translation corpus. We report corpus statistics for each translation task corresponding to each source language X. We present the speech segments in different data splits, the total hours of translated speech, the average duration in seconds of a speech segment, and the number of speakers. Their are 21 translation tasks.

	#Segments					
Source Lang. (X)	Train	Dev	Test	Total (Hrs)	Avg. Dur. (s)	#Spks
fr	207281	14752	14750	309.2	5.2	7420
de	127585	13503	13504	226.0	5.5	6255
es	78958	13203	13204	157.5	5.8	7265
ca	95833	12730	12726	174.7	5.4	3432
it	31637	8877	8892	73.7	5.7	3096
ru	12112	6110	6300	38.7	5.7	455
zh-CN	7085	4842	4898	26.5	5.8	889
pt	9156	3315	4021	20.0	4.5	319
fa	53920	3429	3422	58.8	4.4	2464
et	1782	1576	1568	9.0	6.6	229
mn	2063	1756	1757	8.4	5.5	237
nl	7108	1699	1699	11.2	4.0	597
tr	3966	1624	1629	7.9	4.0	434
ar	2283	1758	1695	5.8	3.7	132
sv-SE	2160	1349	1595	4.3	3.1	94
lv	2337	1125	1629	4.9	3.5	59
sl	1843	509	360	2.9	3.7	31
ta	1358	384	786	3.1	4.4	53
ja	1119	635	684	3.0	4.6	42
id	1243	792	844	3.0	3.8	51
cy	1241	690	690	3.6	5.0	644

Table 2.19: CoVoST-2 English→X speech-to-text translation corpus. We report corpus statistics for each translation task corresponding to each source language X. We present the speech segments in different data splits, the total hours of translated speech, the average duration in seconds of a speech segment, and the number of speakers. There are 15 translation tasks.

	#Segments					
Target Lang. (X)	Train	Dev	Test	Total (Hrs)	Avg. Dur. (s)	#Spks
de	289202	15519	15370	479.8	5.7	23894
zh-CN	289382	15524	15375	480.1	5.7	23894
fa	289367	15531	15374	480.0	5.7	23894
et	287030	15440	15375	476.0	5.7	23894
mn	289402	15531	15362	480.1	5.7	23894
tr	289211	15528	15372	479.8	5.7	23894
ar	289342	15530	15528	480.2	5.7	23894
sv-SE	289301	15526	15368	479.9	5.7	23894
lv	288977	15526	15373	479.4	5.7	23894
sl	289211	15527	15358	479.8	5.7	23894
ta	289395	15531	15375	480.1	5.7	23894
ja	289348	15530	15368	480.0	5.7	23894
id	289398	15530	15372	480.1	5.7	23894

Table 2.20: Europarl X→Y speech-to-text translation corpus. We report corpus statistics for each translation task corresponding to each source language X and target language Y. We present the speech segments in different data splits, the total hours of translated speech, the average duration in seconds of a speech segment, and the number of speakers. There are 72 translation tasks.

		#Segments				
Sorce (X)	Target (Y)	Train	Dev	Test	Total (Hrs)	Avg. Dur. (s)
en	de	32628	1320	1253	83.2	8.4
en	fr	31777	1281	1214	81.4	8.5
en	pt	31750	1294	1262	81.5	8.4
en	es	31607	1272	1267	81.5	8.5
en	nl	31401	1269	1235	80.4	8.5
en	pl	31136	1258	1238	79.6	8.4
en	it	29552	1122	1130	80.0	9.0
en	ro	28598	1070	1095	72.5	8.4
de	en	12904	2603	2631	41.3	8.3
fr	en	12446	1481	1804	39.3	9.1
it	en	11285	1400	1686	45.0	11.1
pl	en	11148	1564	2229	37.8	9.2
fr	nl	8524	1128	1150	27.4	9.2
ro	en	8376	2018	1963	34.6	10.3
fr	pt	8183	1048	1100	26.3	9.2
fr	de	8110	1088	1093	26.1	9.2
fr	es	7857	1072	1098	25.6	9.2
fr	pl	7620	1030	1113	24.9	9.2
de	es	7617	1198	1421	23.8	8.4
de	fr	7443	1167	1401	23.2	8.4
de	nl	7440	1159	1305	22.8	8.3
es	en	7402	1947	1816	31.2	10.1
de	pt	7385	1162	1387	23.1	8.4
de	pl	7351	1159	1376	22.7	8.4
fr	it	7245	1004	1046	24.8	9.6
pl	es	7177	1100	1253	23.9	9.1
pl	de	7117	1085	1283	23.5	9.1
pl	pt	7089	1006	1251	23.4	9.1

Table 2.21: Europarl X→Y speech-to-text translation corpus. We report corpus statistics for each translation task corresponding to each source language X and target language Y. We present the speech segments in different data splits, the total hours of translated speech, the average duration in seconds of a speech segment, and the number of speakers. There are 72 translation tasks.

		#Segments				
Sorce (X)	Target (Y)	Train	Dev	Test	Total (Hrs)	Avg. Dur. (s)
pl	fr	7079	1047	1258	23.5	9.1
de	ro	7065	1031	1233	21.4	8.4
pl	nl	7064	1059	1224	23.2	9.1
fr	ro	6933	839	949	22.1	9.2
de	it	6647	1146	1217	22.1	8.9
it	de	6619	894	922	26.1	11.1
it	es	6614	877	885	26.0	11.2
it	pt	6550	786	871	25.5	11.1
it	fr	6466	845	893	25.4	11.1
pl	it	6454	1076	1178	22.7	9.5
it	pl	6367	831	820	24.9	11.1
it	nl	6296	765	837	24.5	11.1
pl	ro	6166	804	991	19.9	9.1
it	ro	5878	675	742	22.6	11.1
pt	en	4918	1747	2286	26.1	10.5
es	pt	4727	1141	1089	19.6	10.1
es	de	4702	1147	1114	19.6	10.1
es	fr	4673	1115	1082	19.4	10.1
es	nl	4576	1146	1094	19.2	10.1
es	pl	4503	1077	1059	18.8	10.2
es	it	4476	1065	1079	19.1	10.4
ro	de	4291	1185	1231	19.0	10.3
ro	es	4227	1165	1204	18.6	10.3
ro	nl	4182	1123	1210	18.4	10.3
ro	pt	4180	1144	1200	18.5	10.4
es	ro	4156	999	910	17.1	10.1

Table 2.22: Europarl X→Y speech-to-text translation corpus. We report corpus statistics for each translation task corresponding to each source language X and target language Y. We present the speech segments in different data splits, the total hours of translated speech, the average duration in seconds of a speech segment, and the number of speakers. There are 72 translation tasks.

		#Segments				
Sorce (X)	Target (Y)	Train	Dev	Test	Total (Hrs)	Avg. Dur. (s)
ro	fr	4117	1160	1157	18.2	10.3
ro	pl	4037	1156	1164	17.9	10.3
ro	it	3960	1099	1168	18.1	10.6
nl	en	3219	1913	1747	15.2	7.9
pt	fr	3141	1210	1273	16.6	10.5
pt	es	3132	1218	1256	16.5	10.5
pt	de	3124	1233	1271	16.5	10.4
pt	it	3016	1182	1205	16.3	10.8
pt	nl	2986	1107	1228	15.5	10.4
pt	pl	2953	1154	1196	15.6	10.5
pt	ro	2943	1109	1108	15.0	10.4
nl	es	2064	1155	1014	9.6	8.1
nl	de	2057	1192	1063	9.6	8.0
nl	fr	2042	1144	1012	9.3	8.0
nl	pt	1925	1117	942	8.9	8.0
nl	it	1875	977	890	8.9	8.6
nl	pl	1839	1040	967	8.7	8.1
nl	ro	1799	1132	877	8.5	8.0

Chapter 3

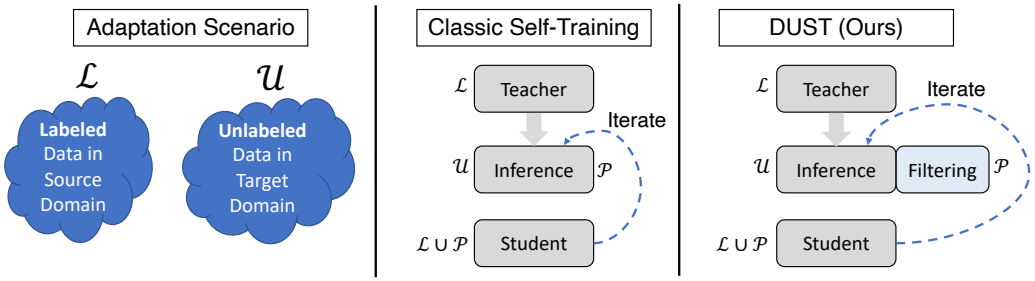
Domain Adaptation of Speech Recognition Models via Dropout-Uncertainty Driven Self-Training

This chapter[1] addresses the transfer learning scenario of source-to-target domain adaptation of End-to-End speech recognition models. The adaptation scenario is that we have labeled (transcribed) speech data $\mathcal{L}$ in a source domain for training an ASR model, and an unlabeled set $\mathcal{U}$ of speech utterances $x_u \in \mathcal{U}$ from the target domain. The goal is to adapt the ASR model trained on the source domain labeled data $\mathcal{L}$ to the target domain by leveraging target domain unlabeled data $\mathcal{U}$. To that end, we propose a Self-Training (ST) algorithm called Dropout-Uncertainty Driven Self-Training (DUST). ST is a pseudo-labeling method that uses a *pseudo-labeling* (*Teacher*) function G_{PL} to label (transcribe) the unlabeled set of speech utterances $x_u \in \mathcal{U}$ in the target domain. The transcripts $\hat{y}_u$ generated for set $\mathcal{U}$ using G_{PL} are known as the pseudo-labels and are denoted by a pseudo-labeled set $(x_u, \hat{y}_u) \in \mathcal{P}$. After pseudo-labeling, we train an ASR model (*Student*) on a combined source domain

[1]The work presented in this chapter is published in Khurana, Moritz, et al. (2021)

labeled and target domain unlabeled set $\mathcal{L} \cup \mathcal{P}$. The initial pseudo-labeler G_{PL} or teacher is an ASR model trained on source domain labeled set $\mathcal{L}$. Since the student is trained on source and target data distributions, it is expected to generalize better to the target domain. ST is an iterative process where the student becomes the pseudo-labeler, and another student ASR model is trained using the new pseudo-labeled set.

Figure 3-1: (Left to Right) Overview of the domain adaptation scenario we tackle in this chapter, the self-training algorithm we use, and our proposed improvement to classic self-training suitable for the domain adaptation problem.



Since, in our case, there is a mismatch between the source and target data distributions, i.e., $P(X_{\text{source}}) \neq P(X_{\text{target}})$, the pseudo-labels generated for the target domain unlabeled data by the teacher that is trained on source domain could be quite erroneous, depending on the severity of the source-target domain mismatch. We address this problem by proposing a Pseudo-Label filtering mechanism, which is the main contribution of this work.

We show the effectiveness of DUST on two domain adaptation scenarios: (i) Wall Street Journal read news speech source to TED-LIUM 3 oratory speech target domain adaptation, and (ii) Wall Street Journal read news speech source to Switchboard conversational speech target domain adaptation. For background on datasets and how they differ, see Section 2.4.1. The problem we tackle in this chapter, the algorithm we use and our contribution is illustrated in Fig. 3-1.

3.1 Introduction

Over the past years, the performance of end-to-end automatic speech recognition (ASR) systems has improved dramatically. This success is driven by improved neural network architectures and training frameworks (Graves, Fernández, et al., 2006; Graves, A. Mohamed, and G. E. Hinton, 2013; Chorowski et al., 2015b; Povey, Peddinti, et al., 2016; Hori, Watanabe, and Hershey, 2017), increasingly large amounts of labeled data (Panayotov et al., 2015; Ardila et al., 2020), and increased computational resources for training complex models. However, ASR performance degrades significantly when the target domain (testing conditions) does not match the source domain (training data). Domain mismatch between training and testing conditions occurs commonly when ASR systems are deployed in the real world, with several factors contributing to it, such as dialectal and accent variations, speaking style (e.g., conversational vs. read), and difference in acoustic conditions (e.g., noisy vs. clean). A straightforward approach to remedy this problem is collecting labeled data in the target domain and using it to adapt a pre-trained source model. However, manually annotating large amounts of data for every new target domain is expensive and time-consuming. Thus, there is a need for unsupervised adaptation algorithms that can leverage unlabeled data for source to target domain adaptation (Bell et al., 2020).

Earlier works on domain adaptation for speech recognition explore the framework of knowledge distillation (G. Hinton, Vinyals, and Dean, 2015), also known as the Teacher/Student (T/S) framework. The T/S framework applied to ASR (J. Li et al., 2014) can be summarized as follows. Given $\mathcal{D}_s$ and $\mathcal{D}_t$, denoting the source and target data distributions, the teacher model is trained on labeled data $\{x_i^s, y_i\}$: $x_i^s \sim \mathcal{D}_s$. Then the student model is trained on the parallel source and target data, $\{x_i^s, x_i^t\}_{i=1}^{N'}$: $x^s \sim \mathcal{D}_s, x^t \sim \mathcal{D}_t$, to minimize the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) between its output senone posterior distribution on x^t to that of the teacher's on x^s. T/S training requires parallel speech data, which can be easily simulated in some cases, such as adding noise to clean speech. Still, it is not clear how to design data transformations from $\mathcal{D}_s$ to $\mathcal{D}_t$ in other scenarios such

as formal speech to dialectal or accented speech, adult speech to child's speech, read speech to conversational speech and so on.

Recently, distribution alignment methods that do not require access to parallel data have become popular for unsupervised domain adaptation. These methods attempt to align the source and target data distributions. Some alignment tools that have shown promise are optimal transport (Courty et al., 2015), domain-adversarial training with gradient reversal layer (GRL) (Ganin et al., 2016), and training using discrepancy losses (Saito et al., 2018). Domain adversarial training uses two players to align the source and target distributions: domain classifier and feature extractor. The features from the feature extractor are shared between the task-specific and domain classifiers. The domain classifier is a binary classifier trained to classify the data sample as a source or target. The feature extractor attempts to align the source and target distributions to fool the discriminator. Apart from the adversarial framework's optimization challenges, this framework does not consider task-specific decision boundaries when aligning distributions, although an improvement is suggested in (Saito et al., 2018). Domain adversarial learning with GRL is used for ASR in (Adams et al., 2019b; Sun et al., 2018).

We focus on self-training (ST) (Scudder, 1965) for unsupervised domain adaptation. ST proceeds by training a teacher model on the labeled source domain data, which generates pseudo-labels for the unlabeled target domain data to obtain pseudo-parallel data. A student model is then trained on the augmented training data, including labeled and pseudo-parallel data, to obtain a model expected to generalize better to the target domain. ST has recently shown excellent performance for neural sequence generation tasks such as machine translation (He et al., 2019) and ASR (Hsu, A. Lee, et al., 2020; Weninger et al., 2020; Moritz, Hori, and Le Roux, 2020), achieving state-of-the-art performance for semi-supervised ASR when applied in an iterative manner (Q. Xu et al., 2020). Classical works in ST (Nigam et al., 2000; Blum and Chawla, 2001; Z.-H. Zhou and M. Li, 2005) suggest that its performance is unstable if the generated pseudo-labels are highly erroneous. Hence, ST is often accompanied by a filtering process to remove such pseudo-labeled utterances from

the training data. However, recent work on ST in the context of neural networks has shown strong results with no filtering at all (Q. Xu et al., 2020). We hypothesize that this is due to two key assumptions made in the work of (Q. Xu et al., 2020):

- No mismatch between the source and target domain. Hence, the teacher model trained with labeled source domain data can generate relatively clean pseudo-labels for the unlabeled target domain data. In (Q. Xu et al., 2020), the labeled and unlabeled data are sampled from the same domain with no mismatch. The labeled and unlabeled data came from the LibriLight audiobooks corpus (J. Kahn et al., 2020).

- Access to large amounts of in-domain text data is used to build a robust language model (LM) for beam search decoding to generate the pseudo-labels for student model training. In-domain language models could have a significant impact on generating clean pseudo-labels.

This work considers where these two assumptions do not hold: we focus on a domain mismatch between the source and target data sets with access to ground-truth labels for the source domain only. In this case, the pseudo-labels generated by the teacher model for the unlabeled target domain data may be less accurate, which increases the need to apply a pseudo-label filtering strategy. To that end, we propose dropout-based uncertainty-driven self-training (DUST), which filters pseudo-labeled data based on the model's uncertainty about its prediction as measured using the degree of agreement between multiple transcriptions obtained with various realizations of dropout and a reference transcription obtained without dropout (Gal and Ghahramani, 2016; Vyas et al., 2019).

We make the following **contributions**. We show that DUST is an effective method for mismatched domain adaptation and substantially improves over the baseline model, which is trained on the source domain labeled data only, as well as over iterative ST without filtering (Q. Xu et al., 2020), whereby the largest gain is observed when the source and target domain mismatch is most severe. In addition, DUST leads to faster and more efficient training than iterative ST since the filtering

process selects only a fraction of the whole unlabeled data set with reliable pseudo-labels. Finally, we perform a preliminary study showing that DUST can be combined with a self-supervised representation learning approach for low-resource conditions.

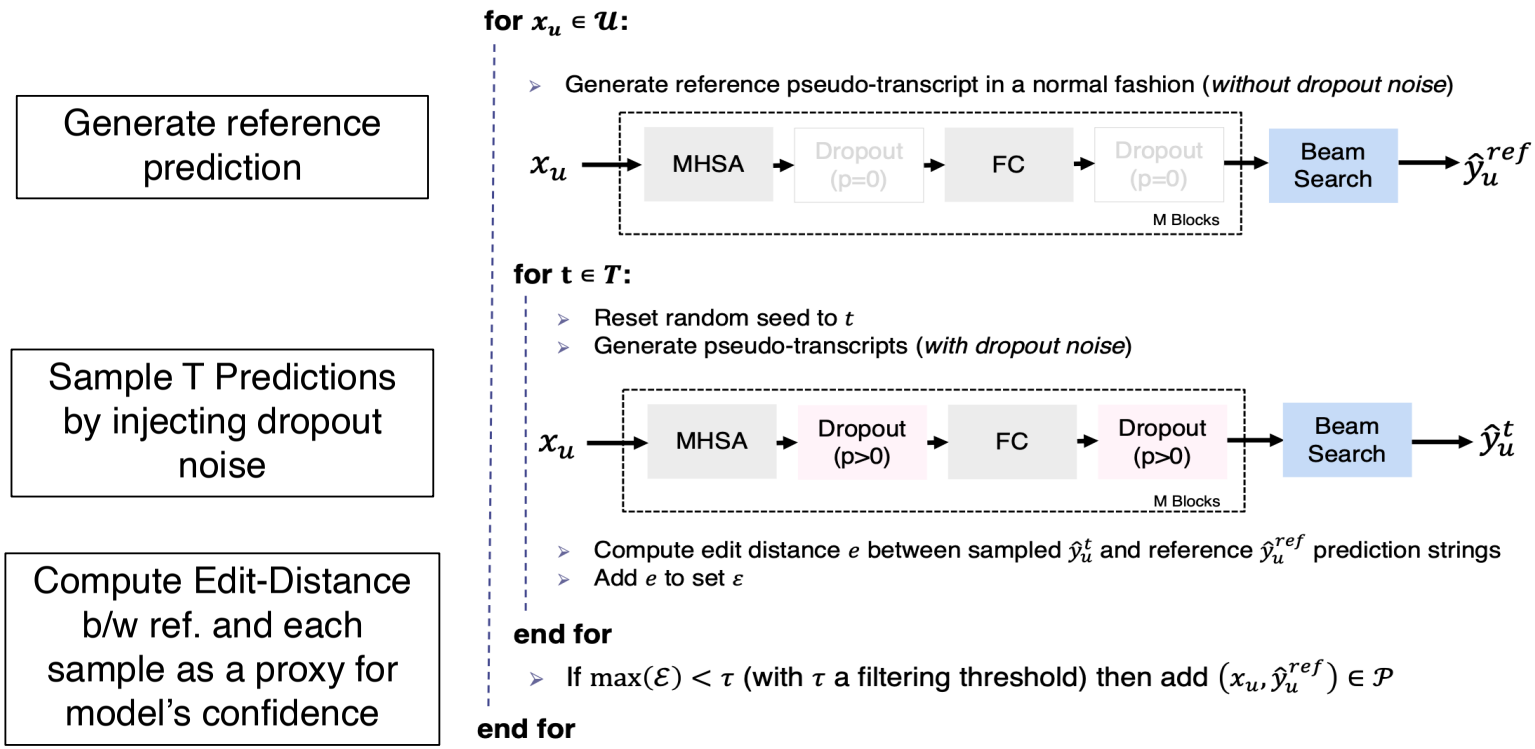
3.2 Method

3.2.1 Using dropout to measure model's uncertainty

DUST uses the model's uncertainty about its predictions $\hat{y}_u$ for an unlabeled target data point x_u to weed out the pseudo-labeled pair $\{x_u, \hat{y}_u\}$, if the model's uncertainty is high. Assuming the model involves dropout layers (Srivastava et al., 2014), uncertainty can be quantified by sampling multiple predictions from the model using dropout and computing agreement between the sampled predictions and a reference prediction obtained without dropout, with low agreement corresponding to high uncertainty. Intuitively, this filtering process can be understood as polling multiple experts to predict an unlabeled data point. If the experts' predictions agree on a particular data point, it will likely be correct. Formally, the method can be understood using the work of Gal and Ghahramani (2016), which connected Bayesian probability theory and neural networks trained with dropout. In particular, they show that a model's predictive variance approximately equals the sample variance of multiple stochastic passes through the network. Here, a stochastic pass refers to inference with a dropout realization. This technique is closely related to (Vyas et al., 2019), which uses a model's prediction uncertainty computed using dropout to estimate word error rates. DUST combines ST and pseudo-label filtering based on the ASR model's uncertainty for an unlabeled speech utterance using dropout.

A pedagogical illustration of how we use dropout to compute the model's predictive uncertainty and weed out noisy pseudo-labels is shown in Fig. 3-3.

Figure 3-3: An illustration of our proposed pseudo-label filtering algorithm. For each unlabeled speech utterance x_u, we generate four hypotheses using the teacher ASR model. We generate a deterministic sample $\hat{y}_u^{\text{ref}}$ using beam search on the probabilities outputted by the acoustic model (transformer encoder). Then, we generate T stochastic samples from the model by injecting noise into the acoustic model. Noise is injected in the form of *dropout*. While generating the sample t $\hat{y}_u^t$, we remove a fraction p (dropout probability) of connections between the neurons in the transformer acoustic model. The fraction p is fixed for all T samples, but different connections are removed for generating each sample. This is akin to generating predictions from different acoustic models. Finally, we compute the edit distances between the T sampled and the one reference prediction. Suppose the edit distances (value between 0 and 1) are less than a pre-defined threshold (such as 0.3). In that case, we accept the pseudo-labeled pair $(x_u, \hat{y}_u^{\text{ref}})$ and add it to the pseudo-labeled set for the next iteration of student acoustic model training.



3.2.2 Self-training with DUST

The overall DUST procedure is summarized in Algorithm 1. We assume that we have access to a set of labeled parallel data $\mathcal{L} = \{x_i, y_i\}_{i=1}^{L}$ in a source domain and a set of unlabeled data $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$ in a target domain, with potentially a strong mismatch between the two domains. DUST proceeds by training a base model f_θ^p on the labeled data $\mathcal{L}$ with dropout layers, using a dropout probability $p \in [0, 1]$. This base model is then used to provide predictions on the unlabeled data $\mathcal{U}$ to generate pseudo-parallel data, of which only a subset $\mathcal{P}$ is selected based on the model's uncertainty on each unlabeled data point, as described further below. Once the subset $\mathcal{P}$ has been determined, a new model is trained on the labeled data $\mathcal{L}$ augmented with the subset $\mathcal{P}$ of pseudo-parallel data. The procedure can be reiterated, with the newly trained model used as the base model.

An unlabeled data point x_u is considered for inclusion in the subset $\mathcal{P}$ of selected pseudo-parallel data as follows. 1) First, a reference hypothesis $\hat{y}_u^{\text{ref}}$ for x_u is generated using the model with disabled dropout layers, resulting in a deterministic inference process which we refer to as deterministic forward pass. 2) Second, multiple hypotheses $\hat{y}_u^t$ are sampled from the model by running it T times with dropout using different random seeds in $\mathcal{T}$, a process we refer to as stochastic forward pass. 3) Finally, the Levenshtein edit distance (Levenshtein, 1966) between each of the T sampled hypotheses, and the reference hypothesis is computed, leading to a set $\mathcal{E}$ of T distances. The edit distance is normalized by the length of the reference hypothesis. If all the values in $\mathcal{E}$ are below a pre-defined threshold ratio τ of the length $|\hat{y}_u^{\text{ref}}|$ of the reference hypothesis, then we add the pseudo-labeled data point $\{x_u, \hat{y}_u^{\text{ref}}\}$ to $\mathcal{P}$, otherwise we reject it. By setting the filtering threshold low, we can accept only pseudo-labeled data points on which stochastic samples have a high agreement, which implies low sample variance and, in turn, low model predictive uncertainty (Gal and Ghahramani, 2016). Our working hypothesis is that data points on which the model has a low predictive uncertainty should be good enough for self-training. We empirically show that low thresholds weed out the noisy pseudo-labels, i.e., inac-

Algorithm 1 Dropout-based Uncertainty-driven Self-Training (DUST)

Given labeled data $\mathcal{L}$ and unlabeled data $\mathcal{U}$
Given a set $\mathcal{T}$ that contains T natural numbers
Train a base model f_θ^p, with dropout p, on labeled data $\mathcal{L}$
repeat
    Let $\mathcal{P}$ be the set of selected pseudo-labeled data points
    Let $\mathcal{E}$ be a set of edit distances
    Initialize $\mathcal{P}$ and $\mathcal{E}$ as empty sets
    for all $x_u \in \mathcal{U}$ **do**
        Compute deterministic forward pass $f_\theta^0(x_u)$
        $\hat{y}_u^{\text{ref}} = \text{beam\_search}(f_\theta^0(x_u))$
        for all $t \in \mathcal{T}$ **do**
            Set random seed to t
            Compute stochastic forward pass $f_\theta^p(x_u)$
            $\hat{y}_u^t = \text{beam\_search}(f_\theta^p(x_u))$
            $e = \text{edit\_distance}(\hat{y}_u^t, \hat{y}_u^{\text{ref}})$
            Add e to the set $\mathcal{E}$
        end for
        if $\max(\mathcal{E}) < \tau|\hat{y}_u^{\text{ref}}|$ (with τ a filtering threshold) **then**
            Add $\{x_u, \hat{y}_u^{\text{ref}}\}$ to the set $\mathcal{P}$
        end if
    end for
    Train a new model f_θ^p on $\mathcal{A} = \mathcal{L} \cup \mathcal{P}$
until convergence or maximum self-training iterations reached

curate pseudo-labels (Section 3.4.1). In practice, running beam search multiple times is computationally expensive, and hence, we only run stochastic beam search $T = 3$ times to draw three samples from the model.

3.3 Experiment Setup

3.3.1 Domain Adaptation Targets

We use the Wall Street Journal (Douglas B Paul and J. Baker, 1992) (WSJ) dataset as our **source domain**, and TED-LIUM 3 (Hernandez et al., 2018b) (TED) as well as Switchboard (Godfrey, Holliman, and McDaniel, 1992) (SWBD) as our **target domains**. WSJ is a read English news speech corpus comprising 80 hours of labeled training data spoken by 280 speakers from different parts of the United States. TED

consists of 350 hours of transcribed English Ted Talks on a wide range of topics by 2,000 speakers worldwide. SWBD consists of 260 hours of two-sided telephone conversations among 543 speakers (302 male, 241 female) from all areas of the United States. The domain mismatch between source and target domains is quite evident (See baseline results in Tables 3.2, and 3.4) and is most severe with the SWBD as the target domain. See Section 2.4.1 for a detailed analysis of the three transcribed speech datasets. The domain mismatch could also be seen from the example transcripts from WSJ, TED, and SWBD in Tables 2.2, 2.4, and 2.5 respectively.

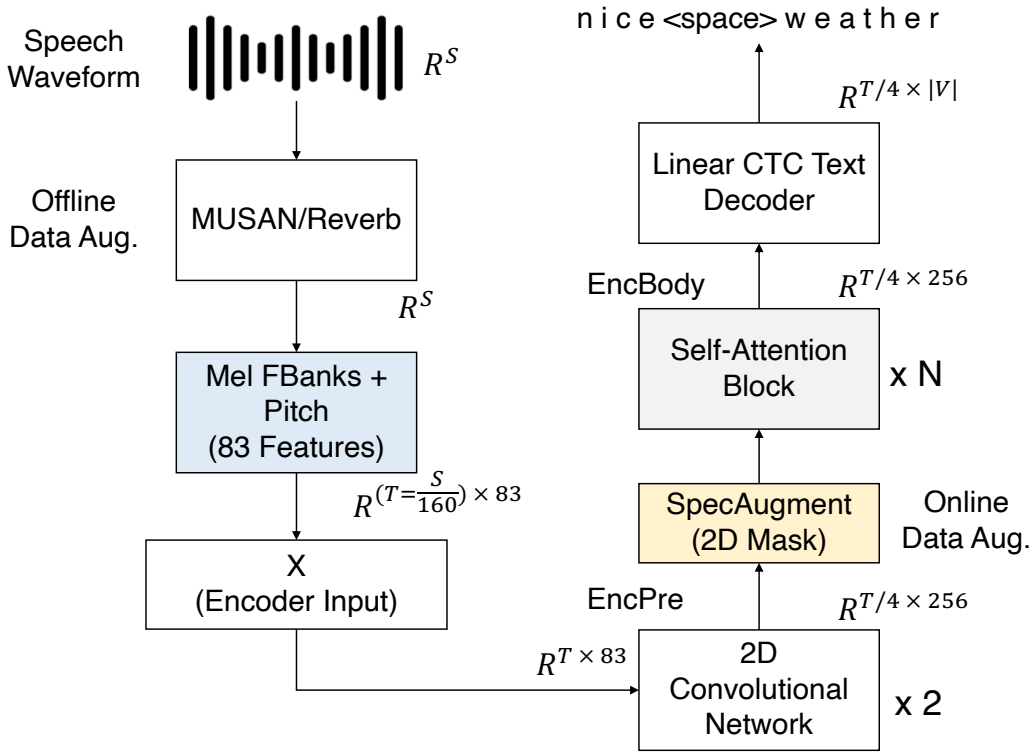
3.3.2 Neural Network Acoustic Model

The neural network model consists of two functions, EncPre($\cdot$), which takes in the input speech sequence and outputs a sub-sampled sequence, and EncBody($\cdot$), which processes the subsampled sequence and outputs the logits for classification (Karita et al., 2019). The input speech is a sequence of 80-dimensional log-mel spectral energies plus three pitch features. EncPre($\cdot$) is a 2-layer Convolutional Neural Network with 256 channels, stride 2, and kernel size 3×3. EncBody($\cdot$) consists of 12 transformer blocks. Each block consists of a self-attention layer followed by two fully connected layers with an interleaved ReLU non-linearity. A dropout layer is applied after self-attention and each fully connected layer. Layer-Norm is used after both self-attention and the two fully connected layers. The number of neurons in the first fully connected layers is 1024. Each self-attention layer consists of 4 attention heads with an attention vector dimension of 256. We set the dropout rate to 0.1 during training and used the same dropout rate when sampling predictions from the model for filtering.

The input to the neural network encoder is the 83 acoustic feature sequence. The feature vector comprises 80 Mel FBanks, and three pitch features. The acoustic features are extracted from the corrupted speech waveform using MUSAN and Reverb data augmentation methods explained in Section 2.2.2. As illustrated in Fig. 3-4, the forward pass of a single speech waveform can be described as follows:

- We start with a speech waveform $\mathbf{a}_{1:S} \in \mathbb{R}^S$, sampled at 16KHz, where S is the

Figure 3-4: An illustration of the ASR model used in this chapter. Both the teacher and student ASR models have this architecture. For better understanding, we separate the model architecture into data processing (left) and neural network (right). The speech waveform is augmented with MUSAN and Reverb data augmentation methods. We extract 80 Mel FBanks + 3 Pitch features from the augmented waveform to get a sequence of acoustic feature vectors. We apply SpecAugment to the acoustic feature sequence. The masked sequence is inputted to the neural network encoder, which consists of a convolutional neural network followed by a stack of Self-Attention transformer blocks. The final layer (Linear CTC) maps the encoder representation to output a character-tokenized transcript. The model is trained using CTC loss.



number of samples.

- We perform MUSAN and Reverb data augmentation (Section 2.2.2) to get the augmented waveform $\tilde{\mathbf{a}}_{1:S} \in \mathbb{R}^S$.

- We transform the augmented waveform $\hat{\mathbf{a}}_{1:S}$ to the acoustic feature sequence $\mathbf{x}_{1:T} \in \mathbb{R}^{T \times 83}$, where $T = S/160$. Each feature vector $\mathbf{x}_t \in \mathbb{R}^{83}$ corresponds to 10ms of the input speech segment.

- The acoustic feature sequence $\mathbf{x}_{1:T} \in \mathbb{R}^{T \times 83}$ is transformed by a two-layered Convolutional Neural Network (CNN) (EncPre$(\cdot)$) that outputs an embedding

sequence $\mathbf{f}_{1:T/4} \in \mathbb{R}^{T/4 \times 256}$. The CNN downsamples the acoustic feature sequence by a factor of four.

- We apply SpecAugment (2D mask) to $\mathbf{f}_{1:T/4}$ to get the masked embedding sequence $\tilde{\mathbf{f}}_{1:T/4}$. We use two frequency and time masks of size 30 and 40, respectively, and perform time warping with a warping factor of five. The above setting of the masking parameters is proposed in the SpecAugment paper (Park, Chan, et al., 2019).

- The masked sequence $\tilde{\mathbf{f}}_{1:T/4}$ is transformed by a stack of Self-Attention transformer blocks (EncBody$(\cdot)$) to a contextual embedding sequence $\mathbf{c}_{1:T/4} \in \mathbb{R}^{T/4 \times 256}$.

- Finally, a linear projection layer maps the contextual embedding sequence $\tilde{\mathbf{c}}_{1:T/4}$ to a distribution over the output character vocabulary $\mathbf{o} \in \mathbb{R}^{T/4 \times |V|}$, where V is the character vocabulary. The output distribution and the character-tokenized ground-truth text transcript are used to compute the CTC loss for training.

Both the teacher and student models have the same architecture.

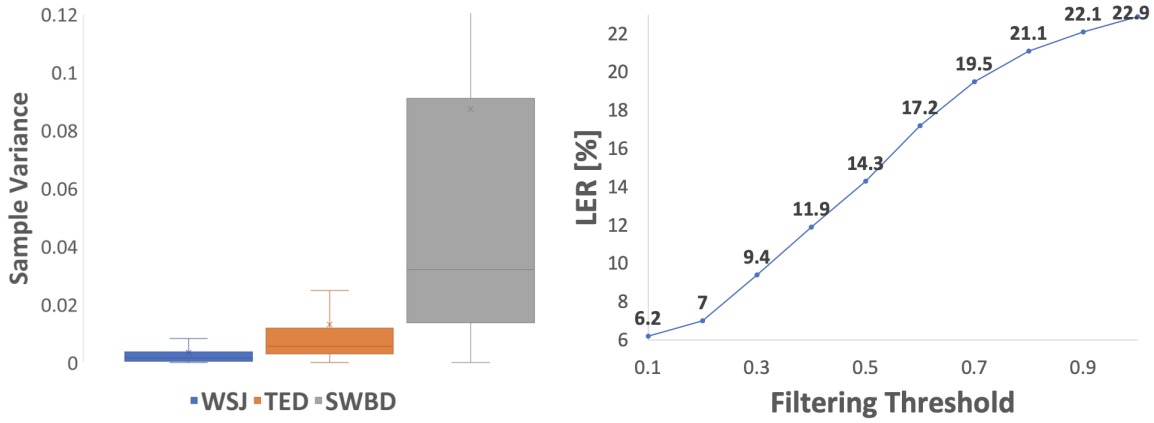
3.3.3 Inference

The inference process can be described below:

$$\hat{y} = \text{argmax}_y \ \log p(y|\mathbf{x}) + \beta \log p(y) \tag{3.1}$$

$p(y|\mathbf{x})$ is output by the neural network acoustic model (AM) described above, and $p(y)$ is an external language model (LM) trained on a text corpus. The scores from the AM and LM are combined to infer the most likely transcript corresponding to an acoustic feature sequence $\mathbf{x}$. This work uses beam search for inference and a 10-gram language model trained using the KenLM language modeling toolkit (Heafield, 2011).

Figure 3-5: (Left) Distribution of the variance of the agreement between stochastic and deterministic samples as a measure of the model's uncertainty on the source (WSJ) and target (TED, SWBD) test data. (Right) Influence of filtering threshold τ on LER [%] of accepted pseudo-labeled utterances for TED.



3.3.4 Hyperparameters

The neural network acoustic model is trained using the Connectionist Temporal Classification (CTC) framework (Graves, Fernández, et al., 2006). The Adam optimizer with a learning scheduler given by (Dong, S. Xu, and B. Xu, 2018, Eq. 10) is used with a learning rate factor of 5.0 and 25k training iterations for warmup. The models are trained for 100 epochs. The final model is obtained by averaging the ten best models with the lowest loss on the validation set. For inference, a beam search decoding algorithm is used with a beam size of 20.

3.4 Evaluation

We show the efficacy of our proposed self-training algorithm DUST on two domain adaptation scenarios; one with a moderate source-target domain mismatch and the other with a severe source-target domain mismatch.

3.4.1 Qualitative Analysis: Pseudo-Label Filtering

In Fig. 3-5, we show the efficacy of our filtering process in weeding out noisy pseudo-labels. The box plot shows the variance of the agreement (normalized edit distance) between the stochastic samples and the deterministic sample on source and target domain data points to measure the model's prediction uncertainty. The source domain is WSJ, and the target domains are TED and SWBD. We train a base ASR model on the source training data and generate ten stochastic samples and a deterministic sample for each utterance in the test sets of the source and target domains via beam search with an LM trained on the source domain text. Then, we compute the edit distance between each sampled hypothesis and the reference hypothesis. Hence, we get a list of ten edit distances for each utterance. We compute the variance of the edit distance list and plot the variance corresponding to each speech utterance in a box plot. Low variance corresponds to the high agreement between the sampled predictions and hence low model predictive uncertainty. We hypothesize that the box plot of variances would have a lower mean for in-domain data and increase as the mismatch between the source and target domains increases. Also, the variance of the box plot would be higher on speech utterances sampled from mismatched target domains (TED and SWBD).

The box plot shows that the model's uncertainty is significantly higher on target domain data than on source domain data, which concurs with our intuition. Furthermore, the line graph shows the relationship between the filtering threshold τ and the label error rate (LER) on the pool $\mathcal{P}$ of accepted pseudo-labeled utterances for TED as the target domain. We see that utterances in the set $\mathcal{P}$ are much cleaner at lower filtering thresholds, as shown by the low LERs.

3.4.2 Topline and Baseline

For all the experiments in subsequent sections, the baseline refers to the model trained on WSJ's labeled source domain training data. Topline refers to models trained on the WSJ training data augmented with labeled data from the target domain. And,

Table 3.1: Results on WSJ source to TED target domain adaptation using a 100k subset of unlabeled target domain data for self-training, investigating different values of filtering threshold τ. An LM trained on source domain text is used during the filtering process and for decoding. #PL[k] is the size of the target domain pseudo-label set PL used in each iteration of student model training. WERR refers to Word Error Rate Recovery ($\frac{\text{topline}-x}{\text{topline}-\text{baseline}}$), where x is the student model's performance in each iteration.

			WER [%]			WERR [%]	
Method	#PL[k]	PL	WSJ/eval92	TED/dev	TED/test	WSJ/eval92	TED/test
Baseline			6.8	37.1	35.0	0	0
Topline			4.6	15.9	14.8	100	100
First Self-Training Iteration							
DUST1 ($\tau=0.1$)	7	14.0	7.4	33.0	30.0	-27.2	24.7
DUST1 ($\tau=0.3$)	38	25.0	**6.1**	**30.0**	**26.8**	**31.8**	**40.6**
DUST1 ($\tau=0.5$)	70	34.0	6.3	31.3	27.6	22.7	36.6
DUST1 ($\tau=0.7$)	90	39.0	6.2	31.1	27.9	27.2	35.1
ST1 (All)	100	42.0	6.3	31.0	27.7	22.7	36.1
Second Self-Training Iteration							
DUST2 ($\tau=0.1$)	35	14.8	6.9	30.1	27.3	-4.54	38.1
DUST2 ($\tau=0.3$)	66	25.0	**6.1**	**28.2**	**24.9**	**31.8**	**50.0**
DUST2 ($\tau=0.5$)	88	33.8	6.6	29.4	25.6	9.10	46.5
DUST2 ($\tau=0.7$)	95	39.0	6.7	29.3	26.1	4.54	44.1
ST2 (All)	100	42.0	6.3	29.4	25.8	22.7	45.5

Self-Training with DUST involves training the model with labeled WSJ data and pseudo-labeled target domain data.

3.4.3 Adaptation Scenario I: WSJ→TED

In Table 3.1, we first compare classic ST, where we use all of the pseudo-labeled data since there is no filtering mechanism in classic ST, and DUST, where filtering is used. We also investigate the effect of different thresholds (τ) on the downstream task performance. This set of experiments is performed using a 100k subset of unlabeled TED target domain data for self-training. We use an LM trained on source domain text (WSJ) during the pseudo-label generation process via beam search. We report both word error rate (WER) and WER recovery rate (WERR). WERR is computed as follows:

$$\frac{\text{topline} - x}{\text{topline} - \text{baseline}}$$

Table 3.2: Results on WSJ source to TED-LIUM target domain adaptation using all unlabeled target domain data for self-training and setting $\tau = 0.3$. An LM trained on source domain text is used during the filtering process. #PL[k] is the size of the target domain pseudo-label set PL used in each iteration of student model training. WERR refers to Word Error Rate Recovery ($\frac{\text{topline}-x}{\text{topline}-\text{baseline}}$), where x is the student model's performance in each iteration.

			WER [%]		WERR [%]	
Method	#PL[k]	PL	WSJ/eval92	TED/test	WSJ/eval92	TED/test
Decoding with an LM trained on source domain text						
Baseline			6.8	35.0	0	0
DUST1	100	25.0	5.9	26.5	40.9	40.3
DUST2	170	25.0	5.7	24.3	50.0	50.7
DUST3	185	24.8	5.7	23.5	50.0	54.5
DUST4	210	25.0	**5.5**	22.4	**59.1**	59.7
DUST5	230	25.4	5.6	**21.1**	54.5	**66.0**
Topline			4.4	13.9	100	100
Decoding with an LM trained on source & target domain text						
Baseline			7.0	33.2	0	0
DUST5	230	25.4	**5.4**	**19.3**	**59.2**	**67.4**
Topline			4.3	12.6	100	100

where x refers to the student model's performance after an iteration of DUST.

We also report the number of selected utterances in PL (#PL) and the WER on these utterances. With a filtering threshold of 0.3, DUST performs slightly better than ST (All), 26.8 % vs. 27.7 %, while using approximately one-third of the pseudo-labeled data (38k vs. 100k). When we compare downstream task performance using different filtering thresholds, we make the following two observations:

- First, the downstream task performance is significantly better than the baseline regardless of the filtering threshold.
- Second, the best results are obtained when the filtering threshold τ is set to mid-range values, with the best setting being 0.3 based on the TED development set.

From here on, we fix the threshold value τ to 0.3. We next perform multiple rounds of self-training using all of the unlabeled target domain data, again using an

LM trained on source domain text during the pseudo-label generation process. The results are shown in Table 3.2. After each iteration, the number of selected utterances in PL increases while their average WER remains roughly constant. Target domain WER shows clear improvements, and after DUST5, we can recover 66% of the WER of the topline when using a decoding LM trained on source domain text and an even slightly higher 67% for a decoding LM trained on both source and target domain text. Beyond five iterations of DUST, we do not see a significant improvement in the student model's performance.

In Table 3.3, we investigate whether an LM is needed for the pseudo-label generation. Our filtering process generates multiple stochastic samples, which requires us to run *beam search* multiple times. This could be an expensive process, especially when we have a large set of unlabeled data, and beam search without an LM for a pseudo-label generation would accelerate the filtering process by a large margin. The results in Table 3.3 show that we can achieve similar or better results without using an LM for pseudo-label generation. In particular, we can recover 80% of WER when decoding using no LM, 79% when using a source domain LM, and 76% when using an LM trained on both source and target domain text. We achieve a WER of 17.6% on the target domain using DUST, which is quite close to the topline WER of 12.6% and outperforms the best WER of 19.3% achieved in Table 3.2, where we used a source domain LM during the pseudo-label generation process. Using a source domain LM for PL generation biases the generated PL towards the source domain text. This hinders generalization, assuming the seed model is good enough without an LM for a sufficient amount of PL utterances to select.

3.4.4 Adaptation Scenario II: WSJ→SWBD

Table 4 shows the results on SWBD as the target domain. As evident from the baseline results, the domain mismatch is quite severe. Nevertheless, DUST can improve over the baseline by 22.4 percentage points (pp) and recover 58.9% of WER when using a source domain decoding LM and 56% when using a decoding LM trained on both source and target domain text. In this case, we could not get good performance

Table 3.3: Results on WSJ source to TED-LIUM target domain adaptation using all unlabeled target domain data, with $\tau = 0.3$ and no LM used during the filtering process. #PL[k] is the size of the target domain pseudo-label set PL used in each iteration of student model training. WERR refers to Word Error Rate Recovery ($\frac{\text{topline}-x}{\text{topline}-\text{baseline}}$), where x is the student model's performance in each iteration.

			WER [%]		WERR [%]	
Method	#PL [k]	PL	WSJ/eval92	TED/test	WSJ/eval92	TED/test
Decoding without LM						
Baseline			15.0	47.9	0	0
DUST1	25	25.0	14.1	37.8	17.6	34.9
DUST2	81	25.0	13.1	31.5	37.2	56.7
DUST3	136	25.8	13.1	28.1	37.2	68.5
DUST4	167	25.0	**12.5**	25.8	**49.0**	76.4
DUST5	178	23.8	12.6	**24.7**	47.0	**80.2**
Topline			9.9	19.0	100	100
Decoding with an LM trained on source domain text						
Baseline			6.8	35.0	0	0
DUST5	178	23.8	**5.7**	**18.4**	**45.8**	**78.6**
Topline			4.4	13.9	100	100
Decoding with an LM trained on source & target domain text						
Baseline			7.0	33.2	0	0
DUST5	178	23.8	**5.6**	**17.6**	**51.8**	**75.7**
Topline			4.3	12.6	100	100

without using an LM during the filtering process, probably due to the severity of the domain mismatch. While the results are encouraging, there is still a large gap between DUST and the topline, which could be addressed by relaxing one of our assumptions regarding not having access to any target domain text data. We leave this investigation for future work.

3.4.5 Self-Supervised Speech Representations and DUST for Low-Resource ASR

Finally, we briefly investigate via a preliminary experiment whether DUST could be effectively combined with a self-supervised representation learning approach for low-resource speech recognition. We train the base source model on just 3 hours of

Table 3.4: Results on WSJ source to SWBD target domain adaptation. An LM trained on source domain text is used during the filtering process. #PL[k] is the size of the target domain pseudo-label set PL used in each iteration of student model training. WERR refers to Word Error Rate Recovery ($\frac{\text{topline}-x}{\text{topline}-\text{baseline}}$), where x is the student model's performance in each iteration.

			WER [%]		WERR [%]
Method	#PL [k]	PL	WSJ/eval92	SWBD/eval2000	SWBD/eval2000
Decoding with an LM trained on source domain text					
Baseline			**6.8**	64.1	0
DUST1	7	33	7.6	50.0	37.1
DUST2	30	35	7.3	47.3	44.2
DUST3	68	35	6.9	44.1	52.6
DUST4	108	34.9	7.0	42.7	56.3
DUST5	125	35.4	7.1	**41.7**	**58.9**
Topline			6.6	26.1	100
Decoding with an LM trained on source & target domain text					
Baseline			7.2	61.7	0
DUST5	125	23.8	**7.1**	**39.9**	**56.2**
Topline			6.6	22.9	100

labeled source data using Wav2Vec (W2V) (Schneider et al., 2019) features as input to the model to remove some of the domain mismatches compared to a baseline trained without Wav2Vec, as shown in Table 3.5. Two DUST iterations using unlabeled source and target domain data significantly improve the performance: DUST improves over the baseline by 31.0 pp and 41.2 pp for the source and target domains, recovering 80% and 50% of the WER, respectively, while Wav2Vec alone only improves by 6.2 pp and 17.4 pp. W2V is trained on Librispeech, and the features are only used to train the base model. We leave a more thorough investigation for future work.

3.5 Chapter Summary

In this chapter, we proposed DUST, a dropout-based uncertainty-driven self-training method for unsupervised domain adaptation. DUST uses only unlabeled speech data from a target domain to transfer a base ASR system trained on a source domain to the target. Unlike classic Self-Training, in DUST, we proposed a pseudo-label

Table 3.5: Results on combination of DUST with Wav2Vec (Schneider et al., 2019) representation learning in low-resource scenario. Baseline and Wav2Vec source models are trained on three hours of data. An LM trained on source domain text is used during the filtering process and for decoding. #PL[k] is the size of the target domain pseudo-label set PL used in each iteration of student model training. WERR refers to Word Error Rate Recovery ($\frac{\text{topline}-x}{\text{topline}-\text{baseline}}$), where x is the student model's performance in each iteration.

			WER [%]		WERR [%]	
Method	#PL [k]	PL	WSJ/eval92	TED/test	WSJ/eval92	TED/test
Baseline			43.2	95.6	0	0
Wav2Vec			37.0	78.2	16.0	21.3
DUST1	60	39.1	16.0	60.0	70.1	43.6
DUST2	112	38.5	**12.2**	**54.4**	**79.9**	**50.4**
Topline			4.4	13.9	100	100

filtering technique to weed out noisy pseudo-labels on target domain unlabeled speech utterances. Through several experiments transferring from WSJ to TED-LIUM 3 and SWITCHBOARD, we show that DUST significantly improves performance over the baseline model trained only on the labeled source domain data and can recover 60% to 80% of the WER on the target domain by using only unlabeled speech from that domain.

Chapter 4

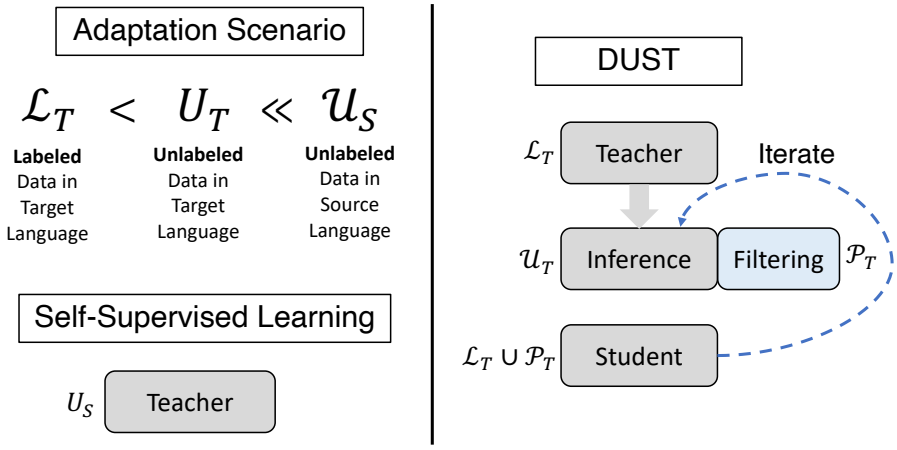
Cross-Lingual Adaptation of Monolingual Pre-Trained Speech Encoders using DUST

In this chapter[1], we apply DUST, the self-training-based domain adaptation algorithm introduced in the previous chapter, to an interesting problem; cross-lingual adaptation of pre-trained speech encoders. We can adapt a pre-trained speech encoder, Wav2Vec-2.0 (Baevski et al., 2020), for Speech Recognition in a different (than English) language using ten hours of annotated data and 100 hours of unlabeled data in the target language. After adaptation, we show that the ASR performance is at par with a state-of-the-art large multilingual pre-trained speech encoder fine-tuned on 10 hours of labeled data for the downstream task of Automatic Speech Recognition. Compared with the monolingual encoder, the multilingual encoder is trained on 566% more unlabeled speech data collected from 128 languages. Hence, in this chapter, we provide an efficient method to perform few-shot learning of Automatic Speech Recognition for an unseen target language. Our suggestion is a three-step transfer learning formula: 1) Pre-train a large speech encoder using unlabeled speech in some source language (e.g., by using the Wav2Vec-2.0 pre-training framework in Baevski et al., 2020). 2) Fine-tune the pre-trained encoder using a small amount (10

[1]The work presented in this chapter is published in Khurana, Laurent, and J. Glass (2021)

hours) of labeled data in the target language. 3) Refine the base model in 2) using 100 hours of unlabeled speech data in the target language via several iterations of the DUST algorithm (Chapter 3). Note that this three-step formula is different from the usual two-step formula: 1) Multilingual pre-training of a large speech encoder (E.g., the XLS-R framework in Babu et al., 2021), and 2) Fine-tune the pre-trained multilingual speech encoder on 10 hours of transcribed speech data in the target language. Below, we formally explain the adaptation scenario we tackle in this chapter.

Figure 4-1: An illustration of the domain adaptation scenario and our proposed cross-lingual adaptation recipe. The goal is to perform few-shot learning of Speech Recognition in a target language. First, we pre-train a speech encoder in a high-resource source language (such as English), followed by DUST in the target language. We use 10 hours of transcribed speech and 100 hours of unlabeled speech data in the target language for DUST.



The following adaptation scenario often occurs in practice: We have massive resources in high-resource languages but want to build language technology for a low-resource language. This work aims at leveraging the knowledge acquired during performing a source task (Self-Supervised Pre-training) in a high-resource source language such as English and using that knowledge to improve ASR performance in a target low-resource language. Combining Self-Supervised Learning and our proposed algorithm, DUST, is a good recipe for efficiently tackling this scenario, as explained above.

Figure 4-1 illustrates our proposed cross-lingual domain adaptation recipe. We have 1) extensive unlabeled data ($\mathcal{U}_S$) in some high-resource source language, such as

English, 2) a few hours of labeled data ($\mathcal{L}_T$) in the target language, and 3) moderately-sized unlabeled data ($\mathcal{U}_T$) in the target language. First, we perform Self-Supervised Pre-Training of an ASR model using unlabeled data $\mathcal{U}_S$ in the source language. Then, we combine labeled $\mathcal{L}_T$ and unlabeled data $\mathcal{U}_T$ in the target language to adapt the Pre-Trained ASR model to the target language. Even though the Self-Supervised Pre-Training is performed on a different source language, it acts as a good initialization for the target language ASR model, which we further improve using Dropout-Uncertainty Driven Self-Training (DUST).

4.1 Introduction

Few-shot learning, the ability to train a machine to exhibit intelligent behavior via a small amount of supervision, has been a long-standing research goal in Artificial Intelligence. To build few-shot learners, we turn to a class of transfer learning (`TL`) methods that extract knowledge from vast quantities of unlabeled data to make learning from a few labeled examples easier. Recently, Self-Supervised Learning (SSL) has emerged as a promising `TL` approach to learning from unlabeled data (T. Chen et al., 2020; Devlin et al., 2019; Oord, Y. Li, and Vinyals, 2019).

SSL (DeSa, 1993; Schmidhuber, 1990) refers to the process of Pre-Training (`PT`), a model on unlabeled data using an SSL task, such as masked self-prediction (Devlin et al., 2019). The Pre-Trained model is then Fine-Tuned (`FT`) on the target task via a few labeled examples. Hence, SSL forms the first stage of the `PT` then `FT` (`PT` $\rightarrow$ `FT`) sequential `TL` framework (D. Wang and Zheng, 2015). Recently, speech neural net encoders Pre-Trained using the `wav2vec2` SSL framework have proven to be excellent few-shot learners for automatic speech recognition (ASR) across multiple languages (Baevski et al., 2020; Conneau, Baevski, et al., 2020). However, `wav2vec2` assumes access to massive amounts of unlabeled data for PT, which diminishes their usefulness to resource-scarce languages, where the *massive unlabeled data* assumption is unrealistic.

To remedy the above issue, Conneau, Baevski, et al., 2020 proposes `XLSR-53`, a

cross-lingual sequential TL framework of the form mPT → FT, i.e., Multilingual Pre-Training of wav2vec2 followed by target language ASR fine-tuning on a few labeled examples. Indeed, Pre-Trained XLSR-53 is an excellent few-shot learner for ASR in multiple languages. However, this work shows that XLSR-53's ASR performance is relatively poor if there is a domain mismatch between the target language speech and the speech data used to Pre-Train XLSR-53. Thus, to make XLSR-53 a truly universal speech model, we would have to Pre-Train on the speech from all languages in all possible speech domains, which is an unscalable strategy. Instead, in this work, we propose a TL framework that could efficiently adapt any Pre-Trained wav2vec2 model, monolingual or multilingual, to make it an excellent few-shot ASR learner in any target language in any speech domain.

In this chapter, motivated by the SSL framework's limitations when developing ASR for a resource-scarce language, we propose a simple yet effective cross-lingual TL framework for wav2vec2 model adaptation to a target language. Our adaptation framework (Section 4.2) is a sequential TL framework consisting of three steps: First, we Pre-Train a wav2vec2 model on a high-resource language. Second, we perform supervised fine-tuning of the Pre-Trained wav2vec2 model using ten hours of labeled data on the target language ASR task. Finally, we perform Dropout Uncertainty-Driven Self-Training (DUST) (developed in Chapter 3) using a hundred hours of unlabeled speech data in the target language for adaptation of the Fine-Tuned wav2vec2 model. We make the following **key observations**:

First, we compare the ASR performance of several pre-trained transformer speech encoders trained on unlabeled speech in the English language with a multilingual pre-trained transformer speech encoder trained on unlabeled speech data collected from 53 languages. We perform ASR on eight target languages. Through this experiment, we analyze the cross-lingual transferability of the representations learned by speech encoders pre-trained only in English for the downstream task of ASR. Interestingly, we observe that the ASR performance of both the monolingual and multilingual speech encoders is at par for a target language sampled from a domain not included in the training pool of the multilingual speech encoder. Unsurprisingly, on target languages

sampled from the same domain as the multilingual speech encoder's training pool, the multilingual speech encoder performs significantly better than the monolingual encoder. Still, the monolingual encoder performs dramatically better than a randomly initialized speech encoder fine-tuned on 10 hours of transcribed speech data in the target language. Hence, an English pre-trained speech encoder is worthy of acting as the first-generation teacher in the iterative process of DUST (Chapter 3).

Starting with the pre-trained English speech encoder, we develop several generations of students via DUST using 100 hours of unlabeled data in the target language. We show that by using just 100 hours of unlabeled target language data, we can perform ASR at par with the multilingual speech encoder, trained with orders of magnitude more unlabeled target language data. Furthermore, on the out-of-domain (for the multilingual speech encoder's training pool) target language, we can surpass the ASR performance of the multilingual speech encoder by adaptation of the English pre-trained speech encoder via DUST.

A key finding of this study is that it is possible to adapt a monolingual speech encoder pre-trained on a high-resource language by using moderately-sized unlabeled data and small-sized labeled data in a target language to achieve similar performance as the multilingual pre-trained speech encoder.

4.2 Method

4.2.1 Transfer Learning Algorithm

The overall transfer learning process is described in Algorithm 1. We assume access to a set $\mathcal{L}_T$ of labeled examples and a set $\mathcal{U}_T$ of unlabeled speech utterances in the target language. Also, we are given a set $\mathcal{U}_S$ of unlabeled speech utterances in the source language. The transfer learning process proceeds by Pre-training a neural network $f_{\phi,p}$ on unlabeled source language set $\mathcal{U}_S$ with dropout layers, using a dropout probability $p \in [0, 1]$. The Pre-training process leads to the initial model $f_{\phi_0,p}$, which is Fine-Tuned on the target language labeled set $\mathcal{L}_T$ to give the first-generation teacher model

$f_{\phi_1,p}$ for Dropout-Uncertainty driven Self-Training (DUST). Next, the base teacher model $f_{\phi_1,p}$ is used to provide predictions on the target language unlabeled set $\mathcal{U}_T$ to provide pseudo-parallel data of which a subset $\mathcal{P}$ is chosen based on the model's uncertainty about its predictions on each unlabeled data point $x_u \in \mathcal{U}_T$. Finally, a student model is trained on the combined labeled $\mathcal{L}_T$ and pseudo-labeled set $\mathcal{P}$. We perform N iterations of the Teacher/Student training, where the student $f_{\phi_n,p}$ from the n^{th} iteration becomes a teacher for the $(n+1)^{th}$ iteration. Usually, in each iteration of DUST, a randomly initialized neural network is used as the student model, but in our adaptation framework, the Pre-Trained source language SSL model $f_{\phi_0,p}$ is used as the student in each DUST iteration.

4.2.2 Pre-Trained Models

In our chapter, we explore the following Pre-Trained wav2vec2 SSL models that provide the initial model $f_{\phi_0,p}$ (Algorithm 1) for transfer learning. See Section 2.2.1 for details on the wav2vec2 SSL framework.

- **Wav2Vec-2.0 Base** (w2v_base) (Baevski et al., 2020): consists of 0.1 billion parameters and is Pre-Trained on the Librispeech 960 hours (LS960) (Panayotov et al., 2015) English speech dataset in the read speech domain.

- **Wav2Vec-2.0 Large** (w2v_large) (Baevski et al., 2020): consists of 0.3 billion parameters and is Pre-Trained on either LS960 or Libri-Light 60k (LL60k) hours (J. Kahn et al., 2020) English read speech dataset.

- **Wav2Vec-2.0 Robust** (w2v_rob) (Hsu, Sriram, et al., 2021): consists of the same architecture as the large model but is trained on three speech datasets, namely Switchboard (SWBD) (300 Hours) (Godfrey, Holliman, and McDaniel, 1992), the English part of CommonVoice (CV-En) (2K hours) (Ardila et al., 2020) and LL60k (J. Kahn et al., 2020). We refer to the combination of these three datasets as LL60k+. Hsu, Sriram, et al., 2021 show that simultaneous pre-training on multiple domains makes the speech encoder relatively more robust

Algorithm 2 Adaptation Recipe for Cross-Lingual Adaptation

Given labeled data $\mathcal{L}_S$ and unlabeled data $\mathcal{U}_S$ in the source language
Given labeled data $\mathcal{L}_T$ and unlabeled data $\mathcal{U}_T$ in the target language
Given R natural numbers
Pre-Train $f_{(\phi,p)}$ on $\mathcal{U}_S$ to get $f_{(\phi_0,p)}$
Fine-Tune $f_{(\phi_0,p)}$ on $\mathcal{L}_T$ to get $f_{(\phi_1,p)}$
for n =1 to N **do**
    $f_{(\phi_{n+1},p)}$ = `DUST`($f_{(\phi_n,p)}$, $f_{(\phi_0,p)}$, $\mathcal{L}_T$, $\mathcal{U}_T$)
end for
function `DUST`($g^{\text{Teacher}}_{(\theta,p)}$, $f^{\text{Student}}_{(\psi,p)}$, $\mathcal{L}$,$\mathcal{U}$)
    Let $\mathcal{P}$ be the set of selected pseudo-labeled data points
    Let $\mathcal{E}$ be a set of edit distances
    Initialize $\mathcal{P}$ and $\mathcal{E}$ as empty sets
    for all $x_u \in \mathcal{U}$ **do**
        Compute deterministic forward pass $g^{\text{Teacher}}_{(\theta,0)}(x_u)$
        $\hat{y}^{\text{ref}}_u = \text{beam\_search}(g^{\text{Teacher}}_{(\theta,0)}(x_u))$
        for all $r \in R$ **do**
            Set random seed to r
            Compute stochastic forward pass $g^{\text{Teacher}}_{(\theta,p)}(x_u)$
            $\hat{y}^{r}_u = \text{beam\_search}(g^{\text{Teacher}}_{(\theta,p)}(x_u))$
            $e = \text{edit\_distance}(\hat{y}^{r}_u, \hat{y}^{\text{ref}}_u)$
            Add e to the set $\mathcal{E}$
        end for
        if $\max(\mathcal{E}) < \tau|\hat{y}^{\text{ref}}_u|$ (with τ a filtering threshold) **then**
            Add $\{(x_u, \hat{y}^{\text{ref}}_u), (x_u, \hat{y}^{0}_u), \ldots, (x_u, \hat{y}^{R}_u\})$ to $\mathcal{P}$
        end if
    end for
    Fine-Tune $f^{\text{Student}}_{(\psi,p)}$ on $\mathcal{A} = \mathcal{L} \cup \mathcal{P}$
    return $f^{\text{Student}}_{(\psi,p)}$
end function

to domain shifts. Since we are interested in cross-lingual adaptation to target languages sampled from different speech domains, it is natural to consider `w2v_rob` pre-trained encoder.

- **XLSR-53** (`XLSR-53`) (Conneau, Baevski, et al., 2020): consists of the same architecture as `w2v_large` which is trained on the following datasets Multilingual Speech (MLS) (Pratap et al., 2020), BABEL[2], and CommonVoice (CV) (Ardila et al., 2020), that combined consists of 53 languages. We refer to the

[2]https://catalog.ldc.upenn.edu/

combination of these three datasets as MLS+.

We use the publicly available Pre-Trained `wav2vec2` model checkpoints[3].

4.2.3 Fine-Tuning

The Fine-Tuning of Pre-Trained SSL models consists of 1) Adding a linear projection layer $h_\alpha : \mathbb{R}^{T \times d} \rightarrow \mathbb{R}^{T \times |V|}$ to the output of the pre-trained speech encoder, where V is the output character vocabulary for the task of ASR, 2) ASR task Fine-Tuning of only the projection layer for the first k training iterations and 3) Joint ASR task Fine-Tuning of both the SSL model and the projection layer until convergence. Note the `wav2vec2` SSL models consist of a Convolutional Neural Network (CNN) feature extractor, followed by a transformer encoder. The CNN feature extractor remains frozen throughout the ASR Fine-Tuning process. This fine-tuning recipe is proposed in Baevski et al., 2020 and is widely used for fine-tuning pre-trained `wav2vec2` models.

4.3 Experiment Setup

4.3.1 Target Languages

We chose seven languages from the MLS dataset (Pratap et al., 2020) as the targets for cross-lingual adaptation of the Pre-Trained `wav2vec2` SSL models, namely French (MLS/fr), German (MLS/de), Italian (MLS/it), Polish (MLS/pl), Spanish (MLS/es), Portuguese (MLS/pt) and Dutch (MLS/nl). In addition, we also target Arabic from the Multi-Genre Multi-Dialectal Broadcast News (MGB) dataset (Ali et al., 2019). To simulate the resource-scarce ASR scenario, we assume access to just ten hours of labeled data and a hundred hours of unlabeled data in each target language. We use the official (Pratap et al., 2020) nine hours labeled split in MLS for training and the one-hour split for validation. We report Word Error Rates (WERs) on the unseen development set. The hundred hours unlabeled set is sampled randomly from the entire training set (minus the utterances in the ten hours split). For Arabic, we

[3] https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec

randomly sample ten hours of labeled data, of which nine hours are used for training and one hour for validation. We also randomly sample a hundred hours of speech from the 1200 hours of MGB training set for cross-lingual adaptation. The results are reported on the standard development set. For the `XLSR-53` model, MGB/ar is considered an out-of-domain target language because `XLSR-53` is Pre-Trained on multiple datasets, including MLS, which are in the read speech and conversational domains, while MGB is in the broadcast news domain. This is evident from the high WERs of the Fine-Tuned `XLSR-53` on the MGB/ar dataset compared to the MLS target languages in Table 4.1.

4.3.2 Hyperparameters For ASR Fine-Tuning

ASR Fine-Tuning of the Pre-Trained speech encoders is performed on the ten hours labeled data $(x, y) \in \mathcal{L}_T$ in the target language T, where x is the input speech waveform and y is the corresponding sub-word token sequence. We choose characters as sub-word units for ASR training. The model is trained using the Connectionist Temporal Classification (CTC) (Graves, 2012) loss. We use the Adam optimizer with a three-phase learning rate scheduler (Baevski et al., 2020) for optimization.

The model is trained for a total of 300 epochs. For the first 4k training iterations, we only train the linear projection layer h_α. Batching is performed by pooling raw speech waveforms so that the total number of samples does not exceed 3.2 million. We use a gradient accumulation factor of four to ensure the model is updated after every four training iterations, leading to an *effective batch size* four times the original. The feature sequence output by the CNN encoder of the SSL models is randomly masked in the time dimension. We mask a span of ten consecutive time steps with a masking probability of 0.65, which leads to 65% of the input signal being masked. We use 4 V100-32gb GPUs for fine-tuning. We use the Espnet2 codebase (Watanabe, Boyer, Chang, Guo, Hayashi, Higuchi, Hori, W.-C. Huang, Inaguma, Kamo, et al., 2020) to perform all our experiments.

4.3.3 Decoding

We use beam search decoding without a language model (LM) with a beam size of 10. We do not use an LM because we are solely concerned about the acoustic model adaptation in this work. Also, we might not have text data to train an LM in a resource-scarce ASR scenario.

4.4 Evaluation

4.4.1 Cross-Lingual Transferability of Pre-Trained Speech Encoders

In **Tables 4.1, 4.2**, we show the cross-lingual transferability of different Pre-Trained `wav2vec2` models on eight target languages. The goal is to analyze how much of the multilingual `XLSR-53` topline's performance can be recovered by simply Fine-Tuning the English `wav2vec2` models on ten hours of labeled data in target languages. We Fine-Tune a randomly initialized transformer encoder with the same architecture as `w2v_base` on ten hours of labeled data in each language to use as a baseline. We perform ASR Fine-Tuning of several Pre-Trained English `wav2vec2` on ten hours of labeled data in target languages and compare their ASR performance against the Fine-Tuned `XLSR-53` model topline.

We make the following conclusions:

Pre-Training Matters. ASR Fine-Tuning of Pre-Trained English `wav2vec2` models significantly improve WERs on target languages over the *randomly initialized encoder* baseline. Through the simple `PT` $\rightarrow$ `FT` process, we can recover on average 79% to 86% of the WER and 88% to 93% of the CER of the `XLSR-53` topline.

Model Size matters. By Fine-Tuning `w2v_large` that is Pre-Trained on the LS960 dataset, we can recover on average 83% of the topline WER compared to 79% achieved by Fine-Tuning `w2v_base` that is also Pre-Trained on LS960. Hence,

Table 4.1: Cross-Lingual Transferability of Pre-Trained English `wav2vec2` models on eight target languages. Seven languages are from the in-domain MLS dataset (Pratap et al., 2020) of read audiobooks, while Arabic is from the out-of-domain MGB broadcast news dataset (Section 2.4.1). We report Word Error Rates for different pre-trained speech encoders fine-tuned on 10 hours of transcribed speech data in the eight target languages. We compare English `wav2vec2` models against multilingual `XLSR-53` topline in terms of Word Error Rate Recovery (WERR), which is given by $\frac{x-\text{topline}}{\text{baseline}-\text{topline}}$, where x is the ASR performance achieved by fine-tuning a `wav2vec2` English pre-trained model.

		Word Error Rate [%]										WERR
Target Langs		MLS/en	MLS/fr	MLS/de	MLS/it	MLS/pl	MLS/es	MLS/pt	MLS/nl	MGB/ar		
Model	PT										Avg.↓	Avg.↑
Baseline		119.1	114.2	106.0	99.5	111.9	99.5	107.0	108.8	112.0	107.4	0
`w2v_base`	LS960	23.4	44.0	28.6	34.1	35.6	37.2	41.1	47.2	47.4	39.4	79.0
`w2v_large`	LS960	17.1	40.9	28.3	33.3	32.0	23.6	38.6	45.0	42.7	35.6	83.6
`w2v_large`	LL60k	12.3	39.9	26.7	31.8	32.8	21.9	35.6	42.6	42.0	34.2	85.6
`w2v_rob`	LL60k+	12.8	38.3	26.7	30.3	34.2	22.9	34.2	39.1	41.6	33.4	**86.3**
`w2v_large_sup`	LL60k	**7.6**	44.2	31.1	37.7	46.5	28.1	40.8	51.3	50.6	41.3	78.8
Topline (`XLSR-53`)	MLS+	17.6	**19.7**	**11.1**	**17.1**	**16.4**	**7.9**	**20.4**	**21.7**	**37.9**	**19.0**	100

Table 4.2: Cross-Lingual Transferability of Pre-Trained English `wav2vec2` models on eight target languages. Seven languages are from the in-domain MLS dataset (Pratap et al., 2020) of read audiobooks, while Arabic is from the out-of-domain MGB broadcast news dataset (Section 2.4.1). We report Character Error Rates for different pre-trained speech encoders fine-tuned on 10 hours of transcribed speech data in the eight target languages. We compare English `wav2vec2` models against multilingual `XLSR-53` topline in terms of Character Error Rate Recovery (CERR), which is given by $\frac{x-\text{topline}}{\text{baseline}-\text{topline}}$, where x is the ASR performance achieved by fine-tuning a `wav2vec2` English pre-trained model.

		Character Error Rate [%]										CERR
Target Langs		MLS/en	MLS/fr	MLS/de	MLS/it	MLS/pl	MLS/es	MLS/pt	MLS/nl	MGB/ar		
Model	PT										Avg.↓	Avg.↑
Baseline		58.5	51.6	41.5	35.0	44.7	37.3	45.3	50.0	51.5	44.6	0
`w2v_base`	LS960	8.1	14.5	6.9	7.3	6.9	8.6	10.9	14.2	15.1	10.6	87.8
`w2v_large`	LS960	5.8	13.3	6.8	6.9	6.2	5.6	10.2	13.3	14.2	9.6	90.5
`w2v_large`	LL60k	4.0	12.7	6.4	6.7	6.4	5.1	9.4	12.6	13.2	9.1	92.1
`w2v_rob`	LL60k+	4.2	12.3	6.4	6.2	6.6	5.3	8.9	11.8	13.1	8.8	**92.5**
`w2v_large_sup`	LL60k	**2.5**	14.2	7.2	7.8	9.0	6.4	10.4	15.3	15.8	10.8	88.6
Topline (`XLSR-53`)	MLS+	6.3	**6.5**	**3.1**	**3.6**	**3.3**	**2.1**	**5.3**	**6.3**	**12.0**	**5.3**	100

a larger pre-trained model achieves better cross-lingual transfer.

Pre-Training dataset size matters upto a point. Fine-Tuned `w2v_large` that is Pre-Trained on LL60k recovers on average 86% of the topline WER compared to 84% recovered by Fine-Tuning `w2v_large` that is Pre-Trained on LS960. But the gap in average Word Error Rate Recovery (WERR) between `w2v_rob` that is Pre-Trained on the combined CV, SWBD, and LL60k datasets, and `w2v_large` that is Pre-Trained only on LL60k is less than one percentage point (pp).

ASR Fine-Tuning of SSL models on source language hurts transfer. The average WERR on target languages of `w2v_large_sup` model, which is Pre-Trained on LL60k followed by its ASR Fine-Tuning on labeled LS960 is worse than directly Fine-Tuning the Pre-Trained `wav2vec2` models on the target languages. The WERR for `w2v_large_sup` is about 8pp worse than `w2v_rob` that is directly Fine-Tuned on target languages.

About the out-of-domain Arabic Target Language. We see that on the seven in-domain languages (MLS/x, where x is the target language) `XLSR-53` achieves an average WER of 16.5% compared to 29.8% achieved by the ASR Fine-Tuning of `w2v_rob`, the best of the English `wav2vec2` models, giving a performance gap of about 14pp between the two. However, on the out-of-domain Arabic target language (MGB/ar), the gap is less than 4pp.

4.4.2 Adaptation of English Wav2Vec-2.0 to French and Arabic

Next, using DUST, we perform a cross-lingual adaptation of Pre-Trained `wav2vec2` models. We choose French and Arabic as the target languages for transfer learning and `w2v_rob` and `XLSR-53` as the target models for adaptation.

In Table 4.3, we use DUST to perform a cross-lingual adaptation of Pre-Trained `w2v_rob` to French (MLS/fr). DUST proceeds as follows: 1) First, we perform the

Table 4.3: Transfer of Pre-Trained `w2v_rob` to the target French language in the MLS dataset

		WER [%]		WERR [%]
Method	$\|\mathcal{P}\|$ [k]	$\mathcal{P}$	MLS / fr	MLS / fr
Baseline (`w2v_rob`)			38.3	0
DUST1	11	20.2	31.9	34.4
DUST2	24	20.3	27.4	58.6
DUST3	30	20.0	24.2	75.8
DUST4	30	19.2	23.5	79.6
DUST5	30	**18.7**	**22.3**	**86.0**
Topline (`XLSR-53`)			19.7	100

ASR Fine-Tuning of the initial `w2v_rob` ($f_{\phi_0,p}$) model using the standard nine hours labeled split provided by MLS/fr dataset to get the first-generation teacher $f_{\phi_1,p}$ (Section 4.2). 2) Second, $f_{\phi_1,p}$ is used to generate pseudo-labels on the random 100 hours unlabeled split from MLS/fr, which amounts to about 30k utterances, using the pseudo-label generation process explained in Section 4.2 to give a set $\mathcal{P}$ of pseudo-parallel data. We use 0.2 as the value of the DUST filtering threshold τ. We choose τ blindly without tuning it on a labeled validation set. 3) Lastly, we Fine-Tune `w2v_rob` (student), $f_{\phi_0,p}$, on the combined labeled and pseudo-labeled data $\mathcal{P}$ to get $f_{\phi_2,p}$, which is used as the teacher for the next iteration of DUST. We perform a total of five DUST iterations. The final student model $f_{\phi_5,p}$ achieves a WER of 22.3% which is 16pp lower than the WER of 38.3% achieved by the first generation teacher model $f_{\phi_1,p}$. Furthermore, $f_{\phi_5,p}$ can recover 86% of the `XLSR-53` topline's WER. Additionally, we make the following observations: 1) Unsurprisingly, the size of the filtered pseudo-label set $\mathcal{P}$ (denoted as $|\mathcal{P}|$ in Table 4.3) is larger in later DUST iterations due to the continual improvement in the quality of the student (see WER [%] in Table 4.3), which leads to an improved teacher for subsequent DUST iterations; an improved teacher leads to cleaner pseudo-labels and hence less rejected unlabeled data points during the pseudo-label filtering process. 2) Also, in the later DUST iterations, the quality of the pseudo-labels improves, which is implied by the lower WER on pseudo-label set $\mathcal{P}$ during the later iterations. Next, we consider Arabic (MGB/ar) the target

Table 4.4: Transfer of Pre-Trained `w2v_rob` and `XLSR-53` models to the target Arabic Language in the MGB dataset

		WER [%]	
Method	$\|\mathcal{P}\|$ [k]	$\mathcal{P}$	MGB / ar
Baseline (`w2v_rob`)			41.6
DUST1	12	21.0	32.7
DUST2	26	21.2	27.4
DUST3	30	20.8	25.2
DUST4	30	19.5	23.1
DUST5	30	**18.7**	**21.2**
Topline (`XLSR-53`)			37.9
Baseline (`XLSR-53`)			37.9
DUST1	13	20.3	31.1
DUST2	29	20.4	26.3
DUST3	30	20.1	24.1
DUST4	30	18.5	22.5
DUST5	30	**18.1**	**20.8**

language for transfer learning, a more challenging transfer learning scenario.

In **Table 4.4**, we perform adaptation of `w2v_rob` and `XLSR-53` to the MGB/ar dataset. Here, the results are achieved by following the same adaptation process detailed above for experiments in Table 4.3. After five DUST iterations, we achieve the final WER of 20.8% when starting with a Fine-Tuned `XLSR-53` model as the first generation teacher $f_{\phi_1,p}$. This result is about 17pp better than the WER of 37.4% with $f_{\phi_1,p}$. Similar improvements are achieved when using the Fine-Tuned `w2v_rob` as $f_{\phi_1,p}$ for DUST iterations.

4.5 Chapter Summary

We conclude by summarizing the key findings of this chapter. We show (Table 4.1, 4.2) that the Pre-Trained English language `wav2vec2` models transfer well across multiple languages. In particular, we show that by performing ASR Fine-Tuning of `wav2vec2_robust` on ten hours of labeled data in a target language, we can recover on average 86% of the performance of the topline multilingual `XLSR-53` model that is

Pre-Trained on 53 languages and Fine-Tuned on the same amount of labeled target language data. This finding concurs with similar findings of (Rivière et al., 2020) on the cross-lingual transfer of monolingual Pre-Trained speech representations to different target languages for phoneme recognition. Our work goes a step further and proposes a simple yet effective cross-lingual transfer learning algorithm (Section 4.2) for adaptation of monolingual `wav2vec2` models via Dropout Uncertainty-Driven Self-Training (DUST) by leveraging hundred hours of unlabeled speech data from the target language. We show (Table 4.3) that DUST improves over the baseline model that is Fine-Tuned only on labeled target language data and can recover 86% of the WER of the topline `XLSR-53` model when adapting to French. We show similar results (Table 4.4) when considering Arabic as the target language.

This chapter proposes a departure from the traditional two-step cross-lingual transfer learning formula of multilingual pre-training followed by target language-specific ASR task fine-tuning. Instead, we perform Pre-Training on a high-resource source language and adapt the pre-trained speech encoder via DUST to the desired target language for the ASR task. Our method is suitable for low-resource ASR scenarios. In low-resource scenarios, we do not have resources for large-scale self-supervised learning of speech encoders. But, we might have a related language (in the same language family) that is high-resource. We can pre-train the speech encoder on the related language and then use DUST to adapt the pre-trained model to the low-resource target language as explained in this chapter (Section 4.2).

Chapter 5

Semantically Aligned Multimodal Cross-Lingual Speech Representations

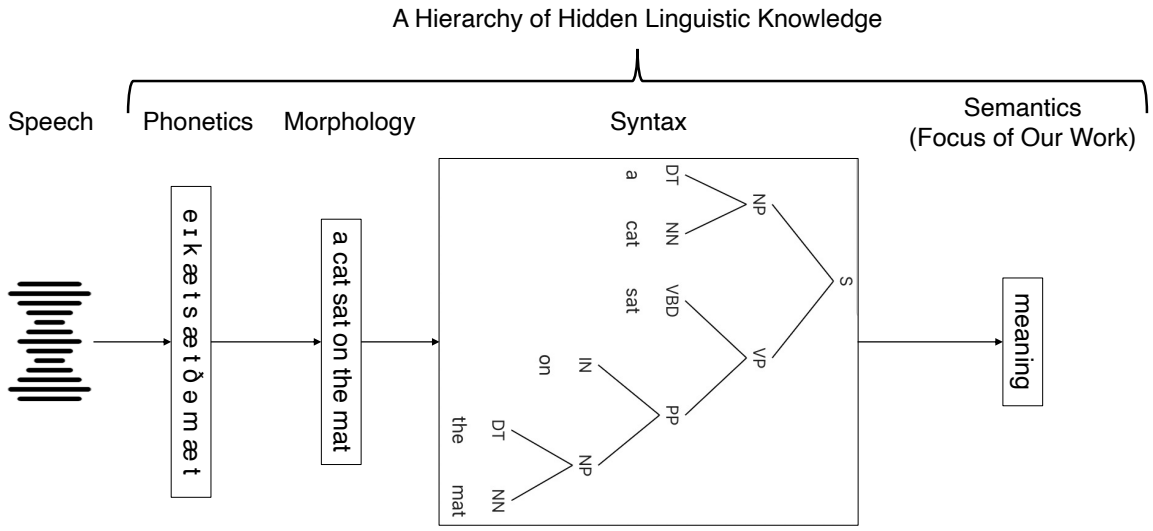
Different types of linguistic information exist in a spoken utterance (Fig. 5-1), from the low-level knowledge of acoustic-phonetics (sound inventories in the utterance), through morphology (word identities and word time stamps), syntax (the discrete structure, such as the constituency parse tree, that governs the relationship among words spoken in the utterance), and finally, semantics which refers to the knowledge about the literal meaning of the spoken utterance. Self-Supervised Representation Learning methods that learn from unlabeled speech, such as Wav2Vec-2.0 and XLS-R that train a transformer-based neural network to encode structured vector representations of speech in its several hidden layers are excellent at encoding low-level linguistic knowledge about phonetics.

This chapter[1] develops a Semantically Aligned MUltimodal Cross-Lingual Speech Representation, `SAMU-XLS-R`, that, unlike the self-supervised transformer encoders such as Wav2Vec-2.0, encodes high-level linguistic knowledge about the meaning of the speech utterance. To that end, we propose a multilingual joint speech-text embedding framework. First, we train a speech encoder using semantic supervision provided by the text modality and a pre-trained Language-Agnostic BERT Sentence Encoder (LaBSE) introduced in Feng et al. (2020) to learn a joint speech-text embedding

[1]The work presented in this chapter is published in Khurana, Laurent, and J. Glass (2022)

space structured to represent semantic knowledge. Then, we analyze the semantic multilingual joint speech-text embeddings space on cross-lingual speech-to-text and speech-to-speech translation retrieval. In the next chapter, we build multilingual speech-to-text translation technology using `SAMU-XLS-R`. `SAMU-XLS-R` improves cross-lingual transfer from high to low-resource language translation tasks.

Figure 5-1: An illustration of a speech utterance's linguistic knowledge hierarchy. This work focuses on training a neural network model that encodes semantic knowledge in its activations.



5.1 Introduction

Recently, self-supervised pre-training of large transformer encoders on massive amounts of unlabeled audio data followed by task-specific fine-tuning has emerged as the de-facto approach for achieving state-of-the-art performance on several tasks in spoken language processing. However, popular self-supervised representation learning (SSL) approaches such as Wav2vec-2.0 (Baevski et al., 2020) and others (Y.-A. Chung and J. Glass, 2020; A. H. Liu, Y.-A. Chung, and J. Glass, 2020; Pascual et al., 2019; Schneider et al., 2019; Khurana, Laurent, Hsu, et al., 2020; Conneau, Baevski, et al., 2020; Hsu, Bolte, et al., 2021; Babu et al., 2021; S. Chen et al., 2021; Y.-A. Chung, Zhang, et al., 2021; Bapna, Cherry, et al., 2022a) learn speech embedding at acoustic

frame-level, i.e., for short speech segments of duration 10 to 20 milliseconds.

Unlike previous works, this work focuses on learning semantically-aligned multimodal utterance-level cross-lingual speech representations (`SAMU-XLS-R`). The SAMU-XLS-R's embedding vector space is multimodal since it is shared between the speech and the text modalities. It is cross-lingual since various languages share it. Furthermore, it's semantically aligned since, in the `SAMU-XLS-R`'s vector space, a spoken utterance is clustered together with its speech and text translations. We show a two-dimensional illustration of the desired embedding vector space in Figure 5-2. For example, consider the English phrase *A bird is bathing in the sink*. Now, in `SAMU-XLS-R`'s embedding space, the written form of the above phrase should be clustered together with its written and spoken forms in various languages (Japanese, French, and Arabic in the figure). And, in some other regions of the embedding space, the phrase *Mr. President* is clustered with its written and spoken form in several languages. Unfortunately, the acoustic frame-level unimodal contextual representation learning frameworks like Wav2vec-2.0 (Baevski et al., 2020) or the multilingual `XLS-R` (Conneau, Baevski, et al., 2020; Babu et al., 2021) do not learn an embedding space with the same properties. Encoding semantics is among the many missing pieces in the self-supervised speech representation learning puzzle.

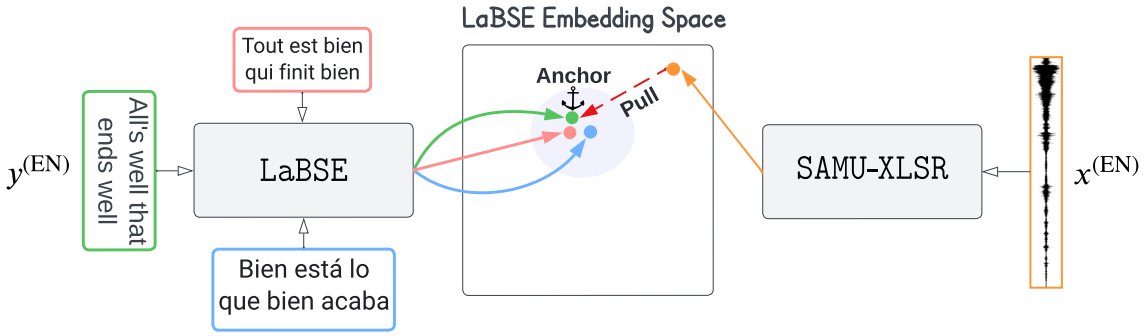
On the other hand, several transformer encoders for text have been proposed in recent years that go beyond token-level contextual representations and learn cross-lingual semantically-aligned sentence embedding vector spaces across several languages (Schwenk and Douze, 2017a; Artetxe and Schwenk, 2019a; Feng et al., 2020). These models have found use in bi-text data mining. The task is to retrieve the text translation in a target language for a given sentence query in a source language by matching the query sentence embedding with those of sentences in the target language search database (Schwenk, 2018; Schwenk, Chaudhary, et al., 2019; Schwenk, Wenzek, et al., 2019a). Given that text encoders can successfully learn semantically aligned cross-lingual sentence embedding spaces, we ask whether it is possible to make these text embedding spaces multimodal by learning to map speech utterances in the semantically-aligned cross-lingual text embedding space.

Figure 5-2: An illustration of the desired cross-lingual joint speech-text embedding space. The embedding space is semantically aligned, i.e., a speech utterance such as *Mr. President* is clustered together with its corresponding speech and text translations in the multimodal embedding space in several other languages.

To that end, we propose a multimodal learning framework for fine-tuning the pre-trained multilingual `XLS-R` speech encoder via knowledge distillation from the pre-trained language-agnostic BERT sentence encoder `LaBSE` (Feng et al., 2020). Also, we append a pooling mechanism and a non-linear projection layer after the last layer of the pre-trained `XLS-R` encoder to transform the frame-level contextual representations into a single utterance-level embedding vector. Then, we train the speech encoder using transcribed speech; given a speech utterance, the parameters of the speech encoder are tuned to accurately predict the text embedding provided by the `LaBSE` encoder of its corresponding transcript. Because `LaBSE`'s embedding vector space is semantically aligned across various languages, the text transcript would be clustered with its text translations. Hence, we get cross-lingual speech-to-text associations for free by simply using transcribed speech to train the speech encoder via the proposed knowledge distillation framework. For a pedagogical description, see Figure 5-3.

One of the use cases of the `SAMU-XLS-R` embedding space described above is for data mining. Recent years have seen remarkable progress in Automatic Speech Recognition across several domains and languages. The next frontier in spoken language process-

Figure 5-3: A pedagogical description of how learning with transcribed speech data using LaBSE as the teacher could lead to the emergence of cross-lingual speech and text associations. In this illustration, we use English speech $x^{(EN)}$ and its transcription $y^{(EN)}$ for training. SAMU-XLS-R's parameters are tuned to close the distance between the speech embedding given by SAMU-XLS-R in orange and LaBSE's embedding (Anchor) of the corresponding text transcript in green. Since LaBSE's text embedding space is semantically aligned across various languages, pulling the speech embedding towards the anchor embedding automatically leads to cross-lingual speech-text alignments in the joint speech-text embedding space without ever seeing cross-lingual associations during training. In practice, we train SAMU-XLS-R with multilingual transcribed speech, not just English.



ing is automatic speech-to-text and speech-to-speech machine translation. Developing speech-based MT systems would require massive amounts of parallel translated speech data in several languages, which could be highly costly to collect. But, the multimodal cross-lingual embedding space illustrated in Fig. 5-2 could address this issue. We could build a cross-lingual speech-to-text and speech-to-speech retrieval pipeline, which could entirely or, in some cases, partially automate the process of collecting either text or speech translations corresponding to a spoken utterance. We advise the reader to look at papers in Natural Language Processing that use multilingual sentence encoders to perform cross-lingual text mining, such as (Schwenk and Douze, 2017b; Artetxe and Schwenk, 2019b; Schwenk, Wenzek, et al., 2019b; Feng et al., 2020).

Cross-lingual speech-to-text mining to create parallel speech-text translation datasets is just one possible application of SAMU-XLS-R. But, the potential application in zero-shot speech-to-text translation motivates us to work on this problem. The success of zero-shot translation depends on learning a semantically-aligned language invari-

ant embedding vector space or an *interlingua* for different spoken languages, where speech utterances and their translations are clustered together. We show that this is an emergent property in SAMU-XLS-R's embedding vector space as a result of training SAMU-XLS-R using the proposed multimodal learning framework (Section 5.4.5). Some text machine translation papers that inspire us in the zero-shot translation are (Gu et al., 2019; Arivazhagan et al., 2019). We make the following **contributions**:

- We propose a simple yet effective multimodal learning framework for semantically-aligned multimodal (joint speech-text) utterance-level speech representation (SAMU-XLS-R) shared across multiple languages (Section 5.2).
- We demonstrate the effectiveness of our models on several zero-shot cross-lingual speech-to-text and speech-to-speech translation retrieval tasks (Section 5.4.5).
- We analyze to understand better the various design decisions that went into constructing SAMU-XLS-R (Section 5.5).

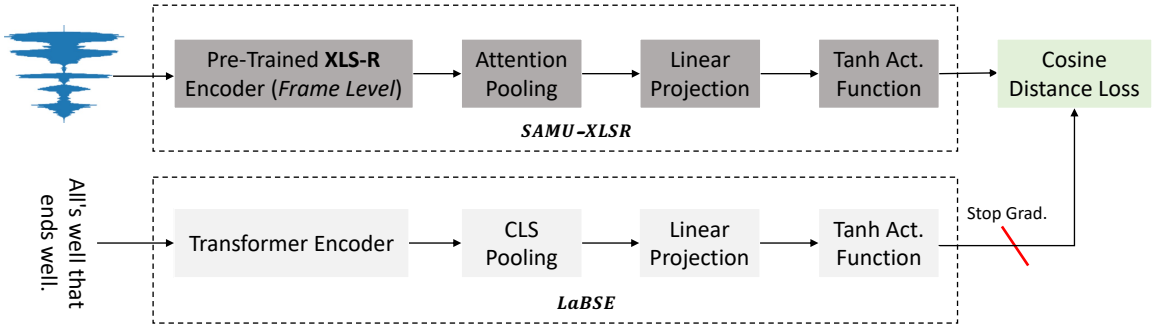
A work similar to ours is presented in (Duquenne, Gong, and Schwenk, 2021). Unlike the previous work, we evaluate our model on multiple datasets across many languages, emphasizing low-resource languages.

5.2 Model

5.2.1 Joint Speech-Text Embedding Framework

We train SAMU-XLS-R using a multilingual set $\mathcal{D}$ of paired examples $(x^{(l)}, y^{(l)})$, where $x^{(l)}$ is the speech waveform, and $y^{(l)}$ is its text transcript in language l. Given a training example, $(x^{(l)}, y^{(l)})$, we transform the sequence of discrete tokens $y^{(l)}$ to a dense embedding vector $\mathbf{z}_T \in \mathbb{R}^d$ using a text encoder g_ϕ, and the series of speech samples $x^{(l)}$ into a dense embedding vector $\mathbf{z}_S \in \mathbb{R}^d$ using a speech encoder f_θ. Then, we update the parameters of the speech encoder f_θ so that the distance between the speech embedding $\mathbf{z}_S$ and the text embedding $\mathbf{z}_T$ is minimized. The following

Figure 5-4: An illustration of our proposed multimodal training framework. The learning framework comprises a speech and a text encoder. The speech encoder transforms a raw speech waveform into an embedding vector. The text encoder transforms the transcript corresponding to the speech utterance into an embedding. The text encoder is initialized using the pre-trained Language-Agnostic BERT Sentence Embedding (LaBSE) model (Feng et al., 2020). The speech encoder below the pooling layer is initialized using the pre-trained XLS-R speech encoder (Babu et al., 2021).



equation gives the training loss for a single example:

$$\mathcal{J}(\theta, \phi) = \text{distance}(\mathbf{z}_S, \mathbf{z}_T) \tag{5.1}$$

We use the pre-trained Language-agnostic BERT Sentence Encoder (`LaBSE`) as the text encoder g_ϕ and `SAMU-XLS-R` as the speech encoder f_θ. The parameters θ of the speech encoder are updated during training, while the parameters ϕ of the text encoder remain fixed. An illustration of the multimodal learning framework is shown in Figure 5-4.

5.2.2 `SAMU-XLS-R` Speech Encoder, f_θ

`SAMU-XLS-R` consists of a pre-trained frame-level `XLS-R` speech encoder (Babu et al., 2021) followed by a mechanism for pooling the frame-level contextual representations into a single embedding vector.

`XLS-R` Encoder to generate Contextual Embedding

Algorithm 3 presents the computations performed by `XLS-R`. The `XLS-R` speech encoder consists of a deep convolutional neural network (`Conv`) that maps 1D time

Algorithm 3 We detail the computations performed by the `XLS-R` transformer speech encoder. `LN` refers to Layer-Normalization, `Conv` to a multi-layered Convolutional Neural Network, `DO` is Dropout, `MHSA` is Multi-Headed Self-Attention, `ACTFn` is an Activation Function like `ReLU`, and `FC` is a Fully-Connected Layer.

Input: Raw speech waveform $\mathbf{a}$
Output: Contextual speech embedding sequence $\mathbf{c}$
$\mathbf{h} = \texttt{LN}(\texttt{Conv}(\mathbf{a}))$
$\mathbf{h} = \texttt{Mask}(\mathbf{x})$
$\mathbf{h} = \mathbf{h} + \texttt{PosConv}(\mathbf{h})$
for $i = 1$ to $L - 1$ **do**
    $\mathbf{h} = \texttt{TransformerLayer}^{(i)}(\mathbf{h})$
end for
$\mathbf{c} = \texttt{LN}(\texttt{TransformerLayer}^{(L)}(\mathbf{h}))$
function $\texttt{TransformerLayer}(\mathbf{x})$
    $\mathbf{x} = \texttt{DO1}(\texttt{MHSA}(\texttt{LN1}(\mathbf{x}))) + \mathbf{x}$
    $\mathbf{x} = \texttt{DO3}(\texttt{FC2}(\texttt{DO2}(\texttt{ACTFn}(\texttt{FC1}(\texttt{LN2}(\mathbf{x})))))) + \mathbf{x}$
end function

series representing the sample values of the speech waveform ($\mathbf{a}$) into a 2D sequence of feature vectors $\mathbf{h} \in \mathbb{R}^{T \times 512}$. Each feature vector $\mathbf{h}_t$ represents 20ms of the speech signal. The time resolution of $\mathbf{h}_t$ is similar to that of an acoustic frame. Therefore, we refer to $\mathbf{h}$ as frame-level representations. Next, the feature sequence $\mathbf{h}$ is transformed into contextual representations $\mathbf{c} \in \mathbb{R}^{T \times 1024}$ by a stack of Self-Attention transformer blocks (`TransformerLayer`). There are 24 transformer blocks. Each block comprises a Multi-Headed Self-Attention (`MHSA`) module, and two Fully-Connected layers (`FC1`, `FC2`). The attention vector size is 1024, with 16 attention heads in each transformer block. We use the publicly available pre-trained `XLS-R` checkpoint[2] which was trained on 400k hours of unlabeled speech data in 128 languages.

Pooling the Contextual Embedding

Next, we use Self-Attention pooling (Safari, India, and Hernando, 2020) strategy to get a single utterance-level embedding vector $\mathbf{e} \in \mathbb{R}^{1024}$. In this pooling strategy, we

[2] https://huggingface.co/facebook/wav2vec2-xls-r-300m

take a weighted combination $\sum_{t=1}^{T} v_t \mathbf{c}_t$ of contextual vectors, where $\mathbf{v} = (v_1, \ldots, v_T)$ is the attention vector, given by the following equation:

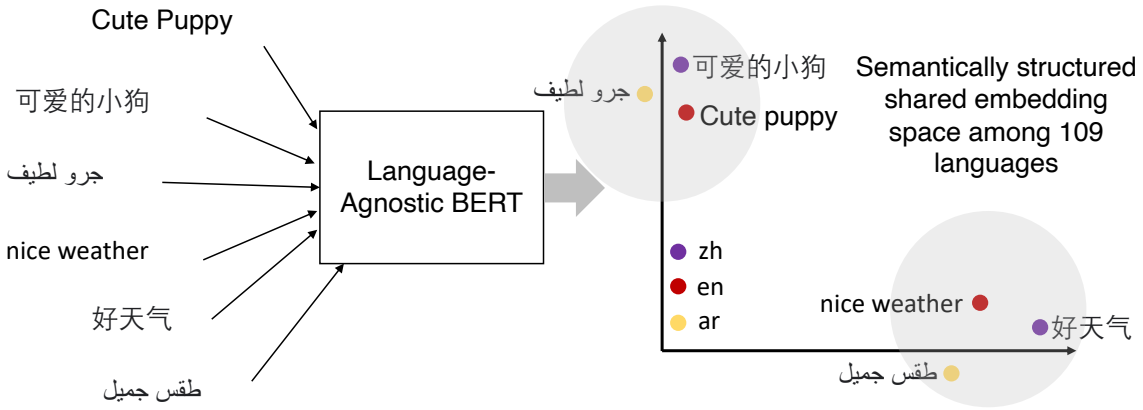
$$\mathbf{v} = \text{softmax}(\mathbf{c} \ @ \ \mathbf{w}) \tag{5.2}$$

where, $\mathbf{c} \in \mathbb{R}^{T \times 1024}$ is the contextual embedding sequence, $\mathbf{w} \in \mathbb{R}^{1024}$, @ refers to the matrix-vector product which gives the attention vector $\mathbf{v} \in \mathbb{R}^{T}$, such that $\sum_t v_t = 1$. The weight vector $\mathbf{w}$ is learned during training.

Finally, we take a non-linear projection of the embedding vector $\mathbf{e}$ to get the speech embedding $\mathbf{z}_S$. The `SAMU-XLS-R` speech encoder consists of approximately 300 million trainable parameters (weights and biases).

5.2.3 LaBSE Text Encoder, g_ϕ

Figure 5-5: An illustration of LaBSE's (Feng et al., 2020) text embedding space. LaBSE is a multilingual text encoder that can embed text from over 100 hundred languages in a shared semantically aligned embedding space, i.e., a sentence such as *Cute Puppy* is clustered together with its translations in hundred other languages supported by LaBSE.



The key ingredient in our proposed multimodal learning framework is the `LaBSE` text encoder g_ϕ, which allows us to learn a joint speech-text embedding space that is semantically aligned and shared across different languages as illustrated in Fig. 5-5. `LaBSE` is a language-agnostic text encoder for text with an architecture similar to

the BERT transformer encoder (Devlin et al., 2019). However, unlike BERT, LaBSE is a sentence embedding model, which is trained using both masked (Devlin et al., 2019) and translation language modeling (Lample and Conneau, 2019b) objective functions. LaBSE consists of a token-level transformer encoder with 12 MHSA layers and a pooling mechanism to construct a dense sentence-level embedding vector.

The LaBSE's transformer encoder takes as input text that is tokenized into "word-pieces" (Schuster and Nakajima, 2012; Yonghui Wu, Schuster, Zhifeng Chen, Le, Norouzi, W. Macherey, Krikun, Cao, Gao, K. Macherey, Klingner, Shah, Johnson, X. Liu, L. Kaiser, et al., 2016) and outputs a sequence of contextual token embedding $\mathcal{W} \in \mathbb{R}^{L \times 768}$. A non-linear projection of the CLS token embedding is used as the sentence embedding $\mathbf{z}_T \in \mathbb{R}^{768}$, which is used as the training target for SAMU-XLS-R training. We use the pre-trained LaBSE model checkpoint[3] hosted on the Huggingface (Wolf et al., 2019) models[4] platform. We use CLS token embedding for sentence representation, called *CLS pooling*.

LaBSE embeds sentences from 109 languages into a shared semantically-aligned embedding vector space. Unlike LaBSE, other multilingual text encoders such as XLM-R (Conneau, Khandelwal, et al., 2019b) do not learn an aligned sentence embedding space. Therefore, to achieve our goal of embedding speech in a semantically aligned vector space, we use LaBSE as the teacher for training SAMU-XLS-R.

5.3 Training

5.3.1 Training Data, $\mathcal{D}$

We train SAMU-XLS-R on transcribed speech in 25 languages derived from the publicly available CommonVoice-v7 (CoVo) dataset. The 25 languages are namely, English (EN), French (FR), German (DE), Spanish (ES), Catalan (CA), Italian (IT), Welsh (CY), Russian (RU), Chinese (China) (ZH_CN), Chinese (Taiwan) (ZH_TW), Chinese (Hong Kong) (ZH_HK), Portuguese (PT), Polish (PL), Persian (FA), Estonian

[3] https://huggingface.co/sentence-transformers/LaBSE
[4] https://huggingface.co/models

(ET), Mongolian (MN), Dutch (NL), Turkish (TR), Arabic (AR), Swedish (SV_SE), Latvian (LV), Slovenian (SL), Tamil (TA), Japanese (JA) and Indonesian (ID). Table 5.1 shows the per-language transcribed data available in CoVo. The total training data size is 6.8K hours.

The data is highly imbalanced. The top 5 high-resource languages make up 72% of the training data, while the bottom 14 low-resource languages make up just 10%. The above-mentioned problem could lead to `SAMU-XLS-R` severely under-fitting on low-resource languages because `SAMU-XLS-R`, during its training lifetime, might encounter transcribed speech data from low-resource languages in its train mini-batch only a few times. Following (Lample and Conneau, 2019a; Y. Liu et al., 2020a) we re-balance the training set $\mathcal{D}$ by up/down-sampling data from each language l with a ratio λ_l:

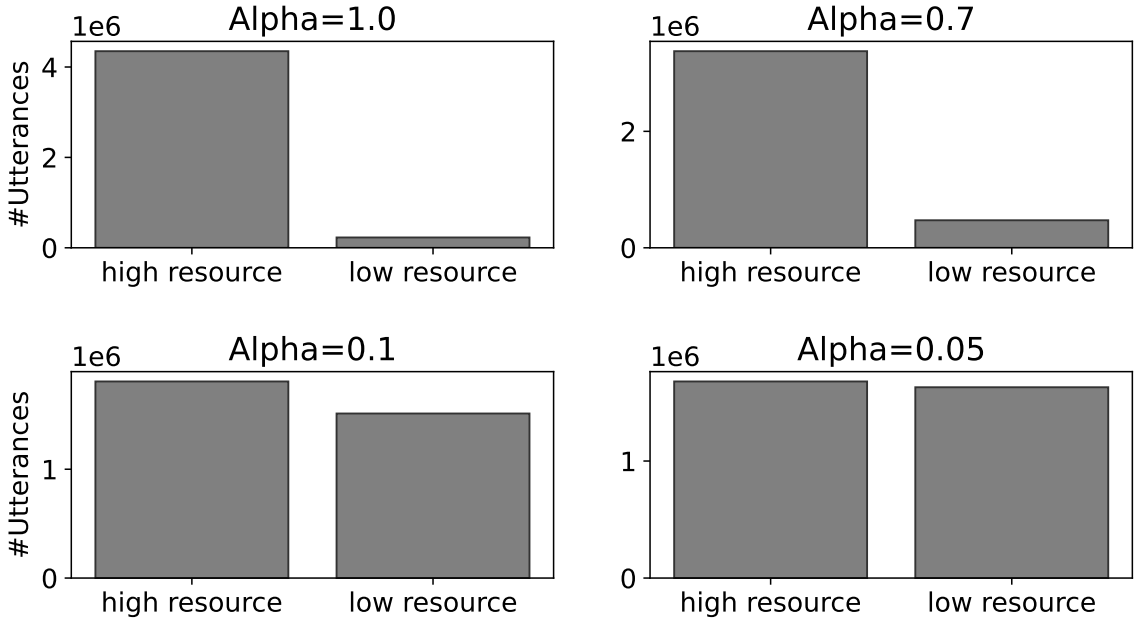
$$\lambda_l = \frac{1}{p_l} \frac{p_l^\alpha}{\sum_l p_l^\alpha} \text{ with } p_l = \frac{n_l}{\sum_{l=1}^{L} n_l} \tag{5.3}$$

where, α is the smoothing parameter, n_l is the number of utterances for language l in the training set. Figure 5-6, shows how varying α between 1.0 and 0.05 re-balances the training set. As we make α smaller, observe that the share of low-resource languages in the training set becomes approximately the same as that of high-resource languages. It is important to note that when we up-sample data from low-resource languages, we repeat the utterances from those languages. Down-sampling data from high-resource languages involve picking random utterances according to the ratio λ_l. Hence, training with a re-balanced training set created using a small value of α could result in a drop in performance on high-resource languages compared to the model trained with the original unbalanced training set. We study the smoothing parameter *alpha*'s effect on the model's downstream task performance in Section 5.5.

5.3.2 Optimization Settings

We train `SAMU-XLS-R` for 400K training iterations, on 32 V100-32gb GPUs, with a per-GPU mini-batch size of approximately 2 hours of transcribed speech. Following

Figure 5-6: Re-balancing the training set with different smoothing parameter values α. As we make α smaller, the share of low-resource languages in the training set becomes approximately the same as that of high-resource languages. Up-sampling data from low-resource languages implies repeating the utterances from those languages. Down-sampling data from high-resource languages involve picking random utterances according to the ratio λ_l



(Conneau, Baevski, et al., 2020), we use the Adam optimizer for updating the model parameters with a three-phase learning rate scheduler; Warm up the learning rate to a maximum value of 1e-4 for the first 10% of the training iterations, then the learning rate remains constant for the next 40% of the training iterations, and finally decays linearly for the rest of the iterations. For the first 10K training iterations, only the projection layer of `SAMU-XLS-R` encoder is trained while the pre-trained frame-level `XLS-R` speech encoder remains fixed. We do not update the weights of the `XLS-R`'s convolutional feature extractor throughout the training process. Also, we use a modified version of SpecAugment (Park, Chan, et al., 2019) on the feature sequence $\mathcal{H}$ (Section 5.2.2) to mask the input to the `XLS-R`'s transformer encoder, which leads to better performance on downstream tasks. The above-mentioned training settings are the standard for fine-tuning the pre-trained `XLS-R` or wav2vec-2.0 speech encoders on downstream ASR tasks (Baevski et al., 2020; Conneau, Baevski, et al., 2020).

Table 5.1: Amount of per language transcribed speech data in the CommonVoice-v7 dataset used for multimodal multilingual training of SAMU-XLS-R speech encoder.

Lang	**EN**	**DE**	**CA**	**FR**	**ES**
Dur [Hrs]	2K	960	790	740	380
Lang	**FA**	**IT**	**CY**	**TA**	**RU**
Dur [Hrs]	290	290	220	200	150
Lang	**PL**	**ZH_HK**	**NL**	**PT**	**AR**
Dur [Hrs]	130	96	93	85	84
Lang	**ZH_CN**	**ZH_TW**	**SV_SE**	**ET**	**TR**
Dur [Hrs]	63	59	34	32	32
Lang	**JA**	**ID**	**MN**	**SL**	**LV**
Dur [Hrs]	27	25	12	9	7

We use the cosine distance between the speech and the text embedding as the training loss (Equation 5.1). We do not update the weights of the LaBSE text encoder throughout training. The reason for this design choice is straightforward. LaBSE's sentence embedding space is already semantically aligned across 109 languages. By fine-tuning LaBSE along with SAMU-XLS-R on transcribed speech data $\mathcal{D}$, we run the risk of destroying this alignment. In fact, LaBSE will have no incentive to maintain an aligned embedding space. Instead, our learning framework attempts to embed speech utterances in the LaBSE's sentence embedding space to make it multimodal. By forcing the speech embeddings outputted by SAMU-XLS-R to be closer to LaBSE text embedding, we get the cross-lingual semantic alignments between speech utterances in different languages and text in 109 languages without ever encountering cross-lingual associations during the model's training. It might be possible to train the LaBSE text encoder along with SAMU-XLS-R and still maintain the LaBSE's semantically aligned embedding space. But, it is out-of-scope of this work.

5.3.3 SAMU-XLS-R Model Card

Table 5.2 summarizes the best configuration of different hyperparameters for training SAMU-XLS-R encoder. Next, we explain what some parameters in the table mean. CoVo_25 refers to the multilingual transcribed speech data used for training the model. We use data in 25 languages from the CoVo dataset. CNN Feature Extractor refers to the pre-trained XLS-R's convolutional encoder that maps the 1D speech waveform to a 2D feature representation used as input to the transformer encoder. We keep its weights fixed to the pre-trained value. Freeze Fine-tune updates refer to the number of training iterations to which we only train the projection layer of SAMU-XLS-R. See Equation 5.3 and the text above it for details on the smoothing factor α. The learning rate scheduler (LR scheduler) has a value of 10-40-50 refers to the learning rate scheduler mentioned in Section 5.3. Training teacher is LaBSE, which refers to the fact that the training targets for SAMU-XLS-R are the embedding vectors corresponding to the text transcripts provided by LaBSE. The model supports 25 spoken languages and 109 written languages since SAMU-XLS-R is trained on the transcribed speech from 25 languages and LaBSE can encode text in 109 languages in its semantically aligned cross-lingual vector space.

5.4 Evaluation

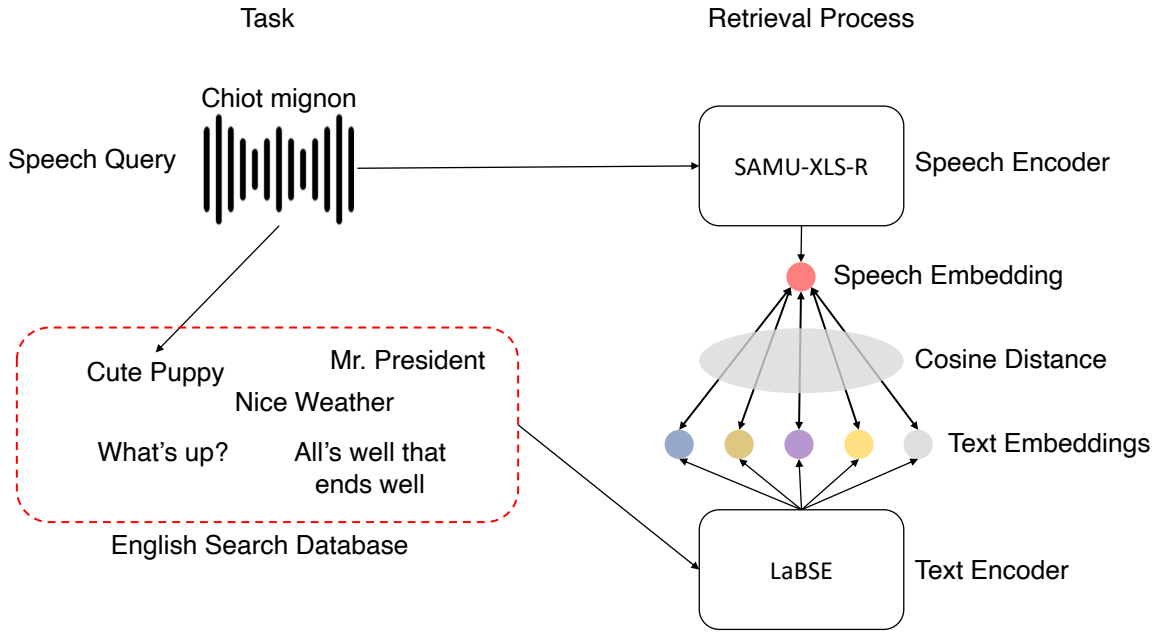
5.4.1 Task Overview

We evaluate our multimodal framework, consisting of SAMU-XLS-R, a speech embedding model, and LaBSE, a text embedding model, on several downstream translation retrieval tasks. An illustration of the retrieval task and pipeline is shown in Fig. 5-7. Retrieval is a common way to evaluate multilingual semantically aligned sentence embedding vector spaces in Natural language processing (Schwenk and Douze, 2017b; Feng et al., 2020). As mentioned, our work aims to learn a semantically aligned cross-lingual multimodal (joint speech-text) embedding space. Hence, if we successfully achieve our desired goal, the SAMU-XLS-R-LaBSE combination should perform well on

Table 5.2: SAMU-XLS-R model card. We summarize the best configuration of different hyperparameters for training the SAMU-XLS-R speech encoder.

Parameters	**Value**
Training Data	CoVo_25
Smoothing factor (α) for data re-balancing	0.05
Training updates	200K
Freeze Fine-tune updates	10k
CNN Feature Extractor	Frozen
Optimizer	Adam
max learning rate (LR)	1e-4
LR scheduler	10-40-50
batch size / GPU	2Hrs
Data Augmentation	Masking **h**
Training Objf.	Cosine Distance
Training Teacher	LaBSE
Pooling Fn.	Self-Attention
Model init.	XLSR Pre-Trained checkpoint
Num. GPUs	32
Supported Spoken Langs	22
Supported text Langs	109

Figure 5-7: Semantic Retrieval task definition and pipeline. (Left) We show an example of a retrieval task. Given a speech query in some language (French), the goal is to retrieve its corresponding text translation in English from a database of English sentences. (Right) We show the retrieval pipeline. We transform the speech query into an embedding using a pre-trained SAMU-XLS-R speech encoder. LaBSE transforms English sentences in the search database into embeddings. We compute the cosine distance between the query embedding and all the English sentence embeddings and pick the one with the smallest distance as the translation of the speech query.



cross-lingual speech-to-text translation retrieval tasks. Also, SAMU-XLS-R alone should be able to perform well on cross-lingual speech-to-speech translation retrieval tasks.

Next, we summarize the retrieval process, evaluation metrics, and the speech-to-text and speech-to-speech translation retrieval tasks we use to evaluate the SAMU-XLS-R's joint speech-text semantic embedding space.

5.4.2 Retrieval process and Evaluation Metrics

We construct two databases (DB), query and search, to perform translation retrieval. The query DB consists of speech utterances in language X, and in the case of text translation retrieval tasks, the search DB consists of text sentences in language Y. The task is to retrieve the correct text translation from the search DB corresponding to

each speech query in the query DB. To that end, we transform the speech utterances in the query DB through `SAMU-XLS-R` to query speech embedding matrix $Q \in \mathbb{R}^{N \times 768}$, where N is the number of speech queries in the query DB. Also, we transform the sentences in the search DB through the `LaBSE` encoder to search text embedding matrix $S \in \mathbb{R}^{M \times 768}$, where M is the number of sentences in the search DB. Given that the vectors are normalized, we could retrieve the text translations for the speech queries as follows:

$$A = QS^T$$

$$\mathbf{r} = \text{argmax}_j A_{:,j}$$

where $A \in \mathbb{R}^{N \times M}$ is the cosine similarity matrix, whose $(i,j)^{th}$ element $A_{i,j}$ is the cosine similarity between the speech query embedding $q_i \in Q$ and the sentence embedding $s_j \in S$, and $\mathbf{r} \in \mathbb{R}^N$ is the index vector, such that it's every component $r_i \in \mathbf{r}$ is the index of the closest match in the text translation search DB. Also, given the index vector $\mathbf{u}$, where each component $u_j \in \mathbf{u}$ is the index of the ground-truth text translation in the search DB, we compute the model's retrieval accuracy as follows:

$$\text{ACC} = 100 * \frac{\sum_{i=1}^{N} 1\{r_i = u_i\}}{N} \tag{5.4}$$

where the function $1\{r_i = u_i\}$ returns one when $r_i = u_i$, the predicted translation index matches the ground-truth translation index. Otherwise, it outputs zero. Hence, the numerator is the number of queries for which the model retrieved the correct translations from the search DB, and the denominator is the total number of queries in the query DB.

We refer to the retrieval accuracy in Equation 5.4 as Recall@1 or R@1, which contrasts with another similar metric, R@5, where the indicator function returns one if any of the top five retrieved search DB indices matches with the correct index. We report R@5 for speech retrieval evaluation tasks. The recall is commonly used to evaluate audio-visual multimodal representation learning models (Harwath, Torralba,

and J. Glass, 2016; Harwath, Hsu, and J. Glass, 2020; Rouditchenko et al., 2020).

In addition to R@1, for text translation retrieval tasks, we also report the Word Error Rate (WER) (Wikipedia contributors, 2020) between the retrieved and the ground-truth text translation. The reason is that it is hard to interpret retrieval accuracies. For example, the WER for model A with a retrieval accuracy of 70% might not be much worse than the WER for model B with a retrieval accuracy of 80% because model A might be worse than model B in retrieving the exact translations. However, it might still recover translations with a significant string overlap with the actual translation. The retrieval accuracy will fail to capture this.

5.4.3 Retrieval Tasks

X→EN Text Translation Retrieval

We use the CoVoST-2 (Changhan Wang, Pino, et al., 2020) X-EN speech-translation dataset for this evaluation task. The speech query DB is in a language X∈{RU, IT, FR, ES, TR, DE, ET, CY, NL, ID, CA, FA, AR, ZH, SV, MN, SL, JA, TA, LV} and the search DB consists of English sentences. To construct the speech query DB for each language X, we use the combined testing and development sets (henceforth, eval set) from CoVoST-2. To construct the search DB, we combine the English text translation from all the 22 X→EN eval sets in CoVoST-2, which we refer to as S_a. In addition, we create a search DB S_b that contains approximately 1.4M English sentences from the CoVo English transcribed speech data. We use the combined search DB $S = S_a \cup S_b$ for all the 22 X→EN text translation retrieval tasks. We add S_b to S_a to make the retrieval task harder than if we search over S_a.

EN→Y Text Translation Retrieval

We use the publicly available CoVoST-2 corpora (Changhan Wang, Pino, et al., 2020) for this evaluation task, which consists of English speech queries paired with their text translations. The speech query DB is in English, and the search DB is in a language Y∈{DE, CA, ZH, FA, ET, MN, TR, AR, SV, LV, SL, TA, JA, ID, CY}.

For each EN→Y retrieval task, the query DB consists of speech utterances in the combined development and testing sets. The search DB consists of the ground-truth text translations in language Y. corresponding to the speech queries. In addition, we add the Y language text translations available in the EN→Y CoVoST-2 training set to make the retrieval task harder. Similarly, we create a search DB for each of the 15 languages (Y) for the EN→Y text translation retrieval task.

We also retrieve text translation for this evaluation scenario on the MUST-C (Mattia A. Di Gangi et al., 2019b) EN→Y corpora. In MUST-C, we have English speech queries paired with their actual text translation in a language Y∈{ES, PT, FR, DE, Romanian (RO), NL, IT, Czech (CS), Vietnamese (VI), FA, TR, AR, RU, ZH}. We create an eval set, a union of MUST-C dev, *tst-COMMON*, and *tst-HE* data splits. The speech query DB consists of speech utterances in the eval set. The search DB for a language Y consists of sentences from the EN→Y MUST-C eval set combined with sentences from the EN→Y training set.

X→Y Text Translation Retrieval

We use the MTEDx (Salesky et al., 2021) speech-translation corpora, which consists of speech queries in language X paired with their ground-truth text translation. For this evaluation task, we have the translation pairs X_Y∈{IT_ES, IT_EN, ES_FR, ES_IT, FR_PT, ES_PT, FR_EN, PT_ES, ES_EN, PT_EN, RU_EN}. For a translation pair X_Y, we have speech queries in language X and the text search DB in language Y. For a retrieval X→Y, the query DB consists of speech utterances in the MTEDx X→Y eval set (dev+test), and the text search DB in language consists of the ground-truth text translations from the X→Y eval set and the X→Y training set. The reader might observe that the search DB is more significant than the query DB for all the text translation retrieval tasks and consists of the actual text translations and random sentences to make the retrieval task harder.

We consider MTEDx X→Y translation retrieval evaluation tasks as out-of-domain because we train `SAMU-XLS-R` on transcribed read speech from the CoVo dataset. At the same time, MTEDx consists of oratory-style speeches collected from TED talks.

Table 5.3: We perform **zero-shot** X→EN text translation retrieval on **In-domain** CoVoST-2 dataset. The search database for all X→EN retrieval tasks comprises 1.6 million English sentences. Below, we give the number of speech utterances in the query database for each retrieval task. The task is to retrieve the correct text translation for the speech queries in language X. We report the Retrieval accuracy (R@1) and the Word Error Rate between the ground truth and retrieved text translations. We compare our retrieval pipeline `SAMU-XLS-R-LaBSE`, with `ASR-LaBSE` and the Topline retrieval model. The `SAMU-XLS-R-LaBSE` retrieval pipeline transforms speech queries to embedding vectors using our `SAMU-XLS-R` speech encoder. Then, we match the query embedding vectors with the `LaBSE` text embeddings of the sentences in the search DB to retrieve the translation. The `ASR-LaBSE` retrieval pipeline first uses an ASR for language X to transcribe speech queries and then uses `LaBSE` to perform text-to-text translation retrieval. The Topline model uses the ground-truth text transcripts for the speech queries and performs text-to-text translation retrieval tasks using `LaBSE`.

X	**RU**	**IT**	**FR**	**ES**	**TR**	**DE**	**ET**	**CY**	**NL**	**ID**	**CA**	**FA**	**AR**	**ZH**	**SV**	**MN**	**SL**	**JA**	**TA**	**Avg.**
Query DB	12K	18K	30K	26K	3.3K	27K	3.1K	1.4K	3.4K	1.6K	25K	6.8K	3.5K	9.7K	2.9K	3.5K	870	1.3K	1.2K	-
`SAMU-XLS-R-LaBSE` Speech(X)→Text(EN) Retrieval																				
R@1[%]	93.5	92.9	92.5	92.9	93.4	90.9	91.5	84.6	89.7	84.4	82.1	83.6	73.7	78.6	72.4	68.2	52.1	48.9	42.4	76.8
WER[%]	2.6	3.0	3.5	3.6	3.7	4.7	4.8	5.1	4.9	9.5	11.0	10.2	13.8	15.2	19.0	26.0	32.4	44.7	57.7	17.2
`ASR-LaBSE` Speech(X)→Text(EN) Retrieval																				
R@1[%]	92.7	90.1	90.4	91.3	90.9	88.2	94.8	81.7	89.3	65.6	80.6	76.1	54.0	55.4	63.9	53.9	64.0	23.6	26.5	71.7
WER[%]	3.0	4.8	5.0	4.6	5.8	6.5	2.1	7.6	5.3	23.4	11.5	16.8	34.3	36.0	17.2	41.3	16.7	72.9	75.0	20.9
Topline `LaBSE` Text(X)→Text(EN) Retrieval																				
R@1[%]	94.4	94.0	94.8	94.3	94.2	93.2	97.5	86.2	90.8	91.3	83.8	85.1	74.5	81.4	87.0	81.3	70.9	83.1	49.2	85.2
WER[%]	2.0	2.5	1.9	2.6	2.9	2.8	0.4	4.1	4.2	2.5	9.9	8.7	13.5	12.8	4.7	14.4	10.2	9.4	51.7	8.7

X→EN Speech Translation Retrieval

Finally, we evaluate our model on speech translation retrieval tasks. We get the parallel X→EN speech-speech translation data from the publicly available VoxPopuli corpora (Changhan Wang, Riviere, et al., 2021). For this task, speech queries are in a language X∈{ES, FR, PL, NL, DE, RO, Croatian (HR), CS}, and the search DB consists of English speech translations corresponding to the queries. Unlike the text translation retrieval tasks, the search DB is the same size as the query DB and consists of only actual speech translations corresponding to the queries.

5.4.4 Baseline Retrieval Models

`ASR-LaBSE` retrieval pipeline

We also perform translation retrieval tasks using an `ASR-LaBSE` combination, where we convert the speech queries into text transcripts in the same language as the queries using an ASR model. Then, we perform ASR transcript-to-text translation retrieval using `LaBSE`. We build 25 language-specific ASR models to cover all the spoken languages in our text translation retrieval tasks. To construct the ASR models, we fine-tune the pre-trained `XLS-R` checkpoint on the downstream ASR task using the transcribed speech data in the target language available from the CoVo dataset (See Table 5.1 for the amount of per language transcribed speech data). We use the standard Connectionist temporal Classification (Graves, Fernández, et al., 2006) based optimization setup for fine-tuning the `XLS-R` model for the ASR task detailed in (Conneau, Baevski, et al., 2020). We use a beam size of 20 and a tri-gram character-level language model for decoding speech queries to text. We use the ESPnet speech recognition toolkit (Watanabe, Hori, Karita, Hayashi, Nishitoba, Unno, Enrique Yalta Soplin, et al., 2018; Watanabe, Boyer, Chang, Guo, Hayashi, Higuchi, Hori, W.-C. Huang, Inaguma, Kamo, et al., 2021) to construct the ASR models and decode them.

Table 5.4: We perform **zero-shot** EN→Y text translation retrieval on **In-domain** CoVoST-2 dataset. The search database for each EN→Y retrieval task consists of 320K sentences in language Y, and the query database consists of 31K English speech utterances. The task is to retrieve the correct text translation for the English speech queries. We report the Retrieval accuracy (R@1) and the Word Error Rate between the ground truth and retrieved text translations. We compare our retrieval pipeline `SAMU-XLS-R-LaBSE`, with `ASR-LaBSE` and the Topline retrieval model. The `SAMU-XLS-R-LaBSE` retrieval pipeline transforms speech queries to embedding vectors using our `SAMU-XLS-R` speech encoder. Then, we match the query embedding vectors with the `LaBSE` text embeddings of the sentences in the search DB to retrieve the translation. The `ASR-LaBSE` retrieval pipeline first uses an English language ASR to transcribe speech queries and then uses `LaBSE` to perform text-to-text translation retrieval. The Topline model uses the ground-truth text transcripts for the speech queries and performs text-to-text translation retrieval tasks using `LaBSE`.

Y	**ZH**	**SL**	**TR**	**LV**	**CY**	**ID**	**DE**	**CA**	**AR**	**SV**	**ET**	**TA**	**FA**	**JA**	**MN**	**Avg.**
`SAMU-XLS-R-LaBSE` Speech(EN)→Text(Y) Retrieval																
R@1[%]	87.2	90.5	89.4	89.9	90.8	91.0	91.5	91.4	88.3	91.7	90.4	90.5	89.0	88.1	86.2	89.7
WER[%]	11.2	6.3	7.4	7.2	6.2	5.9	5.8	5.5	8.5	5.5	6.6	7.3	8.4	11.9	10.9	7.6
`ASR-LaBSE` Speech(EN)→Text(Y) Retrieval																
R@1[%]	87.9	90.6	89.8	90.2	90.7	91.2	91.4	91.6	89.0	91.7	90.5	91.2	89.6	88.4	87.3	90.1
WER[%]	10.7	6.2	7.1	6.9	6.2	5.7	5.8	5.3	7.8	5.4	6.5	6.5	7.7	11.5	9.8	7.3
Topline `LaBSE` Text(EN)→Text(Y) Retrieval																
R@1[%]	95.8	97.1	96.2	96.6	96.7	96.8	97.1	96.9	95.7	97.3	96.7	97.0	95.4	95.5	94.5	96.4
WER[%]	2.7	1.3	1.9	1.8	1.7	1.5	1.5	1.3	2.3	1.3	1.6	1.8	2.8	4.2	3.5	2.1

Topline

As a topline, we use the ground-truth transcriptions corresponding to speech queries and perform ground-truth transcription to text translation retrieval using `LaBSE`. Our `SAMU-XLS-R-LaBSE` retrieval framework cannot perform better than the topline because the best we can do with our proposed multimodal learning framework is to match the `LaBSE` embedding vectors perfectly.

5.4.5 Results

X→EN speech-to-text translation retrieval

Table 5.3 shows the results on X→EN translation retrieval tasks using `SAMU-XLS-R-LaBSE`, `ASR-LaBSE` and Topline `LaBSE` retrieval pipelines. We report the retrieval accuracy (R@1) and WERs for different spoken languages X. The task is to retrieve the English text translation for a given speech query (X). The table shows the number of speech queries per spoken language X. The number of speech queries in the evaluation set varies across languages, with more queries for high-resource and fewer for low-resource languages. It is a function of the evaluation set available for different languages in the CoVoST-2 eval set. The search for the English translation is over a text database comprising 1.6M English sentences. The text DB contains the ground-truth English translations and transcriptions from the CommonVoice English dataset. We added the extra English sentences to make the translation retrieval task harder than searching over a small database of only true English translations. See Section 5.4.3 for more details on X→EN retrieval tasks.

Interestingly, `ASR-LaBSE` is significantly worse than `SAMU-XLS-R-LaBSE` retrieval model on retrieval tasks where the speech queries are in non-European languages. For example, on ID→EN, FA→EN, AR→EN, ZH→EN, MN→EN, JA→EN and TA→EN retrieval tasks, `SAMU-XLS-R-LaBSE` achieves a WER of 9.5%, 10.2%, 13.8%, 15.2%, 26.0%, 44.7% and 57.7% respectively compared to 23.4%, 16.8%, 34.3%, 36.0%, 41.3%, 72.9%, 75.0% respectively by `ASR-LaBSE`. On average, `SAMU-XLS-R-LaBSE` achieves an average WER of 22.6% compared to 33.7% with `ASR-LaBSE` on

Table 5.5: We perform **zero-shot** EN→Y text translation retrieval on **Out-of-domain** MUST-C dataset. The search database for each EN→Y retrieval task consists of approximately 200K sentences in language Y, and the query database consists of about 4K English speech utterances. The task is to retrieve the correct text translation for the English speech queries. We report the Retrieval accuracy (R@1) and the Word Error Rate between the ground-truth and retrieved text translations. We compare our retrieval pipeline `SAMU-XLS-R-LaBSE`, with `ASR-LaBSE` and the Topline retrieval model. The `SAMU-XLS-R-LaBSE` retrieval pipeline transforms speech queries to embedding vectors using our `SAMU-XLS-R` speech encoder. Then, we match the query embedding vectors with the `LaBSE` text embeddings of the sentences in the search DB to retrieve the translation. The `ASR-LaBSE` retrieval pipeline first uses an English language ASR to transcribe speech queries and then uses `LaBSE` to perform text-to-text translation retrieval. The Topline model uses the ground-truth text transcripts for the speech queries and performs text-to-text translation retrieval tasks using `LaBSE`.

Y	**DE**	**PT**	**FR**	**DE**	**RO**	**NL**	**IT**	**CS**	**VI**	**FA**	**TR**	**AR**	**RU**	**ZH**	**Avg.**
`SAMU-XLS-R-LaBSE` Speech(EN)→Text(Y) Retrieval															
R@1[%]	87.4	88.2	87.1	86.8	87.3	86.3	85.6	85.1	82.4	82.5	84.1	83.2	81.3	77.8	84.6
WER[%]	7.0	6.8	7.3	7.4	7.5	7.8	8.5	10.1	10.2	12.3	11.7	13.8	13.4	21.0	10.3
`ASR-LaBSE` Speech(EN)→Text(Y) Retrieval															
R@1[%]	88.8	88.6	88.4	87.9	87.5	87.0	86.6	86.4	83.0	83.8	84.5	83.8	82.7	79.0	85.6
WER[%]	6.2	6.5	6.4	6.7	7.4	7.2	7.7	8.9	9.8	10.3	11.1	13.2	12.3	20.6	9.6
Topline `LaBSE` Text(EN)→Text(Y) Retrieval															
R@1[%]	96.1	96.0	96.1	95.9	95.7	95.3	95.1	95.1	92.9	91.4	92.7	92.4	92.3	87.6	93.9
WER[%]	1.8	1.9	1.8	1.8	2.2	2.1	2.5	3.2	3.1	6.1	5.2	6.5	5.0	10.0	3.8

Table 5.6: We present results on **Out-of-domain** MTEDx X→Y text translation retrieval tasks. For a retrieval task X_Y, the speech queries are in language X, and the search DB consists of sentences in language Y. The task is to retrieve the correct text translation for each speech query. We report the Retrieval accuracy (R@1) and the Word Error Rate between the ground-truth and retrieved text translations. We compare our retrieval pipeline `SAMU-XLS-R-LaBSE`, with `ASR-LaBSE` and the Topline retrieval model. The `SAMU-XLS-R-LaBSE` retrieval pipeline transforms speech queries to embedding vectors using our `SAMU-XLS-R` speech encoder. Then, we match the query embedding vectors with the `LaBSE` text embeddings of the sentences in the search DB to retrieve the translation. The `ASR-LaBSE` retrieval pipeline first uses an ASR model for language X to transcribe speech queries and then uses `LaBSE` to perform text-to-text translation retrieval. The Topline model uses the ground-truth text transcripts for the speech queries and performs text-to-text translation retrieval tasks using `LaBSE`.

X_Y	**IT_ES**	**IT_EN**	**ES_FR**	**ES_IT**	**FR_PT**	**ES_PT**	**FR_EN**	**PT_ES**	**ES_EN**	**PT_EN**	**RU_EN**	**Avg.**
Query DB	1.8K	2K	1.8K	270	2K	1.8K	2K	2K	1.8K	2K	1.8K	-
Search DB	1.6M	270K	220K	250K	270K	210K	1.6M	1.6M	1.6M	210K	270K	-
`SAMU-XLS-R-LaBSE` Speech(X)→Text(Y) Retrieval												
R@1[%]	92.2	87.0	87.5	84.5	86.7	85.9	81.0	80.8	78.6	74.6	61.2	81.8
WER[%]	2.8	5.7	6.2	6.3	6.3	6.4	8.3	9.4	9.6	12.4	26.0	9.0
`ASR-LaBSE` Speech(X)→Text(Y) Retrieval												
R@1[%]	92.5	88.2	90.2	85.7	87.1	88.3	82.9	84.7	82.2	80.1	69.9	84.7
WER[%]	2.4	4.4	4.2	6.1	5.5	4.5	6.9	7.1	7.4	8.8	17.0	6.8
Topline `LaBSE` Text(X)→Text(Y) Retrieval												
R@1[%]	96.1	93.3	94.4	91.7	93.5	94.1	90.9	94.8	90.5	91.7	87.6	92.6
WER[%]	1.0	2.3	1.8	2.9	2.0	1.7	2.9	1.3	3.2	2.6	5.4	2.5

non-European spoken languages (X)→EN translation retrieval tasks. On retrieval tasks, where speech queries are in European languages, SAMU-XLS-R-LaBSE performs at par with ASR-LaBSE retrieval pipeline. For example, on RU→EN, IT→EN, FR→EN, ES→EN, DE→EN, ET→EN, CY→EN, NL→EN, CA→EN, SV→EN, SL→EN and LV→EN translation retrieval tasks, SAMU-XLS-R-LaBSE achieves an average WER of 13.6% compared to 10.2% with ASR-LaBSE retrieval pipeline. These results are unsurprising given that the ASR system is generally better for European languages (high and low-resource) than for non-European languages. This is since the XLS-R speech encoder, which we fine-tune on downstream ASR tasks using language-specific transcribed data, is pre-trained majorly on European language speech data.

Finally, the topline model uses the ground-truth text transcriptions corresponding to the speech queries (X) to retrieve the English text translations. This model uses only LaBSE to perform the text(X)→text(EN) retrieval task. The topline achieves an average WER of 14.5% on non-European languages X and 4.9% on European languages, which implies that we could not quite reach the topline performance with our SAMU-XLS-R-LaBSE retrieval pipeline, and there is room for improvement. We believe that increasing the scale of the training data and using contrastive loss for training SAMU-XLS-R could result in improved performance. However, a training setup with a contrastive loss would require considerable engineering effort because of the engineering complexity involved in mining negative samples across GPUs as done for training LaBSE (Feng et al., 2020). Drawing negative samples from the same GPU device would not be sufficient because of the small per GPU batch size owing to the large speech encoder size and long speech waveforms. Hence, we leave the exploration of contrastive learning for future work.

EN→Y speech-to-text translation retrieval

Table 5.4 and 5.5 shows the results on EN→Y speech→text retrieval tasks using SAMU-XLS-R-LaBSE, ASR-LaBSE and Topline LaBSE retrieval pipelines. We retrieve the text translation in a language Y for a given speech query in English for the EN→Y retrieval tasks. In the results table, first, we show the number of English speech

queries and the sentences in the search database for each language, Y.

For the CoVoST-2 EN→Y retrieval tasks, we have 32K English speech queries in the query DB and 320K sentences in the search DB in language Y for each EN→Y retrieval task. See Section 5.4.3 for more details on the EN→Y CoVoST-2 retrieval tasks.

Table 5.4 shows results on CoVoST-2 EN→Y retrieval tasks. We have 32K English speech queries in the query DB and 320K sentences in the search DB in language Y for each EN→Y retrieval task. See Section 5.4.3 for more details on the EN→Y CoVoST-2 retrieval tasks. We observe that `SAMU-XLS-R-LaBSE` and `ASR-LaBSE` retrieval pipelines perform at par, achieving a retrieval WER of 7.6% and 7.3% respectively. In contrast, the Topline `LaBSE` text(EN)→text(Y) retrieval pipeline achieves an average WER of 2.1% across the 15 retrieval tasks. There is room for improvement. In particular, for retrieving text translations in non-European languages such as ZH, MN, JA, FA, AR, and TA, for which the average WER achieved by our proposed `SAMU-XLS-R-LaBSE` retrieval pipeline is 9.7% compared to 2.8% with the topline `LaBSE` text(EN)→text(Y) retrieval. For European languages, our retrieval model achieves a WER of 6.1% compared to 1.7% for the topline model. Our model performs better in European languages (6.1% WER) than non-European languages (9.7% WER).

Table 5.5 shows EN→Y retrieval results on the out-of-domain MUST-C evaluation corpus. We have the same number of 4K speech utterances in the query DB and 200K sentences in the search DB for all text translation retrieval tasks. We observe that `SAMU-XLS-R-LaBSE` perform at par with `ASR-LaBSE` retrieval pipeline, achieving an average of 10.3% WER compared to 9.6% achieved by the `ASR-LaBSE` retrieval pipeline on the 14 EN→Y retrieval tasks. Our model achieves a WER of less than 10% for most languages except TR, AR, RU, and ZH, for which the model achieves a WER of 11.1%, 13.2%, 12.3%, and 20.6%, respectively. These WERs are approximately double the WERs, achieved by the topline `LaBSE` text(EN)→text(Y) retrieval model. However, the WERs are at a respectable less than 20% mark.

Table 5.7: We perform **zero-shot** X→EN speech translation retrieval on the VoxPopuli dataset. The speech queries are in a language X, and the search database consists of speech utterances that are translations of speech queries. Unlike text translation retrieval tasks, where the search DB is much bigger than the query DB, here, the search and the query DB have the same size. During its training, SAMU-XLS-R had no access to cross-lingual speech-to-speech associations. Hence, semantic alignment among speech utterances in different languages is an emergent property of the embedding vector space learned by SAMU-XLS-R via our proposed multimodal learning framework. We compare SAMU-XLS-R's vector space with XLS-R.

X	**ES**	**FR**	**PL**	**NL**	**DE**	**RO**	**HR**	**CS**	**Avg.**
SAMU-XLS-R Speech(X)→Speech(EN) Retrieval									
Query & Search DB	36K	50K	19K	11K	60K	16K	8K	11K	-
R@1[%]	97.9	97.8	97.7	97.5	96.0	76.0	53.3	52.8	83.6
R@5[%]	98.5	98.4	98.4	98.0	97.1	80.9	59.5	58.2	86.1
XLS-R Speech(X)→Speech(EN) Retrieval									
R@1[%]	-	-	-	-	0.0	-	-	-	0.0

X→Y speech-to-text translation retrieval

Table 5.6 shows results on out-of-domain MTEDx X→Y text translation retrieval tasks using SAMU-XLS-R-LaBSE, ASR-LaBSE and topline LaBSE retrieval pipelines. The table shows the speech queries and text search database combination for each pair X_Y. We observe that SAMU-XLS-R-LaBSE achieves an average retrieval WER of 9% compared to 6.8% with ASR-LaBSE and 2.5% with topline LaBSE on the 11 text translation retrieval tasks. It is unsurprising that ASR-LaBSE retrieval pipeline performs better than the SAMU-XLS-R-LaBSE model. Because the speech queries for X→Y retrieval tasks are in European languages and our European language ASR models are quite good. The results reported here confirm the observation we made for X→EN CoVoST-2 translation retrieval tasks, where SAMU-XLS-R-LaBSE performed better than ASR-LaBSE for non-European languages but not for the European languages. If we had an ASR model that generated text transcripts that matched the ground-truth transcripts, then the performance of ASR-LaBSE would be the same as that of the

topline model.

X→EN speech-to-speech translation retrieval

The SAMU-XLS-R speech encoder learns a semantically aligned vector space across several spoken languages. The model can retrieve the correct English speech translations corresponding to speech queries in a language X with above 96% accuracy for X∈{ES, FR, PL, NL, DE}. For X∈{RO, HR, CS}, SAMU-XLS-R's speech translation retrieval performance lags behind other languages. This result is unsurprising because SAMU-XLS-R did not see any transcribed data from these three languages during training. SAMU-XLS-R achieves an average retrieval R@1 accuracy of 83.6% across the 8 X→EN speech translation retrieval tasks. On the other hand, XLS-R fails on this retrieval task. To get an utterance-level speech embedding from XLS-R, we perform temporal mean pooling of the contextual frame-wise embeddings from the last layer of the model. The poor retrieval results show that the XLS-R representation space is not semantically aligned across different languages. We achieve similarly poor results with representations from different XLS-R layers.

5.5 Analysis

This section studies various design decisions that went into creating the SAMU-XLS-R speech encoder.

Loss and pooling functions. While detailing SAMU-XLS-R in Section 5.2.2, we mentioned that we use the Self-Attention pooling method to construct an utterance-level speech embedding from acoustic frame-level contextual embedding vectors. Also, we use the cosine distance loss for training SAMU-XLS-R. Table 5.8 shows that combining cosine distance loss and the Self-Attention pooling method is better than combining other loss functions and pooling methods. We train SAMU-XLS-R with L1, L2, and cosine distance losses and compare its average text translation retrieval performance across the 21 X→EN CoVoST-2 retrieval tasks. Also, we compare the retrieval perfor-

Table 5.8: Avg. retrieval Performance in terms of retrieval accuracy (R@1), and WER between the retrieved translations and the ground truth translations, on 21 X→EN text translation retrieval tasks for different combinations of loss and pooling functions. We train SAMU-XLS-R with L1, L2, and cosine distance losses and compare its average text translation retrieval performance across the 21 X→EN CoVoST-2 retrieval tasks. Also, we compare the retrieval performance with Mean, Max, and Self-Attention pooling strategies. Three loss functions with three pooling strategies lead to nine possible training configurations

Loss	Pooling	R@5 [%]	R@1 [%]	WER [%]
L1	Max	52.2	44.0	50.9
L1	Mean	52.9	44.6	49.9
L1	Att.	54.0	45.6	48.8
Cos	Max	55.4	46.6	47.5
L2	Max	55.6	46.8	47.3
Cos	Mean	56.3	47.6	46.2
L2	Mean	57.2	48.2	45.4
L2	Att.	57.6	48.6	45.3
Cos	**Att.**	**58.0**	**48.8**	**44.6**

mance with Mean, Max, and Self-Attention pooling strategies. Three loss functions with three pooling strategies lead to nine possible training configurations. For quick analysis, we train SAMU-XLS-R on 8 V100-32GB GPUs for 100K iterations on a subset $\mathcal{D}_S$ of the complete multilingual transcribed training data $\mathcal{D}$. $\mathcal{D}_S$ is constructed by randomly sampling 400K training examples from $\mathcal{D}$. SAMU-XLS-R with the Self-Attention pooling method and trained with cosine distance loss reaches an average retrieval R@1 accuracy of 48.8%, better than the other eight training configurations.

Data Re-balancing Smoothing parameter α. This section studies the effect on the model's average retrieval performance across 21 X→EN retrieval tasks when we train the model with re-balanced training data according to Equation 5.3. The smoothing parameter α is the only hyper-parameter in the data re-balancing equation. First, we construct several re-balanced multilingual transcribed speech datasets

Table 5.9: Avg. retrieval performance measured using Retrieval Accuracy (R@1) and WER between the retrieved and ground-truth translations on 21 X→EN text translation retrieval tasks for different values of α.

α	R@5 [%]	R@1 [%]	WER [%]
1.00	58.0	48.8	44.6
0.70	70.3	60.5	32.2
0.30	79.3	69.5	22.8
0.10	81.6	71.7	20.5
0.01	81.9	72.0	19.9
0.05	**82.2**	**72.4**	**19.6**

Table 5.10: Avg. retrieval performance measured using Retrieval Accuracy (R@1) and WER between the retrieved and ground-truth translations on 7 X→EN low-resource text translation retrieval tasks for different αs.

α	R@5 [%]	R@1 [%]	WER [%]
1.00	32.1	23.8	72.1
0.05	**71.9**	**61.4**	**29.7**

corresponding to different values of α. Then, we randomly sample 400K utterances from re-balanced datasets for `SAMU-XLS-R` model training. We train `SAMU-XLS-R` using the cosine distance loss function for 100K iterations on 8 V100-32GB GPUs.

We observe in Table 5.9 that the models trained with re-balanced data ($\alpha < 1.0$) achieve significantly better average retrieval accuracy across the 21 X→EN text translation retrieval tasks than the model trained with no re-balancing ($\alpha = 1.0$). We achieve the best performance with $\alpha = 0.05$, where the model's average retrieval accu-

Table 5.11: Avg. retrieval performance measured using Retrieval Accuracy (R@1) and WER between the retrieved and ground-truth translations on five high-resource X→EN text translation retrieval tasks for different αs.

α	R@5 [%]	R@1 [%]	WER [%]
0.05	92.0	85.0	9.4
1.00	**93.8**	**87.5**	**7.3**

Table 5.12: Avg. retrieval performance measured using Retrieval Accuracy (R@1) and WER between the retrieved and ground-truth translations on 21 X→EN text translation retrieval tasks for different training data. T1 refers to using multilingual transcribed speech data for training SAMU-XLS-R, T2 refers to SAMU-XLS-R training on paired speech-text translation data, and T3 refers to SAMU-XLS-R training on combined transcribed and translated speech data.

Model	R@5 [%]	R@1 [%]	WER [%]
SAMU-XLS-R_T2	49.9	41.3	54.6
SAMU-XLS-R_T3	79.7	69.5	22.7
SAMU-XLS-R_T1	**82.2**	**72.4**	**19.6**

Table 5.13: Avg. retrieval performance measured using Retrieval Accuracy (R@1) and WER between the retrieved and ground-truth translations on 7 X→EN high-resource text translation retrieval tasks for different training data. T1 refers to using multilingual transcribed speech data for training SAMU-XLS-R, T2 refers to SAMU-XLS-R training on paired speech-text translation data, and T3 refers to SAMU-XLS-R training on combined transcribed and translated speech data.

Model	R@5 [%]	R@1 [%]	WER [%]
SAMU-XLS-R_T2	15.5	9.2	91.4
SAMU-XLS-R_T3	67.3	55.7	36.1
SAMU-XLS-R_T1	**71.9**	**61.4**	**29.7**

racy R@1 is 72.4% compared to 48.8% achieved by SAMU-XLS-R trained on the original dataset without any re-balancing. The massive boost in retrieval performance is due to the model doing much better on X→EN retrieval tasks where speech queries are in low-resource languages, which implies that the model was indeed under-fitting on low-resource languages due to the data imbalance in the training set of SAMU-XLS-R. Table 5.10 shows that SAMU-XLS-R trained with data re-balancing ($\alpha = 0.05$) achieves an average retrieval R@1 accuracy of 61.4% compared to 23.8% achieved by SAMU-XLS-R trained on the unbalanced training set ($\alpha = 1.0$). Also, Table 5.11 shows a negligible performance difference for different αs on X→EN tasks when speech queries are in high-resource languages.

Training Data. In Section 5.3.1, we mention that we train SAMU-XLS-R with multilingual transcribed speech data collected from the CoVo dataset. In this section, we study the effect of training SAMU-XLS-R with paired speech-translation data. We train SAMU-XLS-R using three different training datasets: 1) Transcribed multilingual speech in 25 languages from the CoVo dataset, which we refer to as the training setup T1, and the model trained with this setup as SAMU-XLS-R_T1, 2) The 22 X→EN CoVoST-2 (Changhan Wang, Pino, et al., 2020) speech-translation training sets, where speech utterances are paired with their corresponding English text translations. We refer to that as the training setup T2, and the model trained with this setup as SAMU-XLS-R_T2. 3) A combination of both T1 and T2. We refer to the model trained with this setup as SAMU-XLS-R_T3. Also, we re-balance the different training datasets using $\alpha = 0.05$ and randomly pick 400K examples for training. Finally, we train the model for 100K iterations on 8 V100-32GB GPUs.

Table 5.12 shows average retrieval performance on 21 X→EN retrieval tasks achieved by SAMU-XLS-R trained with the three different training setups mentioned above. We observe that SAMU-XLS-R_T1 achieves the best retrieval performance out of the three models, implying that we can train SAMU-XLS-R with multilingual transcribed speech. Furthermore, table 5.13 shows that SAMU-XLS-R_T1 is notably better for X→EN tasks when speech queries are in low-resource languages. The performance difference among the three models is negligible for speech queries in high-resource languages.

Qualitative Analysis. In Table 5.14, we show some examples of retrieved translations using our retrieval pipeline. We can see a semantic similarity between the speech query and the top-5 English translations retrieved using the joint speech-text embedding space learned by our proposed multimodal learning framework (Section 5.2).

5.6 Chapter Summary

In this chapter, we proposed a semantically-aligned multimodal (joint speech-text) utterance-level cross-lingual speech representation (SAMU-XLS-R) learning framework. Using multilingual transcribed speech to train the proposed representation learning model, we show that cross-lingual alignments between speech utterances and their text and speech translations emerge in the learned joint speech-text embedding vector space.

We show that unlike XLS-R (a speech-only multilingual speech encoder), SAMU-XLS-R in combination with language-agnostic BERT sentence encoder LaBSE can perform zero-shot speech-to-text and speech-to-speech translation retrieval across several spoken and written languages. Furthermore, we show that SAMU-XLS-R performs at par with XLS-R on sequence-to-sequence modeling tasks such as ASR and Phoneme Recognition. In the future, we will extend our multimodal learning framework for zero-shot speech translation and large-scale speech-to-text data mining to create parallel speech-text translation datasets for training speech translation models.

Table 5.14: Given a speech query in language X, we search over a large English database of 1.6M sentences to retrieve the top-5 translations using our proposed `SAMU-XLS-R-LaBSE` retrieval pipeline. We randomly pick five speech queries from the CoVoST-2 eval set, two in French and one each in German, Arabic, and Spanish. For each speech query, we retrieve the top-5 English translations.

Speech Query	**Query Lang.**	**Top-5 Retrieved EN Translations**
La chute de la cité est difficile à expliquer.	FR	1) **The fall of the city is difficult to explain** 2) The origin of the town name is unclear. 3) It's not easy to describe why it happened. 4) Further history of the village is unclear. 5) The origin of the town is not completely clear.
Elle est le chef-lieu du département de l'Okano.	FR	1) It is the seat of Okanogan County. 2) **It is the main city of the Okano District.** 3) It is the county seat of Macon County. 4) It is the capital of Otwock County. 5) Its county seat is Oconto.
Die Blütezeit reicht von März und April vor der Bildung der Laubblätter	DE	1) **The flowering season lasts from March** **until April, just before foliage develops.** 2) The flowering period extends from April through June. 3) Flowering occurs from April through July. 4) Its flowering season is around February to April. 5) The blooming starts in the middle of April and goes almost until mid May.
تزداد جمالاً يوماً بعد يو.	AR	1) She's getting worse every day. 2) It is getting better every day. 3) It's getting warmer day after day. 4) **She gets prettier every day.** 5) It's getting colder day after day.
Fue enfermera voluntaria en la I Guerra Mundial.	ES	1) **She was a volunteer nurse on World War I.** 2) Her mother was a nurse during World War One. 3) During World War One he served as a paramedic. 4) During World War One he was a medical sergeant 5) In World War One, she was a Red Cross nurse.

Chapter 6

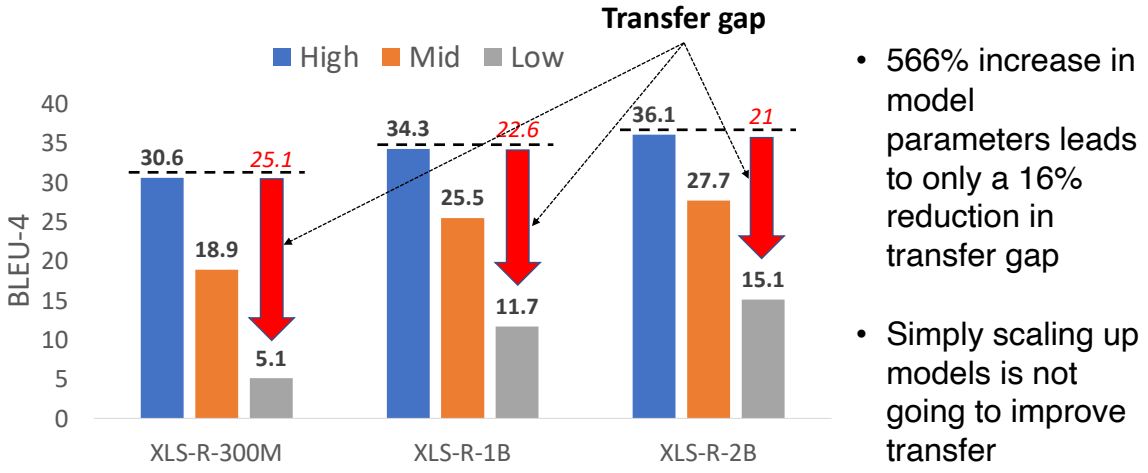
Multilingual Speech-To-Text Translation

In this chapter, we show an application of the SAMU-XLS-R semantic speech embedding model to multilingual speech-to-text translation. Multilingual Automatic Speech Translation refers to the problem of training a single model for several translation tasks. Often the training pool of translation tasks is imbalanced, where most of the training data come from a few high-resource tasks, while most other tasks have very few paired speech-text translation examples for training. We show that building multilingual translation models on top of the multimodal semantically aligned representations learned by SAMU-XLS-R (due to the joint speech-text embedding framework proposed in the previous chapter) leads to significantly better *cross-lingual task* transfer from high to low-resource translation tasks than using other multilingual representations learned by unimodal representation learning frameworks such XLS-R (Babu et al., 2021), and multimodal multilingual representation learning frameworks such as mSLAM (Bapna, Cherry, et al., 2022b; Bapna, Cherry, et al., 2022a).

To motivate our work in this chapter, we show the performance of the multilingual XLS-R speech encoder on the CoVoST-2 speech-to-text translation benchmark. CoVoST-2 comprises 21 X→EN speech-to-text translation tasks, where X refers to the language of speech utterance, and EN is its corresponding English text translation. XLS-R speech encoder is pre-trained via Self-Supervised Learning using unlabeled

speech data in 128 languages. Babu et al. (2021) fine-tunes the pre-trained XLS-R encoder combined with a pre-trained MBART text decoder (Y. Liu et al., 2020b) simultaneously on 21 X→EN translation tasks in CoVoST-2 benchmark. We categorize the 21 translation tasks into high, mid, and low-resource groups. A task is classified as high if it has more than 100 hours of paired speech(X)-text(EN) translation training data, mid if training data is between 10 and 100 hours, and low if training data is less than 10 hours.

Figure 6-1: We report translation performance on 21 X→EN speech-to-text translation tasks in CoVoST-2 benchmark with different sized pre-trained XLS-R encoders fine-tuned on labeled speech translation data. The 21 tasks are categorized into high, mid, and low resource tasks depending on the available labeled training data for a task. We report average BLEU-4 scores in the three categories. The important thing to consider is the performance gap (cross-lingual transfer gap) between high and low-resource tasks. We address this large gap in this chapter.



We report the XLS-R(speech encoder)→MBART(text decoder) transformer translation model on the high, mid, and low translation task groups in Fig. 6-1. We report the average BLEU-4 score on each translation group. The vital thing to observe is the performance gap (cross-lingual transfer gap) between high and low-resource translation groups for different-sized XLS-R encoders ranging from 300M to 2B parameters. So, increasing the model size from 300M to 2B, a more than 500% increase leads to only a 16% reduction in the cross-lingual transfer gap. Since the translation model is built on top of the pre-trained XLS-R encoder's representations, there is some miss-

ing ingredient in the pre-trained representations that leads to a poor cross-lingual translation task transfer. We hypothesize the missing piece is semantic knowledge. We hypothesize that since SAMU-XLS-R is specifically trained to encode semantic information about the speech signal in its internal representations, building a translation model on top of SAMU-XLS-R would lead to better cross-lingual transfer than reported in Fig. 6-1.

6.1 Introduction

Pre-trained speech encoders like `XLS-R` (Babu et al., 2021) are considered "foundation models" (Bommasani and al., 2021) for downstream multilingual speech processing applications such as Multilingual Automatic Speech Recognition (Conneau, Baevski, et al., 2020; Rivière et al., 2020; Babu et al., 2021), Multilingual Automatic Speech Translation (X. Li et al., 2020a; Babu et al., 2021; Bapna, Cherry, et al., 2022b), and other para-linguistic property prediction tasks (Shor et al., 2021; S.-w. Yang et al., 2021). This work focuses on Multilingual Automatic Speech Translation.

Multilingual Automatic Speech Translation (MAST) refers to translating speech in all the source languages in set $\mathcal{X}$ to text in all the target languages in set $\mathcal{Y}$, which implies a total of $|\mathcal{T}| = |\mathcal{X}| \times |\mathcal{Y}|$ translation tasks. In MAST, we train a single model for all the translation tasks given by set $\mathcal{T}$. The benefits of having a single model instead of an individual model for each task $t \in \mathcal{T}$ are two-fold: First, it is convenient to maintain and share a single model that can perform multiple tasks rather than having $|\mathcal{T}|$ separate models, and second, sharing model parameters amongst $|\mathcal{T}|$ translation tasks could lead to knowledge transfer across tasks, especially from high-resource to low-resource.

MAST's standard neural network architecture is the transformer *encoder-decoder* model (Vaswani et al., 2017). Recently, MAST has seen significant improvements owing to; (i) better initialization of the translation model's encoder and decoder with pre-trained speech encoders, like `XLS-R` (Babu et al., 2021), and text decoders, like `MBART` (Y. Liu et al., 2020b), (ii) better fine-tuning strategies (X. Li et al., 2020a), and

(iii) parallel speech-text translation corpora (Iranzo-Sánchez et al., 2019; Changhan Wang, Pino, et al., 2020). However, as we demonstrate later, the performance on low-resource tasks remains poor, and in particular, the gap between high- and low-resource tasks remains large. This is because the XLS-R encoder learns low-level linguistic knowledge like *phonetic knowledge* from unlabeled speech data, which is due to the lack of proper constraints during SSL pre-training for learning representations that encode high-level linguistic knowledge like *semantic knowledge*. Since semantic knowledge is language-agnostic, the representations that encode semantics should achieve better cross-lingual transfer in the downstream MAST task.

To inject semantic knowledge into the learned XLS-R representations, we turn to Semantically-Aligned Multimodal Cross-Lingual Representation Learning framework, SAMU-XLS-R (Khurana, Laurent, and J. Glass, 2022), introduced in the previous chapter. SAMU-XLS-R learns a multilingual multimodal semantically aligned speech representation space. This multilingual semantic information space learned by SAMU-XLS-R encoder could significantly benefit the task of MAST owing to the better cross-lingual transfer of the abstract semantic representations. This is evident from the significant cross-lingual task transfer of SAMU-XLS-R over the baseline XLS-R (Section 6.4.4).

In this chapter, we make the following **contributions**. We doubled the number of languages previously supported by SAMU-XLS-R encoder from 25 to more than 50. (Section 6.2). The SAMU-XLS-R encoder embeds speech at the utterance (a spoken sentence) level. In the previous chapter, we explored using SAMU-XLS-R embedding for translation retrieval. Also, other works explored the use of SAMU-XLS-R embeddings for spoken language understanding tasks (Laperrière et al., 2022). Differently, in this work, we explore the fine-grained representations the SAMU-XLS-R learns below the pooling layer. We show that the fine-grained representations (corresponding to a 20ms duration speech segment in the input speech waveform) are well-suited for the sequence generation task of Multilingual Automatic Speech Translation. Through several experiments, we empirically demonstrate the advantage of using SAMU-XLS-R on MAST over XLS-R.

On the public CoVoST-2 X→English MAST benchmark (Changhan Wang, Pino,

et al., 2020), we show (Section 6.4.4) that for MAST, by switching the XLS-R encoder in the transformer translation model to the SAMU-XLS-R encoder, the performance improves significantly on medium-resource X→EN tasks by 15.6 BLEU points, and on low-resource tasks by 18.9 BLEU points, and overall by 13.8 BLEU points. We also show SAMU-XLS-R's effectiveness in Zero-Shot settings. In this scenario, we only train the transformer translation model on high-resource X→EN translation tasks. We report a significant improvement on unseen (during training) medium and low-resource tasks of 18.8 and 11.9 BLEU points over the baseline.

We report results on Europarl, yet another speech-to-text translation MAST benchmark (Section 6.4.4). Europarl has significantly longer speech utterances than CoVoST-2 (See Section 2.4.2 for corpus comparisons). Europarl consists of 72 translation tasks of the form X→Y. We train the transformer translation model on a subset of translation tasks and evaluate the model on both unseen and unseen tasks. We observe an overall improvement of 8.5 BLEU points average with SAMU-XLS-R encoder over XLS-R on all translation tasks. The most significant improvement is observed on unseen tasks of 17 BLEU points.

6.2 Expanding SAMU-XLS-R to more Languages

Table 6.1: The number of hours of transcribed speech data available for training SAMU-XLS-R in each of the 53 languages from the CommonVoice-Version8 corpus.

ar	be	bg	ca	cs	cy	da	de	el	en	eo
85.2	903.9	8.2	916.8	54.9	116.3	6.6	1062.8	15.9	2185.8	1407.9
es	et	eu	fa	fi	fr	fy-NL	ga-IE	gl	ha	hi
404.6	33.0	98.9	317.3	8.5	826.1	49.6	4.3	10.2	3.4	11.7
hu	id	it	ja	ka	kmr	ky	lt	lv	mn	mt
19.9	25.8	310.6	40.8	7.6	47.0	37.2	17.4	7.1	12.4	8.3
nl	pl	pt	ro	ru	rw	sk	sl	sv-SE	sw	ta
98.0	142.2	112.0	15.8	162.6	2000.7	17.7	9.6	40.8	146.8	217.7
th	tr	tt	ug	uk	uz	vi	zh-CN	zh-HK	zh-TW	
142.1	65.1	29.2	59.8	63.4	81.0	4.5	68.0	99.7	62.6	

In the previous chapter, we trained SAMU-XLS-R on transcribed speech data in 25 languages. In this chapter, we extend the transcribed speech data to 53 languages for training SAMU-XLS-R collected from the CommonVoice-Version8 (CoVo-V8) corpus (Ardila et al., 2020). CoVo-V8 consists of transcribed speech in 87 languages (26 language families). Around 53 languages overlap with the language set supported by Language-Agnostic BERT Sentence Encoder (LaBSE), which provides semantic supervision for training the SAMU-XLS-R speech encoder. The 53 languages are: ar, be, bg, ca, cs, cy, da, de, el, en, eo, es, et, eu, fa, fi, fr, fy-NL, ga-IE, gl, ha, hi, hu, id, it, ja, ka, kmr, ky, lt, lv, mn, mt, nl, pl, pt, ro, ru, rw, sk, sl, sv-SE, sw, ta, th, tr, tt, ug, uk, uz, vi, zh-CN, zh-HK, zh-TW. Table 6.1 presents the number of hours of transcribed speech available in each of the 53 languages for SAMU-XLS-R training. The total training hours is 12.7K.

See Tables 2.13, and 2.14 for language code to language name mapping, and Tables 2.15, and 2.16 for detailed CoVo-V8 corpus statistics (other than hours of transcribed speech mentioned above) for each of the 53 languages. Also, See Section 5.2 for details about the multimodal learning framework for training SAMU-XLS-R speech encoder.

6.3 Translation Model

6.3.1 Overview

We train a transformer model for multilingual speech-to-text translation. We initialize the transformer encoder using the pre-trained SAMU-XLS-R speech encoder (Section 6.2), and the transformer decoder with the decoder of MBART transformer (Y. Liu et al., 2020b), a pre-trained text-to-text translation model. Initializing the transformer decoder of a speech-to-text translation model with MBART is first done in X. Li et al. (2020a). The translation model is trained using paired multilingual speech-text translation data.

The translation model's speech encoder transforms a speech waveform **a** to a con-

Algorithm 4 Computations performed by the decoder of the transformer translation model.

Input: Encoder output $\mathbf{c}$
Input: Ground-Truth Text Translation $\mathbf{y}$ during training
Output: $\log p(\mathbf{y}|\mathbf{c})$
$\mathbf{y} = \texttt{Embed}(\mathbf{y})$
for $i = 1$ to $L - 1$ **do**
    $\mathbf{y} = \texttt{TransformerLayer}^{(i)}(\mathbf{y})$
end for
$\mathbf{y} = \texttt{LN}(\texttt{TransformerLayer}^{(L)}(\mathbf{y}))$
$\mathbf{o} = \texttt{FCProject}(\mathbf{y})$
logprob $= \log \text{softmax}(\mathbf{o})$
function $\texttt{TransformerLayer}(\mathbf{y}, \mathbf{c})$
    $\mathbf{y} = \texttt{CausalMHSA}(\texttt{LN}(\mathbf{y})) + \mathbf{y}$
    $\mathbf{y} = \texttt{DO}(\texttt{EncoderAtt}(q = \texttt{LN}(\mathbf{y}), k = \mathbf{c}, v = \mathbf{c})) + \mathbf{y}$
    $\mathbf{y} = \texttt{DO}(\texttt{FC2}(\texttt{DO}(\texttt{ACTFn}(\texttt{FC1}(\texttt{LN}(\mathbf{y})))))) + \mathbf{y}$
end function

textual embedding sequence $\mathbf{c}$. The encoder has the same architecture as `XLS-R`. The previous chapter gives the encoder's computations in Algorithm 3. The parameters of the speech encoder are initialized using expanded `SAMU-XLS-R` (Section 6.2).

The transformer decoder performs the computations described in Algorithm 4. Decoder comprises several transformer layers `TransformerLayer`. Transformer layers comprise of Causal Multi-Headed Self-Attention (`CausalSelfAtt`), Encoder Cross Attention (`EncoderAtt`) that uses the output of `CausalSelfAtt` as query (q), the encoder output $\mathbf{c}$ as both key (k), and value (v) to compute the output representation. Transformer layer also comprises of two fully-connected layers `FC1` and `FC2`, Dropout (`DO`), Activation Function (`ACTFn`), and Layer Normalization (`LN`). The decoder outputs a probability distribution over the text translation. It is conditioned on the encoder's output $\mathbf{c}$ and the ground-truth text translation corresponding to the speech utterance. During inference, the decoder cannot access ground-truth translation and generates it using *beam search*.

6.3.2 Learning

Given a speech-text translation pair $(\mathbf{a}_{1:S}, \mathbf{y}_{1:L})$, we tune the parameters of the translation model to maximize the likelihood function $p(\mathbf{y}_{1:L}|\mathbf{a}_{1:S})$, output by the model's decoder as explained above. Also, we employ *label-smoothing* (Szegedy et al., 2016a; Müller, Kornblith, and G. Hinton, 2019) to compute a label-smoothed likelihood, where a ground-truth output token $\mathbf{y}_l$ is randomly replaced with the label predicted by the model $\hat{\mathbf{y}}_l$. We set the probability of token replacement as 0.1.

Optimizer Settings. We use the Adam optimizer (Kingma and J. Ba, 2014) with a learning rate of 5e-4. We use a three-phase learning rate scheduler as in Baevski et al. (2020): (i) Warm-Up the learning rate to 5e-4 for the first 10% of the training iterations, (ii) Keep the learning rate constant for the next 40% of iterations, and (iii) Decay the learning rate linearly for the rest of training. We use 28K training iterations and train the model on 8 A100 (80 GB) NVIDIA GPUs. The training can also be carried out on V100 (32 GB) GPUs. A single training iteration uses a batch size of 10 minutes of speech utterances paired with their text translations. We use *mixed-precision* training style (Micikevicius et al., 2017); most computations are performed on half-precision floating point numbers except the final loss computation. We use the *fairseq* toolkit (Ott et al., 2019) for model translation model development. All the hyperparameters (E.g., learning rate) are manually chosen according to the translation model's performance on a small development set.

Masking Parameters. The speech encoder comprising the translation model consists of a Convolutional Network (`Conv`), followed by a multi-layered transformer encoder (See Algorithm 3). Following Baevski et al. (2020), we mask (time and feature dimension) the input of the transformer encoder. The masking process is performed in the following two steps: (i) with some probability, referred to as the *masking probability*, choose a masking index, and (ii) mask M consecutive indices starting from the chosen index. M is known as the *masking span*. For the time dimension, we set 0.3 as the masking probability and the mask span of six. We set the masking probability for

the feature dimension to 0.5 and 64 as the mask span. The time and feature masking parameters are chosen according to a development set. The above-mentioned data augmentation is akin to SpecAugment (Park, Chan, et al., 2019) (Section 2.2.2). The optimal masking parameters mentioned above are chosen according to the translation model's performance on a small development set.

Encoder and Decoder Fine-Tuning. The translation model comprises 700 million trainable parameters (300M encoder and 400M decoder parameters). We fine-tune only 75 million parameters for speech-to-text translation. Most encoder parameters are fixed to the pre-trained `SAMU-XLS-R` parameters, and the decoder parameters are fixed to the pre-trained `MBART` decoder. Below we give details about encoder and decoder fine-tuning.

Encoder Fine-Tuning. We perform adapter-based fine-tuning (Section 2.2.1) of the translation model's encoder, introduced in Houlsby et al. (2019). We insert adapter layers in each transformer layer of the speech encoder. An adapter layer is a Feed-Forward neural network with one hidden layer. The adapter layer's input and output layer sizes are the same size. The hidden layer is a fraction of the input layer size. So, the adapter layer has a bottleneck architecture. We set the downsampling factor of the hidden layer to four, i.e., the size of the hidden layer is $1/4$ the size of the input layer. During translation model training, we only fine-tune the Adapter layers. Algorithm 5 shows the computations performed by the encoder and the placement of Adapter layers. The primary motivation for using adapters is to avoid *forgetting* of semantic knowledge acquired by `SAMU-XLS-R` during its multimodal pre-training phase. We show (Section 6.4.4) that preserving semantic knowledge is essential to achieve good cross-lingual task transfer from high to low-resource translation tasks.

Decoder Fine-Tuning. Like the encoder, we keep most parameters of the translation model's decoder fixed to the values of the pre-trained `MBART` transformer decoder. We fine-tune only the encoder cross-attention `EncoderAtt` and layer normalization `LN` in the decoder. We fine-tune `EncoderAtt` because, previously, it is trained as part

Algorithm 5 Computations performed by the encoder of our translation model. We insert two Adapter layers (`Adapter`) in each transformer layer (`TransformerLayer`) of the encoder. We only Fine-Tune the `Adapter` layer during translation model training. The other layers are frozen to the pre-trained `SAMU-XLS-R`. The frozen layers are represented in gray.

Input: Raw speech waveform $\mathbf{a}$
Output: Contextual speech embedding sequence $\mathbf{c}$
$\mathbf{h} = \texttt{LN}(\texttt{Conv}(\mathbf{a}))$
$\mathbf{h} = \texttt{Mask}(\mathbf{x})$
$\mathbf{h} = \mathbf{h} + \texttt{PosConv}(\mathbf{h})$
for $i = 1$ to $L - 1$ **do**
    $\mathbf{h} = \texttt{TransformerLayer}^{(i)}(\mathbf{h})$
end for
$\mathbf{c} = \texttt{LN}(\texttt{TransformerLayer}^{(L)}(\mathbf{h}))$
function `TransformerLayer`($\mathbf{x}$)
    $\mathbf{x} = \texttt{Adapter}(\texttt{DO1}(\texttt{MHSA}(\texttt{LN1}(\mathbf{x})))) + \mathbf{x}$
    $\mathbf{x} = \texttt{Adapter}(\texttt{DO3}(\texttt{FC2}(\texttt{DO2}(\texttt{ACTFn}(\texttt{FC1}(\texttt{LN2}(\mathbf{x}))))))) + \mathbf{x}$
end function

of a decoder in a text-to-text translation pipeline. Hence, the `EncoderAtt` module needs to be fine-tuned again for the downstream multilingual speech-to-text translation task to make it amenable to the input from the speech encoder. Moreover, we fine-tune `LN` because it is task and dataset-specific and empirically works well, as shown in X. Li et al. (2020a).

6.3.3 Inference

We use *beam search*, with a beam size of 5, to generate text translations for a given speech utterance. The inference process is *offline*, i.e., the decoder takes the input speech utterance into account to generate the output translation. The decoder generates translation in an autoregressive manner. We do not use any *external language model* during inference.

6.4 Evaluation

6.4.1 Translation Scenarios

We tackle the following translation scenarios in this work.

Multilingual Translation. We simultaneously train a translation model on several speech-to-text translation tasks in this scenario. E.g., we train a single model for all 21 X→EN translation tasks in the CoVoST-2 benchmark. Most of the training data come from a few high-resource translation tasks such as FR→EN and DE→EN, while most tasks such as ID→EN are low-resource. In this scenario, we compare different translation models to test for cross-lingual translation task transfer from high to low-resource translation tasks.

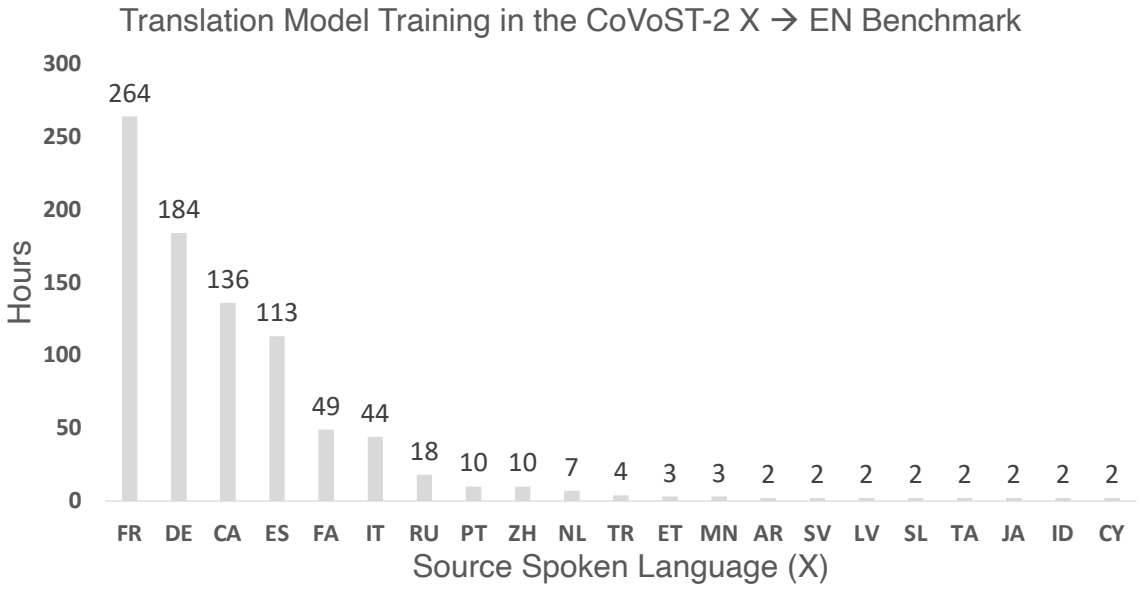
Zero-Shot Multilingual Translation. Given a set of translation tasks, we train a translation model on a subset of the tasks while keeping the rest hidden during training in this translation scenario. E.g., we train an X→EN speech-to-text translation model using high-resource translation tasks in the CoVoST-2 benchmark while keeping the mid and low-resource tasks unseen during training. We compare translation models for zero-shot cross-lingual task transfer in this scenario from high to mid and low-resource X→EN translation tasks.

6.4.2 Translation Tasks

We build translation models to tackle the following translation tasks in this work.

X→EN Speech-to-Text Translation. We perform of X→EN speech-to-text translation in this work. We build translation models for the 21 X(Speech)→EN(Text) translation tasks in the CoVoST-2 translation benchmark (Changhan Wang, Pino, et al., 2020). The 21 spoken languages in CoVoST-2 are fr, de, es, ca, it, ru, zh, pt, fa, et, mn, nl, tr, ar, sv, lv, sl, ta, ja, id, and cy. The 21 translation tasks are divided into high, mid, and low groups, depending on the amount of labeled data available for

Figure 6-2: Number of hours of labeled training data (Y-Axis) for all the 21 X→EN translation tasks in the CoVoST-2 benchmark.



a translation task. High-resource tasks have more than 100 hours of labeled training data, mid-resource tasks have between 10 and 100 hours, and low-resource have less than ten hours of labeled training data. CoVoST-2 has four high-resource translation tasks corresponding to fr, de, es, and ca source languages and five mid-resource tasks corresponding to it, ru, zh, pt, and fa languages. The rest of the tasks are low-resource. Fig. 6-2 presents the training data for each of the 21 translation tasks. Notice the data imbalance among different tasks. See Section 2.4.2 for detailed CoVoST-2 data statistics, such as the number of hours of training data for each translation task, the average duration of a speech utterance, etc.

In the multilingual translation scenario, we simultaneously train translation models on all 21 tasks mentioned above. We only train translation models on the four high-resource tasks in the zero-shot scenario.

X→Y Speech-to-Text Translation. We develop translation models for the 72 X→Y translation tasks in the Europarl benchmark (Iranzo-Sánchez et al., 2019). There are nine spoken languages in Europarl namely, en, fr, de, it, es, pt, pl, ro, and nl. Speech utterances in each spoken language are paired with their corresponding

Table 6.2: Training data (hours) for the 72 translation tasks X→Y in the Europarl Speech-to-Text translation benchmark.

SRC/TGT	FR	DE	IT	ES	PT	PL	RO	NL	EN
FR	-	21	20	21	22	20	18	22	32
DE	18	-	17	18	18	17	17	18	30
IT	21	21	-	21	21	21	19	20	37
ES	14	14	14	-	14	13	12	13	22
PT	10	10	10	10	-	9	9	9	15
PL	18	18	17	18	18	-	16	18	28
RO	12	12	12	12	12	12	-	12	24
NL	5	5	4	5	4	4	4	-	7
EN	81	83	80	81	81	79	72	80	-

text translations in eight other languages. In the zero-shot scenario, we train the translation models on 32 tasks corresponding to the following four source languages en, fr, de, and it. Each of the four source languages is paired with eight target languages. Table 6.2 shows each translation task's labeled training data. Also, See Section 2.4.2 for detailed corpus statistics.

6.4.3 Baseline/Topline Translation Models

We compare our transformer model (SAMU-XLS-R-300M) for translation against several other translation models listed below.

SAMU-XLS-R-300M. We propose SAMU-XLS-R-300M transformer model for translation in this work. The encoder is initialized using the pre-trained SAMU-XLS-R speech encoder (Section 6.2), and the decoder is initialized using the pre-trained MBART text decoder. The suffix 300M in SAMU-XLS-R-300M refers to the model's size of 300M parameters.

XLS-R(-300M, 1B, 2B). We compare SAMU-XLS-R-300M with three XLS-R speech encoder based translation models namely, XLS-R-300M, XLS-R-1B, and XLS-R-2B. The three translation models differ from SAMU-XLS-R-300M model in that the encoder of

the translation model is initialized using pre-trained XLS-R speech encoders of different sizes ranging from 300M to 2B parameters. The decoder is initialized using the pre-trained MBART decoder. Unlike our multimodal SAMU-XLS-R speech encoder, XLS-R is only trained using unlabeled speech data. Also, SAMU-XLS-R is specifically trained to learn semantic representations, while XLS-R has no constraints imposed during its training phase to encode semantic knowledge.

mSLAM. We compare SAMU-XLS-R-300M translation model against two mSLAM (Bapna, Cherry, et al., 2022b) speech encoder based translation models namely, mSLAM-600M, and mSLAM-2B. Like SAMU-XLS-R, mSLAM speech encoder is a multimodal speech-text encoder. Unlike SAMU-XLS-R, which is trained using semantic supervision from a pre-trained semantic text encoder, mSLAM is not trained with explicit semantic supervision.

Cascaded Translation. We compare SAMU-XLS-R-300M with a strong *cascaded* translation system. We perform the translation in two steps: (i) Transcribe the speech utterance using an ASR model, and (ii) Use a text-to-text translation model to translate the ASR transcript to text in a target language. We use whisper-large-v2[1] (Radford, J. W. Kim, et al., n.d.) as the ASR model in the first step and MBART (Y. Liu et al., 2020b) text-to-text translation model for the second step in the cascade. For X→EN cascade, we use MBART-many-to-English text-to-text translation model[2]. The Whisper ASR model is multilingual that supports transcription of around 93 languages. Since Radford, J. W. Kim, et al. (n.d.) shows that whisper achieves state-of-the-art ASR performance on several public benchmarks, we choose Whisper for automatically transcribing speech. MBART-many-to-English translation model can translate text from 50 languages to English.

Transcripts. As a topline, we use the ground-truth text transcripts corresponding to speech utterances and use MBART-many-to-English to translate to English.

[1] https://huggingface.co/openai/whisper-large-v2
[2] https://huggingface.co/facebook/mbart-large-50-many-to-one-mmt

6.4.4 Results

Table 6.3: We compare our proposed SAMU-XLS-R-300M translation model with several other translation models, whose encoders are initialized using differently sized pre-trained XLS-R multilingual unimodal speech encoders. The performance is measured using BLEU-4, Google-BLEU, ROUGE-L, METEOR, BERTScore, and NIST translation metrics.

	BLEU-4				Google-BLEU			
Model	High	Mid	Low	TRFGap	High	Mid	Low	TRFGap
XLS-R-300M	30.6	18.9	5.1	25.1	0.36	0.24	0.10	0.26
XLS-R-1B	34.3	25.5	11.7	22.6	0.38	0.29	0.16	0.22
XLS-R-2B	36.1	27.7	15.1	21.0	0.39	0.31	0.20	0.19
SAMU-XLS-R-300M	34.4	31.1	20.3	14.1	0.38	0.34	0.24	0.13
Cascaded	32.6	29.7	22.5	10.1	0.36	0.33	0.26	0.10
Transcripts	36.4	34.2	27.9	8.5	0.39	0.37	0.32	0.08
	ROUGE-L				METEOR			
Model	High	Mid	Low	TRFGap	High	Mid	Low	TRFGap
XLS-R-300M	0.60	0.44	0.23	0.37	0.62	0.45	0.24	0.38
XLS-R-1B	0.61	0.49	0.31	0.30	0.63	0.50	0.32	0.31
XLS-R-2B	0.63	0.53	0.38	0.25	0.65	0.53	0.39	0.26
SAMU-XLS-R-300M	0.62	0.58	0.42	0.19	0.64	0.59	0.45	0.19
Cascaded	0.60	0.56	0.45	0.14	0.61	0.56	0.46	0.15
Transcripts	0.64	0.61	0.52	0.11	0.66	0.62	0.54	0.12
	BERTScore				NIST			
Model	High	Mid	Low	TRFGap	High	Mid	Low	TRFGap
XLS-R-300M	0.56	0.33	0.04	0.52	8.0	5.0	1.9	6.1
XLS-R-1B	0.58	0.41	0.16	0.42	8.3	5.9	3.0	5.3
XLS-R-2B	0.61	0.45	0.25	0.36	8.6	6.3	3.7	4.9
SAMU-XLS-R-300M	0.59	0.54	0.34	0.25	8.3	7.1	4.4	4.0
Cascaded	0.56	0.50	0.37	0.20	8.2	6.9	4.9	3.3
Transcripts	0.62	0.58	0.47	0.14	8.7	7.7	5.7	3.0

Multilingual X→EN Speech-to-Text Translation. **Table 6.3** shows the performance of different translation models on the high, mid, and low-resource translation groups in the CoVoST-2 speech-to-text translation benchmark. We compare our proposed SAMU-XLS-R-300M translation model against XLS-R-300M, XLS-R-1B, and XLS-R-2B translation models. CoVoST-2 comprises 21 X→EN translation tasks, and

the translation models are trained simultaneously on all translation tasks. See Section 6.4.2 for details about the translation tasks, and Section 6.4.3 for details about the different translation models.

Table 6.4: We compare our proposed SAMU-XLS-R-300M translation model with mSLAM translation models, whose encoders are initialized using different sized pre-trained mSLAM multilingual multimodal speech encoders. The performance is measured using BLEU-4 translation metric.

	BLEU-4			
Model	High	Mid	Low	TRFGap
mSLAM-600M	37.6	27.8	15.1	22.5
mSLAM-2B	37.8	29.6	18.5	19.3
SAMU-XLS-R-300M	34.4	31.1	20.3	14.1

The model performance is measured using the standard translation metrics, namely BLEU-4 (Post, 2018), Google-BLEU (Yonghui Wu, Schuster, Zhifeng Chen, Le, Norouzi, W. Macherey, Krikun, Cao, Gao, K. Macherey, Klingner, Shah, Johnson, X. Liu, Kaiser, et al., 2016), ROUGE-L (Lin, 2004), METEOR (Banerjee and Lavie, 2005), BERTScore (Eddine et al., 2021), and NIST (Doddington, 2002). We make the following **observations**: (i) On **High resource tasks**, the XLS-R-2B model performs the best, with SAMU-XLS-R-300M lagging a couple of points behind. Compared to the similar-sized XLS-R-300M model, SAMU-XLS-R-300M performs 4 BLEU points better. (ii) On **Mid resource tasks**, SAMU-XLS-R-300M outperforms all the models achieving a BLEU score of 31.1, which is significantly better than XLS-R-300M model's BLEU score of 5.1. SAMU-XLS-R-300M even outperforms the much larger XLS-R-2B speech encoder by 3.3 BLEU points. (iii) On **Low resource tasks**, SAMU-XLS-R-300M performs the best. Compared to the similar-sized XLS-R-300M model, SAMU-XLS-R-300M does better by 15 BLEU points. It also outperforms the much larger XLS-R-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 SAMU-XLS-R-300M model compared to other models. Second, to SAMU-XLS-R-300M is XLS-R-2B, which has a TRFGap of 21 BLEU points while having 500% more parameters.

Table 6.4 compares SAMU-XLS-R-300M translation model with mSLAM-600M, and mSLAM-2B models that use differently sized pre-trained multimodal (speech-text) multilingual mSLAM speech encoder. SAMU-XLS-R-300M performs better on mid- and low-resource translation tasks. Importantly, SAMU-XLS-R-300M 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-600M, and 19.3 for mSLAM-2B. The BLEU scores for mSLAM models are lifted from Bapna, Cherry, et al. (2022b). Since Bapna, Cherry, et al. (2022b) report only BLEU scores on the CoVoST-2 benchmark, and we do not have access to mSLAM models, we could not evaluate the model using metrics other than BLEU-4.

The above observations validate our claims that building translation technology on top of semantic speech representations would increase the cross-lingual task knowledge transfer from high to low-resource languages. Similar inferences can be reached by using metrics other than BLEU-4.

Language-wise Performance Breakdown. Table 6.5 shows the performance of different translation models on each of the 21 X→EN speech-to-text translation tasks in the CoVoST-2 benchmark. We observe that SAMU-XLS-R-300M significantly outperforms the similarly sized XLS-R-300M translation model on several mid and low-resource languages. Some notable improvements are for the following source languages: id (34.4 vs. 1.2 BLEU), cy (34.1 vs. 2.8 BLEU), ja (13.1 vs 0.6 BLEU), sv (28.5 vs 11.3 BLEU), tr (28.4 vs 6.7 BLEU), ar (36.6 vs. 3.9 BLEU), fa (22.0 vs. 6.3 BLEU), and pt (44.2 vs. 30.8 BLEU). For some source languages SAMU-XLS-R-300M does not do well, such as lv (1.9 BLEU), sl (12.9 BLEU), mn (3.4 BLEU), et (11.6 BLEU), ta (4.0), ja (13.1 BLEU), and zh (13.1 BLEU). For ta, ja, mn, and zh, the topline (Transcripts) model also performs poorly. For sl, and lv, the topline text-to-text translation and cascaded models perform significantly better than the SAMU-XLS-R-300M model.

Poor performance of SAMU-XLS-R-300M on lv, mn, and sl can be explained by the lack of available transcribed speech data in these languages for multimodal pre-

Table 6.5: We compare our proposed SAMU-XLS-R-300M translation model with several other translation models, whose encoders are initialized using differently sized pre-trained XLS-R multilingual unimodal speech encoders. The performance is measured using the BLEU-4 translation metric.

Model	fr	de	es	ca	it	fa	ru	zh	pt
XLS-R-300M	34.9	29.3	35.9	30.6	30.9	6.3	30.0	5.2	30.8
XLS-R-1B	36.3	31.4	37.8	32.1	33.5	9.1	35.7	6.9	41.4
XLS-R-2B	37.6	33.6	39.1	33.9	35.0	13.0	39.5	9.4	41.8
SAMU-XLS-R-300M	36.1	31.7	37.9	31.9	34.0	22.0	42.1	13.1	44.2
Cascaded	32.7	30.9	36.4	30.3	32.5	14.3	42.9	14.4	44.2
Transcripts	38.4	34.1	41.1	32.1	36.4	24.2	46.0	17.4	46.9

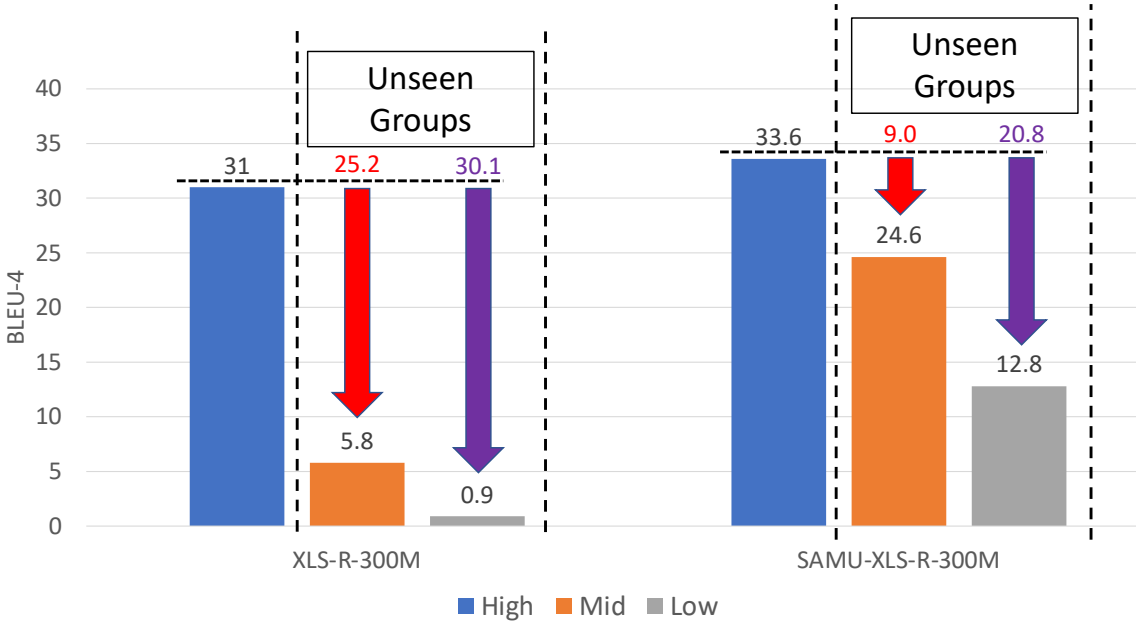
Model	nl	tr	et	mn	ar	sv	lv	sl	ta
XLS-R-300M	25.1	6.7	4.0	0.2	3.9	11.3	7.4	8.3	0.0
XLS-R-1B	29.6	11.1	8.0	0.6	9.3	24.5	15.7	16.8	0.1
XLS-R-2B	31.6	16.9	11.2	1.5	17.1	29.7	19.7	19.0	0.5
SAMU-XLS-R-300M	34.9	28.4	11.6	3.4	36.6	28.5	1.9	12.9	4.0
Cascaded	33.0	25.1	17.4	0.0	35.7	39.6	20.3	27.8	1.6
Transcripts	36.3	28.8	25.6	3.8	44.6	46.4	29.2	38.4	2.2

Model	cy	ja	id
XLS-R-300M	2.8	0.6	1.2
XLS-R-1B	6.61	1.3	7.8
XLS-R-2B	14.2	3.5	16.4
SAMU-XLS-R-300M	34.1	13.1	34.4
Cascaded	6.4	19.4	43.6
Transcripts	9.0	20.8	49.8

training of SAMU-XLS-R (Section 6.2). We have 7 hours for lv, 12 hours for mn, and 9 hours of transcribed speech for sl, compared to 317 hours for fa, 85 hours for ar, 116 hours for cy, 98 hours for nl, 40 hours for sv, 25.8 hours for id, 162 hours for ru, 400 hours for es, and close to 1K hours for fr, de, and ca. However, for ta, zh, and ja we have a decent amount of transcribed speech, but still, the performance is relatively poor. This can be explained away by observing the performance of the topline (Transcripts), where we use a pre-trained MBART-many-to-English text-to-text translation model (Details in Section 6.4.3) that translates the ground-truth transcripts corresponding to speech utterances in language X to text in English. The topline performance for zh, ja, and ta is relatively poor. Since we use the decoder

of MBART-many-to-English model to initialize the decoder of our speech-to-text translation model SAMU-XLS-R-300M, we also observe poor speech-text-translation performance on these tasks.

Figure 6-3: We report average BLEU-4 for the zero-shot X→EN multilingual speech-to-text translation scenario on the high, mid, and low resource task groups in the CoVoST-2 benchmark. We compare our translation model SAMU-XLS-R-300M with the similarly sized XLS-R-300M translation model. The translation models are only trained on high-resource groups, while the mid and low-resource groups are *unseen* during training.



Zero-Shot X→EN Speech-to-Text Translation. Next, we train the translation models on four high-resource X→EN translation tasks in the CoVoST-2 benchmark (See Section 6.4.2 for details). We evaluate the X→EN translation models on the high, mid, and low task groups to test for zero-shot cross-lingual transfer capability of SAMU-XLS-R-300M from high to mid and low-resource X→EN tasks. We compare SAMU-XLS-R-300 with XLS-R-300M translation model. Figure 6-3 presents the results. We observe that SAMU-XLS-R-300M performs on average 18.8 BLEU points better in the mid-resource and 11.9 BLEU points in the low-resource group. The cross-lingual transfer gap between the high & mid and high & low groups is significantly smaller for SAMU-XLS-R-300M (9.0, and 20.8) than XLS-R-300M (25.2, and 30.1).

These results strengthen our claims that building speech translation technology with semantic speech representations would improve cross-lingual transfer across languages. Note that zero-shot implies that the translation model during its training does not see any paired X→EN translation data for mid and low-resource languages X. But, transcribed speech data was available for these languages during multimodal pre-training of SAMU-XLS-R speech encoder which is used to initialize the encoder of the SAMU-XLS-R-300M translation model.

Zero-Shot X→Y Speech-to-Text Translation. Finally, to further bolster our claims about the usefulness of semantic speech representations for translation, we compare translation models on the 72 X→Y translation tasks in the Europarl Benchmark. See Section 6.4.2 for details about translation tasks and available data for training translation models.

Figure 6-4: Absolute BLEU score improvements using SAMU-XLS-R-300M over XLS-R-300M baseline on the 72 X→Y translation tasks in the Europarl benchmark. The translation models are trained on a subset of 32 translation tasks, corresponding to four source languages, while 40 tasks are unseen during training corresponding to five source languages.

	SRC/TGT	FR	DE	ES	IT	PL	PT	RO	NL	EN
Training Data	FR	NA	0.8	0	-0.4	-0.1	-1.5	0.7	-0.1	1.2
	DE	2.3	NA	1.5	1.3	1	0.9	1.6	-0.1	2
	ES	0.9	0.8	NA	0.3	1.7	0.3	1.4	1	2.9
	IT	-0.8	-0.1	0.1	NA	-0.8	-1.6	0.4	-0.4	1.3
Unseen During Training	PL	19.6	13.7	19.9	14.5	NA	16.1	16.1	15.5	23.8
	PT	1.4	1.1	1.6	0.3	0.5	NA	1.9	0.4	2
	RO	18.4	11.5	17.2	13.7	11.2	14.8	NA	12.7	21.9
	NL	11.5	9.5	8.9	9	7.4	10.9	10.1	NA	13.3
	EN	17.7	13.5	19.6	14.1	9.7	16.6	17.5	13.9	NA

We train translation models SAMU-XLS-R-300M and XLS-R-300M on a 32-task subset out of 72 tasks. The 32 tasks correspond to four source spoken languages: fr, de, es, and it, while the remaining 40 tasks correspond to five source languages: pl, pt, ro, nl, and en. Speech utterances in each source language are paired with text translations in eight other languages. Unlike X→EN translation models discussed above, we initialize the decoder of X→Y translation models with the decoder of MBART-many-to-many[3]

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

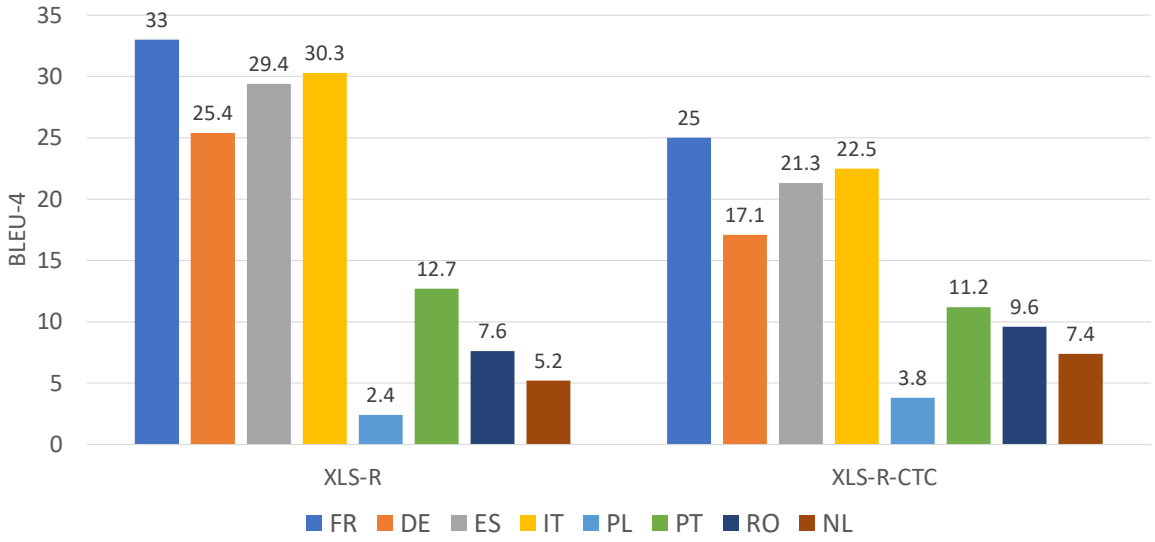
model instead of MBART-many-to-English, since we have to generate translations in multiple target languages.

Figure 6-4 compares the performance of XLS-R-300M, and SAMU-XLS-R-300M on the 72 translation tasks. We report the absolute BLEU-4 score improvement that SAMU-XLS-R-300M achieves over XLS-R-300M baseline translation model. The darker the cell in the figure, the greater the improvement in the BLEU score. We observe that SAMU-XLS-R-300M performs drastically better than XLS-R-300M on the 40 unseen (during translation model training) translation tasks and at par on the 32 seen (during translation model training) translation tasks. We observe the biggest improvements in unseen tasks such as pl→en (23.8), ro→en (21.9), pl→en (19.9), pl→fr (19.6), ro→es (17.2), etc. Overall, SAMU-XLS-R-300M improves over XLS-R-300M baseline model by an average of 12 BLEU-4 points.

6.5 Analysis

XLS-R vs. XLS-R-CTC. Figure 6-5 The experiments above compare SAMU-XLS-R with XLS-R multilingual speech encoder. XLS-R is trained using multilingual unlabeled speech data in 128 languages via self-supervised learning, while SAMU-XLS-R fine-tunes pre-trained XLS-R using multilingual transcribed speech data via semantic knowledge distillation. But, what if instead of fine-tuning pre-trained XLS-R with semantic supervision from text transcriptions like SAMU-XLS-R does, we fine-tune XLS-R to generate the transcriptions automatically? We compare the translation performance of XLS-R and XLS-R-CTC encoders in Fig. 6-5. XLS-R-CTC refers to the encoder we get after supervised CTC-based fine-tuning of pre-trained XLS-R using multilingual transcribed speech data. CTC framework (Graves, Fernández, et al., 2006) for training end-to-end ASR models is briefly described in Section 2.3. For CTC fine-tuning of XLS-R, we choose the same multilingual transcribed speech data used for multimodal pre-training of our semantic speech encoder SAMU-XLS-R. We observe that the pre-trained XLS-R encoder performs better than XLS-R-CTC encoder. XLS-R-CTC encoder is slightly better on some translation tasks. Still, the difference is so small that our previous

Figure 6-5: We compare on the Europarl X→EN benchmark XLS-R and XLS-R-CTC initialization of the translation model's encoder. XLS-R is pre-trained using unlabeled speech via self-supervised learning, while XLS-R-CTC refers to the XLS-R encoder that is fine-tuned (after self-supervised pre-training) using transcribed speech data. We report the BLEU-4 score for eight source spoken languages. Speech utterances in each source language are paired with its English text translations.



observations on the efficacy of our proposed semantic speech encoder SAMU-XLS-R are not impacted, even though we compared SAMU-XLS-R with the XLS-R encoder, which is trained using unlabeled multilingual speech.

Table 6.6: We compare the translation model's performance when using Adapter-based fine-tuning vs. fine-tuning all the encoder parameters. The performance is measured using the **BLEU-4** translation metric.

Model	fr	de	es	it	pl	pt	ro	nl
XLS-R-300M-Full	33.0	25.4	29.4	30.3	2.4	12.7	7.6	5.2
XLS-R-300M-Adapter	29.8	22.4	25.5	27.2	3.2	15.6	9.1	8.2
SAMU-XLS-R-300M-Full	33.2	26.3	29.7	30.7	3.0	12.2	9.8	7.0
SAMU-XLS-R-300M-Adapter	32.3	25.5	29.3	28.7	24.3	18.4	29.9	24.1

Adapter vs. Full Encoder Fine-Tuning. As mentioned in Section 6.3.2, we perform Adapter-based fine-tuning of the translation model's encoder, where we insert new task-specific parameters in the form of adapters in each encoder layer. We only

fine-tune adapter layers while keeping the rest of the layers frozen to their pre-trained values. Table 6.6 compares the translation model's performance when using Adapter-based fine-tuning vs. fine-tuning all the encoder parameters. We observe that using adapter fine-tuning with XLS-R-300M translation model brings performance gains on low-resource tasks such as ro→en, nl→en, pt→en, and pl→en. In contrast, the performance degrades significantly for higher-resource tasks compared to full encoder fine-tuning. We observe a similar trend with SAMU-XLS-R-300M translation model. But, the performance gains for low-resource tasks are drastic with adapter-based fine-tuning of the encoder. Again, this is due to our proposed *semantic* speech encoder SAMU-XLS-R, which results in a significant cross-lingual transfer from high to low-resource translation tasks. This result also shows that preserving semantic knowledge during training for downstream translation tasks, learned by the SAMU-XLS-R encoder as a result of our multimodal learning framework (Section 5.2) is essential.

Translation Model Predictions Tables 6.7, 6.8, 6.9, and 6.10 present outputs from SAMU-XLS-R-300M, XLS-R-300M, and XLS-R-1B translation models on a few randomly selected speech utterances from the CoVoST-2 X→EN low-resource translation groups. We couldn't draw any exciting conclusions from visually comparing the translations generated by different models.

6.6 Chapter Summary

This chapter 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 standard encoder-decoder model. We analyze the impact of different encoder initializations on the downstream translation task performance. We show that by initializing the encoder with an encoder that we pre-train using the newly introduced *semantic knowledge distillation framework* SAMU-XLS-R (Chapter 5), we achieve significantly better cross-lingual transfer in the downstream speech-to-text translation task than the baselines. The baseline trans-

lation models use the state-of-the-art multilingual pre-trained speech encoder `XLS-R`, and others for initialization.

To substantiate our claims, we perform multilingual translation on two public benchmarks, CoVoST-2 and Europarl. On the 21 X→English CoVoST-2 speech translation tasks, we achieve an average improvement of 12.8 BLEU points. In the zero-shot scenario, where we train the translation model only on the four high-resource languages while keeping the rest 17 languages unseen (during training), we achieve an average improvement of 11.8 BLEU points over the baseline `XLS-R` encoder initialization. In particular, we achieve drastic improvements of 18.8 and 11.9 average BLEU points on medium and low-resource languages, respectively. We made similar observations on the Europarl X→Y speech-to-text translation benchmark.

Our work has limitations. Currently, training `SAMU-XLS-R` requires access to multilingual transcribed data, which could be hard to get for many spoken languages. Also, the dependence on a pre-trained text encoder hinders expanding `SAMU-XLS-R` to more languages. Hence, future work should focus on injecting semantic information via *weakly supervised learning* using unaligned speech and text data and without using the `LaBSE` text encoder.

Table 6.7: We subset the test split of the CoVoST-2 X→EN low-resource corpus. The subset consists of speech utterances for which SAMU-XLS-R-300M translation model achieves more than 0.8 Google-BLEU. We present the reference (ground-truth) English translation and predicted translations with SAMU-XLS-R-300M, XLS-R-300M, and XLS-R-1B translation models. X refers to the language of the speech utterance.

X	Reference	SAMU-XLS-R-300M	XLS-R-300M	XLS-R-1B
sv	You can come back and get more.	You can come back and get more.	You can come back and do more.	You can come back and get more.
ar	I like drinking hot coffee.	I like drinking hot coffee.	I like to play golf.	I like ice cream very much.
ar	It is better to go now before you miss the bus.	It's better to go now before you miss the bus.	One day, I found out that the house was empty.	As long as you don't do it, you won't be able to do it.
nl	Is the L of Land Rover, Lexus or Lotus.	It is the L of Land Rover, Lexus or Lotus.	It's the L of land Rovers, Lexus or Lotus.	It is the L of Land Rover, Lexus or Lotus.
tr	Approximately a hundred and thirty experts will participate in the meetings.	Around one hundred and thirty experts will participate in the meetings.	The competitions will last for around thirty-eight hours.	Approximately thirty-three people will participate in the events.
id	Tom died of rabies after being bitten by a bat.	Tom died of rabies after being bitten by a larva.	Tom said that he was going to buy a new car.	Tom listened to the tape recorder while he was listening to the music.
cy	Do you need to sit down?	Do you need to sit down?	I want to go to school.	Do you want to eat?
sv	I just want to end it.	I just want to end it.	I'll give it to you all the time.	I don't want to see you anymore.
ja	Tanaka can play tennis.	Tanaka can play tennis.	Tanaka is a tennis player.	Tanaka's tennis is good.
tr	The final decision will be taken at the summit.	The final decision will be taken at the summit.	The book will not be published yet.	The final decision will be made in Serbia.

Table 6.8: We subset the test split of the CoVoST-2 X→EN low-resource goup. The subset consists of speech utterances for which SAMU-XLS-R-300M translation model achieves between 0.6 to 0.8 Google-BLEU. We present the reference (ground-truth) English translation and predicted translations with SAMU-XLS-R-300M, XLS-R-300M, and XLS-R-1B translation models. X refers to the language of the speech utterance.

X	Reference	SAMU-XLS-R-300M	XLS-R-300M	XLS-R-1B
nl	The exam consisted of fifty multiple choice questions.	The exam consisted of fifty more choice questions.	The exam consisted of fifty more questions.	The exam consisted of fifty more questions.
tr	The project is expected to be completed in three days.	The project is expected to be completed within three years.	The project is expected to be completed in three years.	The project is expected to be completed in three years.
ar	Her new novel will be published next month.	Her novels will be published next month.	I'm going to play the violin next week.	I'm going to buy a new pair of shoes.
cy	Morgan and Sara playing hide and seek.	Morgan and Sara are playing hide and seek.	Morgan's sister is dying. chess.	Morgan and Sara are playing
et	I am sure there will be new elections, new opportunities.	There will be new elections, new opportunities.	Then there are new developments and new opportunities.	Only new elections and new opportunities remain.
sv	I suggest we have a naughty dinner together.	I suggest we have a nice dinner together.	I suggest that we have a breakfast at midday.	I believe that we have a bad meeting again.
id	I brush my teeth after eating rice.	I brushed my teeth after eating rice.	I've never been to Japan before.	I used to play guitar until I was a kid.
ar	Return the book where you found it.	Return the book where I found it.	There is a book in the library.	This book is just for you.
lv	Awesome, isn't it?	Superb, isn't it?	Can you help me?	Can you help me?
sv	Our plan is to change the future.	Our plan is to change this future.	Your plan is to change this country.	Our plan is to change this future.

Table 6.9: We subset the test split of the CoVoST-2 X→EN low-resource goup. The subset consists of speech utterances for which SAMU-XLS-R-300M translation model achieves between 0.4 to 0.6 Google-BLEU. We present the reference (ground-truth) English translation and predicted translations with SAMU-XLS-R-300M, XLS-R-300M, and XLS-R-1B translation models. X refers to the language of the speech utterance.

X	Reference	SAMU-XLS-R-300M	XLS-R-300M	XLS-R-1B
sv	It's enough for a nice vacation.	That is enough for a nice semester.	It's about the end of the semester.	There comes a final semester.
tr	However, the government hasn't paid this amount.	However, the government has not paid this figure.	However, the government does not agree with this opinion.	However, the government does not accept this figure.
nl	They climbed the Mont Blanc together.	They climbed the Mont-Blanc together.	They gathered the Montblanc together.	They signed the Montblain together.
cy	The truth is, he couldn't get to Wales from somewhere like London before the afternoon post.	The truth is that we could not get to Wales from somewhere like London before post-dinner.	It is said that there will be no change to the old Linden street that will be in the post office next year.	From time to time, you can't see Cameroon from the streets of London until the end of the day.
tr	This is the first visit of Amon to the country.	Visiting Ardon was the first visit to the country.	The journey took eight days.	The visit was attended by both countries.
id	Being burnt with love.	Turned up with love.	For Macarthy, go to Quinta.	To make a fool of myself.
sv	You were in the meeting.	You were at the meeting.	You were on time.	You were at the table.

Table 6.10: We subset the test split of the CoVoST-2 X→EN low-resource goup. The subset consists of speech utterances for which SAMU-XLS-R-300M translation model achieves between 0.0 to 0.4 Google-BLEU. We present the reference (ground-truth) English translation and predicted translations with SAMU-XLS-R-300M, XLS-R-300M, and XLS-R-1B translation models. X refers to the language of the speech utterance.

X	Reference	SAMU-XLS-R-300M	XLS-R-300M	XLS-R-1B
tr	This year, the partner country of the event was Turkey.	Turkey was the joint country of the event this year.	Turkey was the eighth country in this event.	Turkey was the first country in this year's event.
et	There was nowhere to rise nor fall before the point giving Power Stage, though there was five extra points available - two more than in previous seasons.	There were no more points to be won than at the start of the Powell Stage, but there were up to five extra points, two more than in the previous two seasons.	The point of the boat was to get get out of the Poulstead and get out of the Stead and get out of the Stead and get out of the Stead	When the battles started, the Paul-Steig-Hae was no longer able to win any of the points, but he was able to win just five points, two more than in the previous years.
ja	If you eat lots of rice, your body will grow strong.	If you eat a lot of food, you will become a healthy body.	There is a lot of work to be done at the station.	There's a lot of things to do in the garden.
mn	This is second card, he said.	This is the second circle.	It's like that.	That's why I'm so happy.
nl	I scooped up all the dirt from the water with a small fishing net.	With a net I pulled all the garbage out of the water.	With a glass of paper, I put all the felt out of the water	I used a net to fish all the fish out of the water.

Chapter 7

Conclusions

This thesis addresses two transfer learning problems; unsupervised domain adaptation for End-to-End Automatic Speech Recognition (ASR) and cross-lingual transfer learning in Automatic Speech-to-Text translation.

7.1 Unsupervised Domain Adaptation

7.1.1 Summary

The performance of an ASR model degrades significantly when the training data distribution (source domain) does not match the data distribution the model encounters during deployment (target domain). A straightforward remedy is to collect labeled examples in the target domain to re-train the ASR model. But, collecting labeled examples is expensive and time-consuming, while unlabeled target domain data is often readily available. Therefore, we propose an unsupervised domain adaptation method for source-to-target domain adaptation.

We focus on self-training, a classic algorithm for semi-supervised learning, which has recently shown excellent results on neural sequence generation tasks such as speech recognition, translation, and text-based machine translation. Self-Training (ST) is a teacher/student learning framework that trains an initial model on labeled examples and improves it iteratively using unlabeled data points.

An iteration of self-training consists of three steps: a) A teacher model is trained on labeled examples using supervised learning. b) Teacher is used to generate predictions for unlabeled data points. These predictions are called pseudo-labels. c) A student model is trained on combined labeled and pseudo-labeled examples. Pseudo-labeled examples are a data augmentation method that improves the student model's generalization performance. But, the performance of self-training depends on the quality of pseudo-labels. Pseudo-labels can be erroneous, which could lead to sub-optimal student training. The noisy pseudo-label problem is amplified in our scenario where a feature distribution mismatch exists between the labeled source and the unlabeled target.

In Chapter 3, we propose Dropout Uncertainty-Driven Self-Training (DUST) by augmenting the classic ST algorithm with a pseudo-label filtering method to alleviate this issue. Our pseudo-label filtering method involves sampling multiple predictions from the teacher model for an unlabeled data point and computing agreement among the sampled predictions. We weed out the unlabeled point and its corresponding pseudo-label if the agreement is low. We generate different samples by injecting dropout noise in the model during inference.

We show the effectiveness of DUST in several domain adaptation scenarios; Read Speech source to Oratory and Conversational speech target domain adaptation. We make the following **key observations**. DUST is better than ST in severe source-target domain mismatch. DUST is more compute and data-efficient than ST in moderate domain mismatch scenarios. DUST combined with a Self-Supervised Pre-Trained speech encoder can be quite effective in the low-resource ASR scenario.

We show an interesting application of DUST in **Chapter 4** for few-shot learning of ASR in a target language. Currently, a two-step sequential transfer learning formula is popular for few-shot learning: First, pre-train a large neural network speech encoder (usually a transformer encoder) via Self-Supervised Learning (SSL) (E.g., Wav2Vec-2.0) on massive amounts of unlabeled data in the target language. Second, fine-tune the pre-trained encoder on the downstream ASR task using a few labeled examples. The two-step formula fails when the massive unlabeled data assumption does not hold,

which is true for many low-resource languages. We address this issue by proposing a three-step sequential transfer learning formula.

First, we pre-train a transformer encoder on some high-resource source language (E.g., English) via SSL. Second, we fine-tune the pre-trained encoder on few-labeled examples (10 hours) in the target language unseen during pre-training (E.g., Arabic). Third, we start with the fine-tuned target language ASR model and iteratively improve its performance by using unlabeled speech (100 hours) in the target language via DUST.

We make the following **interesting observations**. Pre-trained English speech encoders trained using the Wav2Vec-2.0 SSL framework can be used to build decent ASR models for languages other than English. We fine-tuned different Wav2Vec-2.0 encoders on eight target languages and found the ASR performance is dramatically better than fine-tuning a randomly initialized speech encoder on the target language. By using just 100 hours of unlabeled speech in a target language, we can achieve significant improvements in performance by using DUST; the ASR performance is at par with a multilingual pre-trained speech encoder fine-tuned on the downstream ASR task in the target language. Our work proposes a departure from the traditional multilingual pre-training, followed by a target language fine-tuning sequential transfer learning framework since our three-step method requires much less data than the two-step multilingual method.

7.1.2 Future Work

Our work has several future extensions. In particular, future work should explore alternate methods for computing the model's confidence other than sampling multiple predictions from the model and using sample agreement as the proxy for the model's confidence, like DUST. This is because generating samples requires running beam search decoding numerous times, which could be expensive, especially if the size of the unlabeled set is significant, which is usually the case. Some other research questions for future work are the following. Analyze the impact of initializing the student model with the parameters of the teacher in each DUST iteration, which could lead to faster

student model training compared to learning the task from scratch. But, this could also lead to the student model parameters collapsing to the same values as the teacher model. It might be better to give the student model freedom to explore and hence, converge to a different, better parameter configuration than the teacher. Another interesting direction would be to explore the impact of the size of unlabeled target domain data and labeled source domain data on the domain adaptation performance.

7.2 Cross-Lingual Transfer Learning

7.2.1 Summary

Cross-lingual learning is a learning paradigm where we train a language processing model, such as Speech-to-Text translation, on a subset of languages and expect it to do well with little to no fine-tuning on another unseen set of languages during training. A strategy for cross-lingual learning in speech processing is the following two-step process: First, pre-train a single multilingual speech encoder using (labeled, unlabeled, or both) speech data collected from several languages. Second, fine-tune the pre-trained multilingual encoder on downstream language processing tasks, such as speech translation.

The pre-training step is essential for achieving good cross-lingual transfer since the downstream application is built on top of the representations learned by the pre-trained encoder. Multilingual pre-training promises that the inductive bias of having a single model shared across languages would lead to the model encoding abstract high-level (possibly semantics) linguistic knowledge from speech. Our work shows that the current pre-training paradigms, such as XLS-R and others, fall short of this promise.

To address the shortcoming of the multilingual pre-training paradigms mentioned above, in **Chapter 5**, we propose a joint multilingual speech-text embedding space learning framework to inject semantic knowledge in the learned representation space of the pre-trained multilingual speech encoder using semantic supervision from the

text modality. We fine-tune the speech encoder in a teacher/student learning framework, utilizing a pre-trained multilingual semantic text encoder Language-Agnostic BERT as the teacher for training the student speech encoder. Chapter 5 analyzes the semantic structure of the joint speech-text embedding space learned due to our learning framework using several cross-lingual speech-to-text and speech-to-speech translation retrieval tasks. Unlike previous representation learning methods, we show that our approach can learn a semantically structured speech embedding space. We term our speech encoder the Sematically-Aligned Multimodal Cross-Lingual Speech Representations (SAMU-XLS-R).

In Chapter 6, we apply the SAMU-XLS-R speech encoder for translation. We develop a multilingual transformer encoder-decoder model for end-to-end automatic speech-to-text translation. We build several translation models and compare their performance on public speech-to-text translation benchmarks. We initialize the encoder of the translation model using different multilingual speech encoders for performance comparison, including our proposed speech encoder SAMU-XLS-R. The decoder of all translation models is initialized with the decoder of a pre-trained text-to-text translation model MBART. Each translation model (corresponding to different multilingual speech encoders) is simultaneously trained on several translation tasks. An example of a translation task is FR–>EN, where speech utterances are in French, and the task is to generate their English text translations. We show that under different translation task settings, such as multilingual and zero-shot translation, the translation model initialized using SAMU-XLS-R speech encoder achieves significantly better cross-lingual transfer from high to low-resource translation tasks than the models initialized using other multilingual speech encoders. We argue that this is due to SAMU-XLS-R encoding semantic knowledge in its learned speech representations.

7.2.2 Future Work

We require a pre-trained text encoder as semantic supervision for fine-tuning the speech encoder. However, it could be expensive to train such a massive text encoder.

Therefore, future work could explore extracting semantic supervision from the visual modality. It would require a focus on data engineering to extract meaningful examples of semantic correspondences between speech and visual modality in several spoken languages. Currently, it is not clear how to build such a dataset cheaply. A CommonVoice-like (Ardila et al., 2020) project is required, where instead of prompting humans with text on the web, they are prompted with images on the web and asked to provide a spoken caption for the image in their language. Past work has explored using visual modality as a source of semantic supervision (Harwath, Torralba, and J. Glass, 2016). Another source of semantic supervision worth exploring is the encoders of large text-to-text translation models, such as the recently introduced NLLB-200 model (Costa-jussà et al., 2022), which supports 200 written languages, unlike 109 supported by the Language-Agnostic BERT encoder used in our work. NLLB-200 encoder can help expand the number of languages our speech encoder SAMU-XLS-R supports.

Bibliography

Adams, Oliver et al. (June 2019a). "Massively Multilingual Adversarial Speech Recognition". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 96–108. DOI: 10.18653/v1/N19-1009. URL: https://aclanthology.org/N19-1009.

— (2019b). "Massively multilingual adversarial speech recognition". In: *arXiv preprint arXiv:1904.02210*.

Ali, Ahmed et al. (2019). *The MGB-2 Challenge: Arabic Multi-Dialect Broadcast Media Recognition*. arXiv: 1609.05625 [cs.CL].

Ao, Junyi et al. (2021). "Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing". In: *arXiv preprint arXiv:2110.07205*.

Ardila, Rosana et al. (2020). "Common Voice: A Massively-Multilingual Speech Corpus". In: *arXiv preprint arXiv:1912.06670*.

Arivazhagan, Naveen et al. (2019). "The missing ingredient in zero-shot neural machine translation". In: *arXiv preprint arXiv:1903.07091*.

Artetxe, Mikel and Holger Schwenk (Nov. 2019a). "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond". In: *Transactions of the Association for Computational Linguistics* 7, pp. 597–610. DOI: 10.1162/tacl_a_00288. URL: https://doi.org/10.1162%2Ftacl_a_00288.

— (Nov. 2019b). "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond". In: *Transactions of the Association for Computa-*

tional Linguistics 7, pp. 597–610. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00288. URL: http://dx.doi.org/10.1162/tacl_a_00288.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). *Layer Normalization*. arXiv: 1607.06450 [stat.ML].

Babu, Arun et al. (2021). *XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale*. DOI: 10.48550/ARXIV.2111.09296. URL: https://arxiv.org/abs/2111.09296.

Baevski, Alexei et al. (2020). "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations". In: *arXiv preprint arXiv:abs/2006.11477*.

Bahl, L. R. et al. (1991). "Context Dependent Modeling of Phones in Continuous Speech Using Decision Trees". In: *Proceedings of the Workshop on Speech and Natural Language*. HLT '91. Pacific Grove, California: Association for Computational Linguistics, pp. 264–269. DOI: 10.3115/112405.112453. URL: https://doi.org/10.3115/112405.112453.

Banerjee, Satanjeev and Alon Lavie (June 2005). "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 65–72. URL: https://www.aclweb.org/anthology/W05-0909.

Bapna, Ankur, Colin Cherry, et al. (2022a). "mSLAM: Massively multilingual joint pre-training for speech and text". In: *arXiv preprint arXiv:2202.01374*.

— (2022b). "mSLAM: Massively multilingual joint pre-training for speech and text". In: *arXiv:2202.01374*. DOI: 10.48550/ARXIV.2202.01374. URL: https://arxiv.org/abs/2202.01374.

Bapna, Ankur, Yu-an Chung, et al. (2021). "SLAM: A Unified Encoder for Speech and Language Modeling via Speech-Text Joint Pre-Training". In: *arXiv preprint arXiv:2110.10329*.

Baum, Eric and Frank Wilczek (1987). "Supervised Learning of Probability Distributions by Neural Networks". In: *Neural Information Processing Systems*. Ed. by

D. Anderson. Vol. 0. American Institute of Physics. URL: https://proceedings.neurips.cc/paper/1987/file/eccbc87e4b5ce2fe28308fd9f2a7baf3-Paper.pdf.

Baxter, Jonathan (2000). "A model of inductive bias learning". In: *Journal of artificial intelligence research* 12, pp. 149–198.

Bell, Peter et al. (2020). "Adaptation Algorithms for Speech Recognition: An Overview". In: *arXiv preprint arXiv:2008.06580*.

Bellegarda, J.R. and D. Nahamoo (1990). "Tied mixture continuous parameter modeling for speech recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38.12, pp. 2033–2045. DOI: 10.1109/29.61531.

Blum, Avrim and Shuchi Chawla (June 2001). "Learning from labeled and unlabeled data using graph mincuts". In: *Proc. ICML*.

Bommasani, Rishi and et. al. (2021). "On the Opportunities and Risks of Foundation Models". In: *arXiv:2108.07258*. DOI: 10.48550/ARXIV.2108.07258. URL: https://arxiv.org/abs/2108.07258.

Brown, Tom B. et al. (2020). *Language Models are Few-Shot Learners*. DOI: 10.48550/ARXIV.2005.14165. URL: https://arxiv.org/abs/2005.14165.

Caruana, Rich (July 1997). "Multitask Learning". In: *Machine Learning* 28. DOI: 10.1023/A:1007379606734.

Chan, William et al. (2016). "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964. DOI: 10.1109/ICASSP.2016.7472621.

Chen, Sanyuan et al. (2021). "Wavlm: Large-scale self-supervised pre-training for full stack speech processing". In: *arXiv preprint arXiv:2110.13900*.

Chen, Ting et al. (2020). *A Simple Framework for Contrastive Learning of Visual Representations*. arXiv: 2002.05709 [cs.LG].

Cheng, Yong et al. (2022). "Mu2SLAM: Multitask, Multilingual Speech and Language Models". In: *arXiv preprint arXiv:2212.09553*.

Chiu, Chung-Cheng et al. (2018). *State-of-the-art Speech Recognition With Sequence-to-Sequence Models*. arXiv: 1712.01769 [cs.CL].

Chorowski, Jan et al. (2015a). *Attention-Based Models for Speech Recognition*. DOI: 10.48550/ARXIV.1506.07503. URL: https://arxiv.org/abs/1506.07503.

— (Dec. 2015b). "Attention-Based Models for Speech Recognition". In: *Proc. NIPS*.

Chung, Yu-An and James Glass (2020). "Generative pre-training for speech with autoregressive predictive coding". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3497–3501.

Chung, Yu-An, Yu Zhang, et al. (2021). *W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training*. DOI: 10.48550/ARXIV.2108.06209. URL: https://arxiv.org/abs/2108.06209.

Collobert, Ronan, Christian Puhrsch, and Gabriel Synnaeve (2016). *Wav2Letter: an End-to-End ConvNet-based Speech Recognition System*. DOI: 10.48550/ARXIV.1609.03193. URL: https://arxiv.org/abs/1609.03193.

Conneau, Alexis, Alexei Baevski, et al. (2020). *Unsupervised Cross-lingual Representation Learning for Speech Recognition*. arXiv: 2006.13979 [cs.CL].

Conneau, Alexis, Kartikay Khandelwal, et al. (2019a). "Unsupervised Cross-lingual Representation Learning at Scale". In: *arXiv:1911.02116*. DOI: 10.48550/ARXIV.1911.02116. URL: https://arxiv.org/abs/1911.02116.

— (2019b). *Unsupervised Cross-lingual Representation Learning at Scale*. DOI: 10.48550/ARXIV.1911.02116. URL: https://arxiv.org/abs/1911.02116.

Costa-jussà, Marta R et al. (2022). "No language left behind: Scaling human-centered machine translation". In: *arXiv preprint arXiv:2207.04672*.

Courty, Nicolas et al. (2015). "Optimal Transport for Domain Adaptation". In: *arXiv preprint arXiv:1507.00504*.

DeSa, Virginia R. (1993). "Learning Classification with Unlabeled Data". In: *Proceedings of the 6th International Conference on Neural Information Processing Systems*. NIPS'93. Denver, Colorado: Morgan Kaufmann Publishers Inc., pp. 112–119.

Devlin, Jacob et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].

Di Gangi, Mattia A et al. (2019a). "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)*. Association for Computational Linguistics, pp. 2012–2017.

— (June 2019b). "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2012–2017. DOI: 10.18653/v1/N19-1202. URL: https://aclanthology.org/N19-1202.

Doddington, George (2002). "Automatic Evaluation of Machine Translation Quality Using N-Gram Co-Occurrence Statistics". In: *Proceedings of the Second International Conference on Human Language Technology Research*. HLT '02. San Diego, California: Morgan Kaufmann Publishers Inc., pp. 138–145.

Dong, L., S. Xu, and B. Xu (Apr. 2018). "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition". In: *Proc. ICASSP*, pp. 5884–5888. DOI: 10.1109/ICASSP.2018.8462506.

Duong, Long, Antonios Anastasopoulos, et al. (2016). "An attentional model for speech translation without transcription". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 949–959.

Duong, Long, Trevor Cohn, et al. (July 2015). "Low Resource Dependency Parsing: Cross-lingual Parameter Sharing in a Neural Network Parser". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, pp. 845–850. DOI: 10.3115/v1/P15-2139. URL: https://aclanthology.org/P15-2139.

Duquenne, Paul-Ambroise, Hongyu Gong, and Holger Schwenk (2021). "Multimodal and multilingual embeddings for large-scale speech mining". In: *Advances in Neural Information Processing Systems* 34.

Eddine, Moussa Kamal et al. (2021). "FrugalScore: Learning Cheaper, Lighter and Faster Evaluation Metrics for Automatic Text Generation". In: *arXiv preprint arXiv:2110.08559*.

Feng, Fangxiaoyu et al. (2020). *Language-agnostic BERT Sentence Embedding*. DOI: 10.48550/ARXIV.2007.01852. URL: https://arxiv.org/abs/2007.01852.

Gal, Yarin and Zoubin Ghahramani (June 2016). "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning". In: *Proc. ICML*, pp. 1050–1059.

Ganin, Yaroslav et al. (2016). "Domain-adversarial training of neural networks". In: *JMLR* 17.1, pp. 2096–2030.

Glass, James R (2003). "A probabilistic framework for segment-based speech recognition". In: *Computer Speech & Language* 17.2. New Computational Paradigms for Acoustic Modeling in Speech Recognition, pp. 137–152. ISSN: 0885-2308. DOI: https://doi.org/10.1016/S0885-2308(03)00006-8. URL: https://www.sciencedirect.com/science/article/pii/S0885230803000068.

Godfrey, John J, Edward C Holliman, and Jane McDaniel (Mar. 1992). "SWITCHBOARD: Telephone speech corpus for research and development". In: *Proc. ICASSP*. Vol. 1, pp. 517–520.

Graves, Alex (2012). "Sequence Transduction with Recurrent Neural Networks". In: *arXiv preprint arXiv:abs/1211.3711*.

Graves, Alex, Santiago Fernández, et al. (June 2006). "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks". In: *Proc. ICML*.

Graves, Alex and Navdeep Jaitly (22–24 Jun 2014). "Towards End-To-End Speech Recognition with Recurrent Neural Networks". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Je-

bara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, pp. 1764–1772. URL: https://proceedings.mlr.press/v32/graves14.html.

Graves, Alex, Navdeep Jaitly, and Abdel-rahman Mohamed (2013). "Hybrid speech recognition with Deep Bidirectional LSTM". In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 273–278. DOI: 10.1109/ASRU.2013.6707742.

Graves, Alex, Abdelrahman Mohamed, and Geoffrey E. Hinton (May 2013). "Speech recognition with deep recurrent neural networks". In: *Proc. ICASSP*.

Gu, Jiatao et al. (July 2019). "Improved Zero-shot Neural Machine Translation via Ignoring Spurious Correlations". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 1258–1268. DOI: 10.18653/v1/P19-1121. URL: https://aclanthology.org/P19-1121.

Hannun, Awni (2017). "Sequence Modeling with CTC". In: *Distill*. https://distill.pub/2017/ctc. DOI: 10.23915/distill.00008.

Hannun, Awni, Carl Case, et al. (2014). *Deep Speech: Scaling up end-to-end speech recognition*. DOI: 10.48550/ARXIV.1412.5567. URL: https://arxiv.org/abs/1412.5567.

Hannun, Awni, Ann Lee, et al. (2019). *Sequence-to-Sequence Speech Recognition with Time-Depth Separable Convolutions*. DOI: 10.48550/ARXIV.1904.02619. URL: https://arxiv.org/abs/1904.02619.

Harwath, David, Wei-Ning Hsu, and James Glass (2020). "Learning Hierarchical Discrete Linguistic Units from Visually-Grounded Speech". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=B1elCp4KwH.

Harwath, David, Antonio Torralba, and James Glass (2016). "Unsupervised learning of spoken language with visual context". In: *Advances in Neural Information Processing Systems* 29.

He, Junxian et al. (2019). "Revisiting self-training for neural sequence generation". In: *arXiv preprint arXiv:1909.13788*.

Heafield, Kenneth (2011). "KenLM: Faster and smaller language model queries". In: *Proc. WMT*, pp. 187–197.

Hernandez, Francois et al. (2018a). "TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation". In: *Lecture Notes in Computer Science*, pp. 198–208. ISSN: 1611-3349. DOI: 10.1007/978-3-319-99579-3_21. URL: http://dx.doi.org/10.1007/978-3-319-99579-3_21.

— (Sept. 2018b). "TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation". In: *Proc. SPECOM*. Springer, pp. 198–208.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean (2015). "Distilling the Knowledge in a Neural Network". In: *arXiv preprint arXiv:1503.02531*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hori, Takaaki, Shinji Watanabe, and John R. Hershey (2017). "Joint CTC/attention decoding for end-to-end speech recognition". In: *Proc. ACL*.

Houlsby, Neil et al. (2019). "Parameter-Efficient Transfer Learning for NLP". In: *ICML*.

Hsu, Wei-Ning, Benjamin Bolte, et al. (2021). "Hubert: Self-supervised speech representation learning by masked prediction of hidden units". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29, pp. 3451–3460.

Hsu, Wei-Ning, Ann Lee, et al. (2020). "Semi-Supervised Speech Recognition via Local Prior Matching". In: *arXiv preprint arXiv:2002.10336*.

Hsu, Wei-Ning, Anuroop Sriram, et al. (2021). *Robust wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training*. arXiv: 2104.01027 [cs.SD].

Inaguma, Hirofumi et al. (2020). "ESPnet-ST: All-in-one speech translation toolkit". In: *arXiv preprint arXiv:2004.10234*.

Iranzo-Sánchez, Javier et al. (2019). "Europarl-ST: A Multilingual Corpus For Speech Translation Of Parliamentary Debates". In: *arXiv:1911.03167*. DOI: 10.48550/ARXIV.1911.03167. URL: https://arxiv.org/abs/1911.03167.

Kahn, J. et al. (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). https://github.com/facebookresearch/libri-light, pp. 7669–7673.

Kahn, Jacob, Ann Lee, and Awni Hannun (May 2020). "Self-Training for End-to-End Speech Recognition". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. DOI: 10.1109/icassp40776.2020.9054295. URL: http://dx.doi.org/10.1109/ICASSP40776.2020.9054295.

Karita, Shigeki et al. (Dec. 2019). "A comparative study on transformer vs RNN in speech applications". In: *Proc. ASRU*.

Khurana, Sameer, Antoine Laurent, and James Glass (2021). *Magic dust for cross-lingual adaptation of monolingual wav2vec-2.0*. arXiv: 2110.03560 [cs.CL].

— (2022). "SAMU-XLSR: Semantically-Aligned Multimodal Utterance-level Cross-Lingual Speech Representation". In: *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–13. DOI: 10.1109/JSTSP.2022.3192714.

Khurana, Sameer, Antoine Laurent, Wei-Ning Hsu, et al. (2020). *A Convolutional Deep Markov Model for Unsupervised Speech Representation Learning*. arXiv: 2006.02547 [eess.AS].

Khurana, Sameer, Niko Moritz, et al. (2021). "Unsupervised Domain Adaptation for Speech Recognition via Uncertainty Driven Self-Training". In: *Proc. ICASSP*. arXiv: 2011.13439 [cs.CL].

Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization". In: *arXiv:1412.6980*. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

Ko, Tom, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur (2015). "Audio augmentation for speech recognition". In: *Proc. Interspeech 2015*, pp. 3586–3589. DOI: 10.21437/Interspeech.2015-711.

Ko, Tom, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, et al. (Mar. 2017a). "A study on data augmentation of reverberant speech for robust speech recognition". In: *Proc. ICASSP*.

Ko, Tom, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, et al. (2017b). "A study on data augmentation of reverberant speech for robust speech recognition". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5220–5224. DOI: 10.1109/ICASSP.2017.7953152.

Kullback, S. and R. A. Leibler (Mar. 1951). "On Information and Sufficiency". In: *Ann. Math. Statist.* 22.1, pp. 79–86. DOI: 10.1214/aoms/1177729694. URL: https://doi.org/10.1214/aoms/1177729694.

Lample, Guillaume and Alexis Conneau (2019a). *Cross-lingual Language Model Pretraining*. DOI: 10.48550/ARXIV.1901.07291. URL: https://arxiv.org/abs/1901.07291.

— (2019b). "Cross-lingual language model pretraining". In: *arXiv preprint arXiv:1901.07291*.

Laperrière, Gaëlle et al. (2022). "On the Use of Semantically-Aligned Speech Representations for Spoken Language Understanding". In: *arXiv:2210.05291*. DOI: 10.48550/ARXIV.2210.05291. URL: https://arxiv.org/abs/2210.05291.

Lee, K.-F. (1990). "Context-independent phonetic hidden Markov models for speaker-independent continuous speech recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38.4, pp. 599–609. DOI: 10.1109/29.52701.

Levenshtein, V. I. (Feb. 1966). "Binary Codes Capable of Correcting Deletions, Insertions and Reversals". In: *Soviet Physics Doklady* 10, p. 707.

Li, Jinyu et al. (Sept. 2014). "Learning small-size DNN with output-distribution-based criteria". In: *Proc. Interspeech*.

Li, Xian et al. (2020a). "Multilingual Speech Translation with Efficient Finetuning of Pretrained Models". In: *arXiv:2010.12829*. DOI: 10.48550/ARXIV.2010.12829. URL: https://arxiv.org/abs/2010.12829.

— (2020b). "Multilingual speech translation with efficient finetuning of pretrained models". In: *arXiv preprint arXiv:2010.12829*.

Lin, Chin-Yew (July 2004). "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, pp. 74–81. URL: https://aclanthology.org/W04-1013.

Liptchinsky, Vitaliy, Gabriel Synnaeve, and Ronan Collobert (2017). *Letter-Based Speech Recognition with Gated ConvNets*. DOI: 10.48550/ARXIV.1712.09444. URL: https://arxiv.org/abs/1712.09444.

Liu, Alexander H., Yu-An Chung, and James Glass (2020). *Non-Autoregressive Predictive Coding for Learning Speech Representations from Local Dependencies*. DOI: 10.48550/ARXIV.2011.00406. URL: https://arxiv.org/abs/2011.00406.

Liu, Yinhan et al. (2020a). *Multilingual Denoising Pre-training for Neural Machine Translation*. DOI: 10.48550/ARXIV.2001.08210. URL: https://arxiv.org/abs/2001.08210.

— (2020b). "Multilingual Denoising Pre-training for Neural Machine Translation". In: *Transactions of the Association for Computational Linguistics* 8, pp. 726–742. DOI: 10.1162/tacl_a_00343. URL: https://aclanthology.org/2020.tacl-1.47.

Maas, Andrew et al. (May 2015). "Lexicon-Free Conversational Speech Recognition with Neural Networks". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 345–354. DOI: 10.3115/v1/N15-1038. URL: https://aclanthology.org/N15-1038.

Micikevicius, Paulius et al. (2017). "Mixed Precision Training". In: *arXiv:1710.03740*. DOI: 10.48550/ARXIV.1710.03740. URL: https://arxiv.org/abs/1710.03740.

Mohamed, Abdelrahman, Dmytro Okhonko, and Luke Zettlemoyer (2019). *Transformers with convolutional context for ASR*. DOI: 10.48550/ARXIV.1904.11660. URL: https://arxiv.org/abs/1904.11660.

Mohri, Mehryar, Fernando Pereira, and Michael Riley (2008). "Speech Recognition with Weighted Finite-State Transducers". In: *Springer Handbook of Speech Processing*. Ed. by Jacob Benesty, M. Mohan Sondhi, and Yiteng Arden Huang. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 559–584. ISBN: 978-3-540-49127-9. DOI: 10.1007/978-3-540-49127-9_28. URL: https://doi.org/10.1007/978-3-540-49127-9_28.

Morgan, N. and H. Bourlard (1995). "Continuous speech recognition". In: *IEEE Signal Processing Magazine* 12.3, pp. 24–42. DOI: 10.1109/79.382443.

Moritz, Niko, Takaaki Hori, and Jonathan Le Roux (2020). "Semi-Supervised Speech Recognition via Graph-based Temporal Classification". In: *arXiv preprint arXiv:2010.15653*.

Müller, Rafael, Simon Kornblith, and Geoffrey Hinton (2019). "When Does Label Smoothing Help?" In: *arXiv:1906.02629*. DOI: 10.48550/ARXIV.1906.02629. URL: https://arxiv.org/abs/1906.02629.

Nakamura, Satoshi et al. (2006). "The ATR multilingual speech-to-speech translation system". In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.2, pp. 365–376.

Neto, Joao et al. (1995). "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system". In.

Newell, Alan F. (1973). "Speech understanding systems : Final report of a study group". In.

Ney, Hermann (1999). "Speech translation: Coupling of recognition and translation". In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*. Vol. 1. IEEE, pp. 517–520.

Nigam, Kamal et al. (2000). "Text classification from labeled and unlabeled documents using EM". In: *Machine Learning* 39.2-3, pp. 103–134.

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals (2019). *Representation Learning with Contrastive Predictive Coding*. arXiv: 1807.03748 [cs.LG].

Ortmanns, S. and H. Ney (2000). "The time-conditioned approach in dynamic programming search for LVCSR". In: *IEEE Transactions on Speech and Audio Processing* 8.6, pp. 676–687. DOI: 10.1109/89.876301.

Ott, Myle et al. (2019). "fairseq: A Fast, Extensible Toolkit for Sequence Modeling". In: *arXiv:1904.01038*. DOI: 10.48550/ARXIV.1904.01038. URL: https://arxiv.org/abs/1904.01038.

Pan, Sinno Jialin and Qiang Yang (2010). "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10, pp. 1345–1359. DOI: 10.1109/TKDE.2009.191.

Panayotov, V. et al. (Apr. 2015). "LibriSpeech: An ASR corpus based on public domain audio books". In: *Proc. ICASSP*.

Park, Daniel S, William Chan, et al. (2019). "SpecAugment: A simple data augmentation method for automatic speech recognition". In: *arXiv preprint arXiv:1904.08779*.

Park, Daniel S, Yu Zhang, et al. (2020). "Improved noisy student training for automatic speech recognition". In: *arXiv preprint arXiv:2005.09629*.

Pascual, Santiago et al. (2019). *Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks*. DOI: 10.48550/ARXIV.1904.03416. URL: https://arxiv.org/abs/1904.03416.

Paul, Douglas B and Janet Baker (Feb. 1992). "The design for the Wall Street Journal-based CSR corpus". In: *Proc. SNLW*.

Paul, Douglas B. and Janet M. Baker (1992). "The Design for the Wall Street Journal-based CSR Corpus". In: *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*. URL: https://aclanthology.org/H92-1073.

Pfeiffer, Jonas, Aishwarya Kamath, et al. (2021). *AdapterFusion: Non-Destructive Task Composition for Transfer Learning*. arXiv: 2005.00247 [cs.CL].

Pfeiffer, Jonas, Ivan Vulić, et al. (2020). *MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer*. arXiv: 2005.00052 [cs.CL].

Pino, Juan et al. (2020a). "Self-Training for End-to-End Speech Translation". In: *Proc. Interspeech 2020*, pp. 1476–1480. DOI: 10.21437/Interspeech.2020-2938.

— (2020b). "Self-training for end-to-end speech translation". In: *arXiv preprint arXiv:2006.02490*.

Post, Matt (Oct. 2018). "A Call for Clarity in Reporting BLEU Scores". In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. Belgium, Brussels: Association for Computational Linguistics, pp. 186–191. URL: https://www.aclweb.org/anthology/W18-6319.

Povey, Daniel, Arnab Ghoshal, et al. (Dec. 2011). "The Kaldi speech recognition toolkit". In: *Proc. ASRU*.

Povey, Daniel, Vijayaditya Peddinti, et al. (Sept. 2016). "Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI". In: *Proc. Interspeech*, pp. 2751–2755.

Pratap, Vineel et al. (2020). "MLS: A Large-Scale Multilingual Dataset for Speech Research". In: *ArXiv* abs/2012.03411.

Rabiner, L. and B. Juang (1986). "An introduction to hidden Markov models". In: *IEEE ASSP Magazine* 3.1, pp. 4–16. DOI: 10.1109/MASSP.1986.1165342.

Rabiner, Lawrence and Biing-Hwang Juang (1993). *Fundamentals of speech recognition*. Prentice-Hall, Inc.

Radford, Alec, Jong Wook Kim, et al. (n.d.). "Robust Speech Recognition via Large-Scale Weak Supervision". In: ().

Radford, Alec and Karthik Narasimhan (2018). "Improving Language Understanding by Generative Pre-Training". In.

Renals, Steve et al. (1994). "Connectionist probability estimators in HMM speech recognition". In: *IEEE transactions on speech and audio processing* 2.1, pp. 161–174.

Rivière, Morgane et al. (2020). *Unsupervised pretraining transfers well across languages*. arXiv: 2002.02848 [eess.AS].

Robbins, Herbert and Sutton Monro (1951). "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3, pp. 400–407. DOI: 10.1214/aoms/1177729586. URL: https://doi.org/10.1214/aoms/1177729586.

Robinson, Tony, Mike Hochberg, and Steve Renals (1996). "The use of recurrent neural networks in continuous speech recognition". In: *Automatic speech and speaker recognition*. Springer, pp. 233–258.

Rosenberg, Andrew et al. (2022). "MAESTRO: Matched Speech Text Representations through Modality Matching". In.

Rouditchenko, Andrew et al. (2020). *AVLnet: Learning Audio-Visual Language Representations from Instructional Videos*. DOI: 10.48550/ARXIV.2006.09199. URL: https://arxiv.org/abs/2006.09199.

Ruder, Sebastian (2019). "Neural Transfer Learning for Natural Language Processing". PhD thesis. National University of Ireland, Galway.

Ruder, Sebastian, Anders Søgaard, and Ivan Vulić (July 2019). "Unsupervised Cross-Lingual Representation Learning". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Florence, Italy: Association for Computational Linguistics, pp. 31–38. DOI: 10.18653/v1/P19-4007. URL: https://aclanthology.org/P19-4007.

Rybach, David et al. (2011). "RASR - The RWTH Aachen University Open Source Speech Recognition Toolkit". In.

Safari, Pooyan, Miquel India, and Javier Hernando (2020). "Self-attention encoding and pooling for speaker recognition". In: *arXiv preprint arXiv:2008.01077*.

Saito, Kuniaki et al. (June 2018). "Maximum classifier discrepancy for unsupervised domain adaptation". In: *Proc. CVPR*, pp. 3723–3732.

Salazar, Julian, Katrin Kirchhoff, and Zhiheng Huang (May 2019). "Self-attention Networks for Connectionist Temporal Classification in Speech Recognition". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. DOI: 10.1109/icassp.2019.8682539. URL: https://doi.org/10.1109%5C%2Ficassp.2019.8682539.

Salesky, Elizabeth et al. (2021). *The Multilingual TEDx Corpus for Speech Recognition and Translation*. DOI: 10.48550/ARXIV.2102.01757. URL: https://arxiv.org/abs/2102.01757.

Schmidhuber, Jürgen (1990). *Making the World Differentiable: On Using Self-Supervised Fully Recurrent Neural Networks for Dynamic Reinforcement Learning and Planning in Non-Stationary Environments*. Tech. rep.

Schneider, Steffen et al. (2019). *wav2vec: Unsupervised Pre-training for Speech Recognition*. DOI: 10.48550/ARXIV.1904.05862. URL: https://arxiv.org/abs/1904.05862.

Schuster, Mike and Kaisuke Nakajima (2012). "Japanese and Korean voice search". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5149–5152. DOI: 10.1109/ICASSP.2012.6289079.

Schwartz, R. et al. (1985). "Context-dependent modeling for acoustic-phonetic recognition of continuous speech". In: *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 10, pp. 1205–1208. DOI: 10.1109/ICASSP.1985.1168283.

Schwenk, Holger (2018). "Filtering and Mining Parallel Data in a Joint Multilingual Space". In: DOI: 10.48550/ARXIV.1805.09822. URL: https://arxiv.org/abs/1805.09822.

Schwenk, Holger, Vishrav Chaudhary, et al. (2019). *WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia*. DOI: 10.48550/ARXIV.1907.05791. URL: https://arxiv.org/abs/1907.05791.

Schwenk, Holger and Matthijs Douze (2017a). *Learning Joint Multilingual Sentence Representations with Neural Machine Translation*. DOI: 10.48550/ARXIV.1704.04154. URL: https://arxiv.org/abs/1704.04154.

— (2017b). *Learning Joint Multilingual Sentence Representations with Neural Machine Translation*. DOI: 10.48550/ARXIV.1704.04154. URL: https://arxiv.org/abs/1704.04154.

Schwenk, Holger, Guillaume Wenzek, et al. (2019a). *CCMatrix: Mining Billions of High-Quality Parallel Sentences on the WEB*. DOI: 10.48550/ARXIV.1911.04944. URL: https://arxiv.org/abs/1911.04944.

— (2019b). *CCMatrix: Mining Billions of High-Quality Parallel Sentences on the WEB*. DOI: 10.48550/ARXIV.1911.04944. URL: https://arxiv.org/abs/1911.04944.

Scudder, H (1965). "Probability of error of some adaptive pattern-recognition machines". In: *IEEE Trans. Inf. Theory* 11.3, pp. 363–371.

Shor, Joel et al. (2021). "Universal Paralinguistic Speech Representations Using Self-Supervised Conformers". In: *arXiv:2110.04621*. DOI: 10.48550/ARXIV.2110.04621. URL: https://arxiv.org/abs/2110.04621.

Snyder, David, Guoguo Chen, and Daniel Povey (2015a). *MUSAN: A Music, Speech, and Noise Corpus*. arXiv:1510.08484v1. eprint: 1510.08484.

— (2015b). "Musan: A music, speech, and noise corpus". In: *arXiv preprint arXiv:1510.08484*.

Sperber, Matthias et al. (2018). *Self-Attentional Acoustic Models*. DOI: 10.48550/ARXIV.1803.09519. URL: https://arxiv.org/abs/1803.09519.

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *JMLR* 15, pp. 1929–1958.

Sun, Sining et al. (Apr. 2018). "Domain adversarial training for accented speech recognition". In: *Proc. ICASSP*, pp. 4854–4858.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014). "Sequence to Sequence Learning with Neural Networks". In: *arXiv:1409.3215*. DOI: 10.48550/ARXIV.1409.3215. URL: https://arxiv.org/abs/1409.3215.

Swietojanski, Pawel, Arnab Ghoshal, and Steve Renals (2013). "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition". In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 285–290. DOI: 10.1109/ASRU.2013.6707744.

Szegedy, Christian et al. (2016a). "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.

— (2016b). "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

Vaswani, Ashish et al. (2017). *Attention Is All You Need*. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

Vyas, Apoorv et al. (May 2019). "Analyzing uncertainties in speech recognition using dropout". In: *Proc. ICASSP*, pp. 6730–6734.

Wang, Changhan, Juan Pino, et al. (2020). *CoVoST: A Diverse Multilingual Speech-To-Text Translation Corpus*. DOI: 10.48550/ARXIV.2002.01320. URL: https://arxiv.org/abs/2002.01320.

Wang, Changhan, Morgane Riviere, et al. (Aug. 2021). "VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation". 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)*. Online: Association for Computational Linguistics, pp. 993–1003. DOI: 10.18653/v1/2021.acl-long.80. URL: https://aclanthology.org/2021.acl-long.80.

Wang, Dong and Thomas Fang Zheng (2015). *Transfer Learning for Speech and Language Processing*. arXiv: 1511.06066 [cs.CL].

Watanabe, Shinji, Florian Boyer, Xuankai Chang, Pengcheng Guo, Tomoki Hayashi, Yosuke Higuchi, Takaaki Hori, Wen-Chin Huang, Hirofumi Inaguma, Naoyuki Kamo, et al. (2021). "The 2020 espnet update: new features, broadened applications, performance improvements, and future plans". In: *2021 IEEE Data Science and Learning Workshop (DSLW)*. IEEE, pp. 1–6.

Watanabe, Shinji, Florian Boyer, Xuankai Chang, Pengcheng Guo, Tomoki Hayashi, Yosuke Higuchi, Takaaki Hori, Wen-Chin Huang, Hirofumi Inaguma, Naoyuki Kamo, et al. (2020). *The 2020 ESPnet update: new features, broadened applications, performance improvements, and future plans*. arXiv: 2012.13006 [eess.AS].

Watanabe, Shinji, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, et al. (Sept. 2018). "ESPnet: End-to-End Speech Processing Toolkit". In: *Proc. Interspeech*, pp. 2207–2211.

Watanabe, Shinji, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, et al. (2018). *ESPnet: End-to-End Speech Processing Toolkit*. DOI: 10.48550/ARXIV.1804.00015. URL: https://arxiv.org/abs/1804.00015.

Watanabe, Shinji, Takaaki Hori, Suyoun Kim, et al. (2017a). "Hybrid CTC/Attention Architecture for End-to-End Speech Recognition". In: *IEEE Journal of Selected Topics in Signal Processing* 11.8, pp. 1240–1253. DOI: 10.1109/JSTSP.2017.2763455.

— (2017b). “Hybrid CTC/Attention Architecture for End-to-End Speech Recognition”. In: *IEEE Journal of Selected Topics in Signal Processing* 11, pp. 1240–1253.

Weninger, Felix et al. (Oct. 2020). “Semi-Supervised Learning with Data Augmentation for End-to-End ASR”. In: *Proc. Interspeech*.

Wikipedia contributors (2020). *Word error rate — Wikipedia, The Free Encyclopedia*. [Online; accessed 23-April-2022]. URL: https://en.wikipedia.org/w/index.php?title=Word_error_rate&oldid=939575741.

Wolf, Thomas et al. (2019). *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. DOI: 10.48550/ARXIV.1910.03771. URL: https://arxiv.org/abs/1910.03771.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, et al. (2016). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. DOI: 10.48550/ARXIV.1609.08144. URL: https://arxiv.org/abs/1609.08144.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, et al. (2016). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. arXiv: 1609.08144 [cs.CL].

Xie, Qizhe et al. (2020). “Self-training with noisy student improves imagenet classification”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10687–10698.

Xu, Qiantong et al. (2020). “Iterative Pseudo-Labeling for Speech Recognition”. In: *arXiv preprint arXiv:2005.09267*.

Yang, Shu-wen et al. (2021). *SUPERB: Speech processing Universal PERformance Benchmark*. arXiv: 2105.01051 [cs.CL].

Yang, Yongxin and Timothy M Hospedales (2016). “Trace norm regularised deep multi-task learning”. In: *arXiv preprint arXiv:1606.04038*.

Young, S. J., J. J. Odell, and P. C. Woodland (1994). "Tree-Based State Tying for High Accuracy Acoustic Modelling". In: *Proceedings of the Workshop on Human Language Technology*. HLT '94. Plainsboro, NJ: Association for Computational Linguistics, pp. 307–312. ISBN: 1558603573. DOI: 10.3115/1075812.1075885. URL: https://doi.org/10.3115/1075812.1075885.

Zeghidour, Neil et al. (2018). *Fully Convolutional Speech Recognition*. DOI: 10.48550/ARXIV.1812.06864. URL: https://arxiv.org/abs/1812.06864.

Zhou, Shiyu et al. (2018). *Syllable-Based Sequence-to-Sequence Speech Recognition with the Transformer in Mandarin Chinese*. DOI: 10.48550/ARXIV.1804.10752. URL: https://arxiv.org/abs/1804.10752.

Zhou, Zhi-Hua and Ming Li (2005). "Tri-training: Exploiting unlabeled data using three classifiers". In: *IEEE Trans. Knowl. Data Eng.* 17.11, pp. 1529–1541.

Zoph, Barret et al. (2020). *Rethinking Pre-training and Self-training*. arXiv: 2006.06882 [cs.CV].