*INTERSPEECH 2021*
30 August – 3 September, 2021, Brno, Czechia

# Robust wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training

*Wei-Ning Hsu*[1*], *Anuroop Sriram*[1*], *Alexei Baevski*[1], *Tatiana Likhomanenko*[1], *Qiantong Xu*[1], *Vineel Pratap*[1], *Jacob Kahn*[1], *Ann Lee*[1], *Ronan Collobert*[1], *Gabriel Synnaeve*[2], *Michael Auli*[1]

Facebook AI, USA
Facebook AI, France
{wnhsu,anuroops}@fb.com

## Abstract

Self-supervised learning of speech representations has been a very active research area but most work is focused on a single domain such as read audio books for which there exist large quantities of labeled and unlabeled data. In this paper, we explore more general setups where the domain of the unlabeled data for pre-training data differs from the domain of the labeled data for fine-tuning, which in turn may differ from the test data domain. Our experiments show that using target domain data during pre-training leads to large performance improvements across a variety of setups. With no access to in-domain labeled data, pre-training on unlabeled in-domain data closes 66-73% of the performance gap between the ideal setting of in-domain labeled data and a competitive supervised out-of-domain model. This has obvious practical implications since it is much easier to obtain unlabeled target domain data than labeled data. Moreover, we find that pre-training on multiple domains improves generalization performance on domains not seen during training. We will release pre-trained models.

**Index Terms**: speech recognition, self-supervised learning, unsupervised domain adaptation, robustness

## 1. Introduction

Self-supervised learning of speech representations has received a lot attention [1, 2, 3, 4, 5] and has been demonstrated to work well both in low- and high-resource labeled data settings for automatic speech recognition (ASR) [6]. However, the majority of studies focus on settings where there is little domain mismatch between the unlabeled data for pre-training, the labeled data for fine-tuning and the domain of the test data, or the *target domain*. It is well known that the performance of ASR systems trained from scratch with conventional supervised objectives can degrade significantly when tested on domains mismatched from training data [7, 8]. However, the impact of domain mismatch in self-supervised speech representation learning has been much less studied.

In this paper we present a series of experiments to better understand the impact of the various domain mismatches that can occur in the self-supervised data pipeline. We study the effect of increasing the amount of both in-domain and out-of-domain unlabeled data when fine-tuning the resulting model on both in-domain and out-of-domain labeled data. We also investigate the robustness of models on domains not seen during pre-training or fine-tuning. Our findings include that adding unlabeled data whose domain matches the test data always improves performance, even if the labeled data for fine-tuning is out-of-domain. This has immediate practical applications since it is much easier to obtain unlabeled data for a particular target domain than labeled data. We also find that pre-training on multiple domains increases robustness to completely unseen domains.

[*] Equal contribution.

## 2. Related Work

This paper is related to a large body of work on robust ASR and domain adaptation. There are two popular lines of approaches. The first one is feature-based, which focuses on creating robust features [9, 10]. Both signal processing-based [11] and learned [12] features have been explored. The other line is model-based, which exposes a model to diverse data while minimally pre-processing speech input in order to exploit the model capacity. This includes data augmentation [13, 14], self-training on target domain [15], domain adversarial training [16], and joint training [8]. The self-supervised approach explored in this paper can be categorized as model-based; however, unlike the aforementioned model-based methods, it does not require any labeled data during pre-training by using a self-supervised objective.

The most related work to this paper is [17], which investigated domain-shift for self-supervised learning, but did not dissect the domains of data used during pre-training. We extend this work by also examining the effect of pre-training data domain. Furthermore, the pre-trained feature extractor in [17] is fixed during supervised fine-tuning, making it more similar to feature-based approaches. Other related work includes pre-training on multiple languages and investigating how representations transfer between languages [18, 19].

## 3. Experimental Setup

We experiment with wav2vec 2.0 [6, 19] which consists of a convolutional feature encoder $f : \mathcal{X} \mapsto \mathcal{Z}$ to map raw audio $\mathbf{x}$ to latent speech representations $\mathbf{z}_1, \dots, \mathbf{z}_T$ input to a Transformer $g : \mathcal{Z} \mapsto \mathcal{C}$ to output context representations $\mathbf{c}_1, \dots, \mathbf{c}_T$ [20, 21]. Each $\mathbf{z}_t$ represents about 25ms of audio strided by 20ms and the Transformer architecture follows BERT [22, 23]. During training, latent representations are discretized to $\mathbf{q}_1, \dots, \mathbf{q}_T$ with a quantization module $\mathcal{Z} \mapsto \mathcal{Q}$ to represent the targets in the objective. The quantization module uses a Gumbel softmax to choose entries from $G = 2$ codebooks with $V = 320$ entries each and the chosen entries are concatenated to obtain $\mathbf{q}$ [24, 25, 20]. The model is trained to identify the true quantized latent $\mathbf{q}_t$ using $\mathbf{c}_t$ for each masked time-step within a set of $K = 100$ distractors $\mathbf{Q}_t$ sampled from other masked time steps.

Table 1 summarizes the setups explored in each subsequent section. Let in-domain and out-of-domain (OOD) be defined relative to the test data. We first study in Section 4.1 whether adding in-domain data for pre-training is beneficial when OOD data is used for fine-tuning. In Section 4.2, we study if adding data that are OOD to pre-training can improve the performance, since previous work has only verified the effectiveness of increas-



http://dx.doi.org/10.21437/Interspeech.2021-236

ing in-domain pre-training data [6]. In Section 4.3, robustness of a model is evaluated through testing on domains that are not seen during pre-training or fine-tuning. All of the experiments mentioned above are fine-tuned on labeled data of 10 hours for low-resource setups. In Section 4.4, a 100-hour labeled set is used for fine-tuning to validate if the conclusions from earlier sections still hold with more labeled data.

Careful ablation studies are conducted in the following two sections. Section 4.5 keeps the total amount of data used in pre-training constant and varies the domain similarity by mixing OOD and in-domain data with different ratios. Section 4.6 dives deeper into the question raised in Section 4.1 by showing how much test performance is improved with respect to the amount of in-domain pre-training data. In the last section, we scale up the experiments by using even more data for pre-training/fine-tuning and a larger wav2vec 2.0 model to compare with previous work.

Table 1: *Experiment overview. Each capitalized letter denotes one domain, and "(t)" is added whenever the size from that domain is of the interest for the experiments in that section.*

| Sec. | Pre-Train | Fine-Tune | Test |
|---|---|---|---|
| 4.1 | B vs. {B, A} | B/C | A |
| 4.2 | B vs. {B, C} | A/B/C | A |
| | A vs. {A, C} | A/B/C | A |
| 4.3 | A vs. {A, B} | A/B/C | D/E/F |
| 4.4 | B vs. {B, A} | B($t$)/C($t$) | A |
| 4.5 | B($t_1$) + A($t_2$), s.t. $t_1 + t_2 = const$ | A/B/C | A |
| 4.6 | B + A($t$) | A/B/C | A |
| 4.7 | bigger {A, B}, bigger model | A/B | A/C |

### 3.1. Domains and Datasets

We consider English datasets from six domains, where datasets are regarded as from same domain if they are collected with the same process. The six domains are: (1) **LibriSpeech** [26] (**LS**) and **Libri-light** [27] (**LL**), which contain about 960 and 60K hours of 16kHz crowd-sourced audiobook recordings, respectively, derived from the LibriVox project; (2) **TED-LIUM v3** [28] (**TD**), which contains 452 hours of TED conference audio in 16kHz; (3) **Switchboard** [29] (**SB**) and **Fisher** [30, 31] (collectively denoted as **SF**), which consist of about 300 and 2K hours telephone conversational speech recorded in 8KHz, respectively; (4) **Common Voice** [32] (**CV**) with almost 700 hours crowd-sourced recordings of Wikipedia sentences in 48kHz; (5) **Wall Street Journal** (**WS**) [33, 34] with over 80 hours of read news text recordings, and (6) **VoxPopuli** (**VP**) [35], which contains 552 hours 16kHz speech recordings sourced from European Parliament plenary sessions.

Standard train/dev/test splits are considered for these datasets. Since SB/SF do not have a standard split, we follow the setup of [8] that uses RT-03S [36] for validation, and Hub05 Eval2000 [37] SwitchBoard (H-SB) and CallHome (H-CH) subsets for testing. The chosen datasets cover a wide variety of linguistic and acoustic domains, enabling comprehensive studies for various domain shift scenarios. We re-sample all datasets to 16kHz for consistency. Transcripts are pre-processed following [8], which upper-cases letters and removes punctuation except for apostrophes, resulting in 27+1(space) symbols.

### 3.2. Pre-training

For experiments from Section 4.1 to Section 4.6, we consider pre-training on various combinations of LS, TD, and SF. The three largest datasets (LL, SF, CV) are combined for the scaling experiments in the last section. The wav2vec 2.0 BASE architecture [20] is used for all but the last section, which uses the LARGE model. BASE/LARGE contain 12/24 transformer blocks, each of which with 768/1024 input and output dimensions, 3,072/4,096 inner (FFN) dimension and 12/16 attention heads. For BASE, 10% dropout is applied to the quantize/Transformer input, and output after attention, activation, and FFN layer within each transformer block. Additionally, LayerDrop [38] is applied with $p = 5\%$. LayerNorm [39] is used at each convolution layer in the feature encoder for both BASE and LARGE models.

The same pre-training hyperparameters are used for all BASE models following [6] regardless what combinations of datasets they are trained on, except for the number of updates. Models are trained for 400K steps for the ablation studies in Section 4.5 and 4.6 for efficiency, and 800k steps elsewhere. In our preliminary studies, we found that the improvement is marginal beyond 800k.

### 3.3. Fine-tuning

We consider 10 hour subsets from LS, SF, and TD for supervised fine-tuning as the low-resource setup in most sections, 100 hour subset of TD as the mid-resource in Section 4.4, and full LS and SB in Section 4.7 for the high-resource setup. LS-10h is taken from the official Libri-light split, and other subsets are sampled from the corresponding training set with genders balanced.

Pre-trained models are fine-tuned with connetionist temporal classification [40]. The same fine-tuning hyperparameters are used for all pre-trained models when fine-tuned on the same labeled set. These parameters are tuned using the in-domain pre-training and fine-tuning setup. For example, TD-10h parameters are selected based on the TD-dev greedy decoding word error rate (WER) of a TD pre-trained model fine-tuned on TD-10h.

### 3.4. Language model and decoding

We decode each fine-tuned model using the word-level $n$-gram language model (LM) from the target domain built with KenLM [41] and report its WER. Wav2letter++ [42] beam search decoder are used with a beam size 50, beam threshold 100, and Bayesian optimization [43] is used to find decoding hyperparameters over 100 trials: LM weight ($[0, 8]$), word score ($[-5, 5]$), and silence score ($[-5, 5]$). We use the LMs provided in [8] for LS, SB, TD, CV, WS, and that from [35] for VP.

## 4. Results

### 4.1. Does adding in-domain pre-training data help?

Often times, it is much easier to obtain unlabeled speech for a particular domain than labeled data which requires annotation. Motivated by this, we examine the benefit of adding unlabeled in-domain data to pre-training. We pre-train models on all 7 possible combinations of LS, TD, SB, and fine-tune each of them on the ten hour subsets of the labeled version of each corpus, {LS,TD,SB}-10h, respectively. Validation WERs of these three domains are presented in Table 3. Unshaded columns are setups where the fine-tuning data are out-of-domain, and red numbers denotes models pre-trained with in-domain data.

For this section, we compare each black number with the red number on its right, where the red number is pre-trained additionally with the in-domain data. Both results are based on

Table 2: *Validation WER on domains unseen during pre-training or fine-tuning: WS, CV, and VP. {TD, LS, SB}-10h denote the labeled data used for fine-tuning for the corresponding column. We also report the average WER over the dev sets of WS, CV, VP, TD, LS, SB.*

| PT on X | *WS-dev WER* | | | *CV-dev WER* | | | *VP-dev WER* | | | *Avg WER (over 6 devs)* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TD-10h | LS-10h | SB-10h | TD-10h | LS-10h | SB-10h | TD-10h | LS-10h | SB-10h | TD-10h | LS-10h | SB-10h |
| TD | 11.32 | 9.87 | 11.10 | 36.93 | 34.53 | 46.04 | 18.59 | 16.67 | 22.41 | 22.65 | 20.34 | 22.85 |
| LS | 9.62 | 9.18 | 10.10 | 31.80 | 31.49 | 43.63 | 20.47 | 18.50 | 27.64 | 19.98 | 19.85 | 21.76 |
| SF | 14.65 | 12.79 | 99.25 | 44.14 | 43.08 | 94.53 | 22.88 | 24.73 | 99.97 | 23.60 | 22.51 | 83.09 |
| TD+LS | 9.08 | 8.00 | 8.95 | 28.54 | 27.34 | 37.59 | 14.77 | 14.43 | 17.77 | 17.25 | 16.21 | 17.63 |
| TD+SF | 10.64 | 9.83 | 10.01 | 32.98 | 32.02 | 36.09 | 16.19 | 16.69 | 17.25 | 17.69 | 16.90 | 17.90 |
| LS+SF | 9.76 | 8.72 | 9.32 | 28.67 | 28.29 | 34.12 | 15.49 | 16.18 | 19.60 | 15.86 | 15.06 | 16.97 |
| TD+LS+SF | 9.13 | 8.44 | 8.94 | 28.44 | 27.13 | 30.92 | 15.03 | 15.44 | 16.91 | 15.42 | 14.66 | 15.37 |

Table 3: *Validation WER on TD, LS, and SB of models pre-trained (PT) on various subsets of {TD, LS, SB}, and fine-tuned (FT) on TD-10h, LS-10h, or SB-10h.*

| X | *TED-LIUM (TD) dev WER* | | | | | |
|---|---|---|---|---|---|---|
| | FT on TD-10h | | FT on LS-10h | | FT on SB-10h | |
| | PT on X | X+TD | PT on X | X+TD | PT on X | X+TD |
| None | diverge | 9.93 | diverge | 10.99 | diverge | 11.32 |
| SF | 12.12 | 9.60 | 14.82 | 11.08 | 99.63 | 11.04 |
| LS | 9.81 | 8.59 | 12.92 | 8.91 | 13.08 | 10.39 |
| SF+LS | 9.13 | 8.91 | 10.61 | 9.67 | 12.25 | 10.75 |
| X | *LibriSpeech (LS) dev-other WER* | | | | | |
| | FT on TD-10h | | FT on LS-10h | | FT on SB-10h | |
| | PT on X | X+LS | PT on X | X+LS | PT on X | X+LS |
| None | diverge | 14.60 | diverge | 10.53 | diverge | 17.92 |
| SF | 28.91 | 14.30 | 20.36 | 10.44 | 94.38 | 15.53 |
| TD | 23.44 | 12.81 | 15.36 | 9.71 | 27.50 | 15.46 |
| SF+TD | 20.50 | 13.58 | 14.42 | 10.39 | 21.99 | 13.89 |
| X | *Switchboard (SB) RT03 WER* | | | | | |
| | FT on TD-10h | | FT on LS-10h | | FT on SB-10h | |
| | PT on X | X+SF | PT on X | X+SF | PT on X | X+SF |
| None | diverge | 18.90 | diverge | 19.30 | diverge | 10.80 |
| TD | 35.70 | 16.20 | 34.60 | 17.40 | 18.70 | 11.00 |
| LS | 33.60 | 17.80 | 36.50 | 16.10 | 18.20 | 11.00 |
| TD+LS | 29.70 | 17.40 | 28.90 | 16.90 | 15.60 | 10.80 |

fine-tuning and testing on the same data. The benefit of adding in-domain data is clearly shown as all the red numbers are lower than the black numbers in Table 3. Note that the model pre-trained on SF and fine-tuned on SB-10h fail catastrophically when tested on OOD data (TD and LS). We did not report *PT on None* as they all fail to converge when fine-tuned on 10h sets.

### 4.2. Does adding pre-training data help if out-of-domain?

Here we still pay attention to Table 3 but compare numbers vertically. We split the question to be answered into two scenarios: (1) the original pre-training data does not contain in-domain data, and (2) otherwise. For the former, we compare black numbers in the second and the third row (pre-trained on one OOD dataset) with those in the last row (on two OOD ones). The black numbers in the last row are consistently better than those above, confirming the benefit of adding OOD data in this case.

For the other scenario, we compare red numbers within each column, where we found the hypothesis holds most of the time when increasing pre-training data from one domain (row 1) to two domains (row 2 and 3), with the only exception being the SB RT03 WERs when fine-tuned on SB-10h. However, when further increasing from two to three domains (row 4), the results are mixed, where about half of the cases improve. In conclusion, adding OOD pre-training data does not always help.

Table 4: *Effect of more labeled data (LS dev-other WER).*

| X | *LibriSpeech (LS) dev-other WER* | | | |
|---|---|---|---|---|
| | FT on TD-100h | | FT on TD-10h | |
| | PT on X | X+LS | PT on X | X+LS |
| SF | 19.45 | 10.98 | 28.91 | 14.30 |
| TD | 18.35 | 9.84 | 23.44 | 12.81 |
| SF+TD | 15.30 | 10.57 | 20.50 | 13.58 |

### 4.3. Does pre-training on diverse data improve robustness?

We test the 21 fine-tuned models (7 pre-training dataset combinations $\times$ 3 fine-tuning datasets) from earlier sections on three domains not seen during pre-training or fine-tuning: Wall Street Journal (WS), Common Voice (CV), and VoxPopuli (VP), and report the results in Table 2. In general, by comparing numbers within each column, one can observe that a model pre-trained on more domains tends to perform better than those pre-trained on fewer. To derive a summary statistic, we report the average WER over the six domains (LS, SB, TD, CV, WS, VP) for each fine-tuned model in the last three columns of Table 2. It shows that pre-training on three domains achieves better performance than on two domains, which in turn is better than on one domain, regardless of what labeled data they are fine-tuned on.

### 4.4. Is it still effective and robust with more labeled data?

We fine-tune four models on a larger TD-100h labeled set and test on LS dev-other to verify if the conclusions from Section 4.1 and 4.2 hold when more labeled data are available. Results in Table 4 confirm that both adding in-domain pre-training data (black vs. red) and adding out-of-domain data (row 1 vs row 2) are effective except from TD+LS to SF+TD+LS.

### 4.5. Effect of pre-training data similarity to target domain

Section 4.1 shows that adding in-domain unlabeled data helps (red vs. black), but the improvement may be not just be due to domain similarity but it may also be due to simply increasing pre-training data size. To better understand the effect of domain similarity alone, we fix the amount of pre-training data to 450 hours and vary the ratio of TD/LS data to control domain similarity with respect to the test data, LS dev-other.

Table 6 shows performance improvements when increasing the amount of in-domain unlabeled data up to 50% of all pre-training data. If there is perfect domain match for the unlabeled data, labeled data and the target domain, then more unlabeled data leads consistently to better performance (grey shaded results). However, if the labeled data domain differs, then performance saturates at either 50% of in-domain unlabeled data for fine-tuning with TD-10h and 75% with SB-10h. The effect for this is particularly strong for TD-10h and we believe that it is beneficial to have some of the unlabeled data match the domain of the labeled data for fine-tuning.

Table 5: *Validation and test WER of* LARGE *models pre-trained on LV+SF+CV and fine-tuned on subsets of LS and SB. For comparison, $n$-gram decoding results from [8] are included (note test sets are donwsampled to 8kHz in [8]). SOTA results with supervised training on in-domain datasets are reported in the last row (these models are not tested on OOD data).*

| PT | FT | LibriSpeech | | | | Switchboard | | | CommonVoice | | TED-LIUM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | dev-c | dev-o | test-c | test-o | RT03 | H-SB | H-CH | valid | test | valid | test |
| LV+SF+CV | LS-10h | 2.78 | 5.78 | 3.17 | 6.26 | 17.1 | 13.1 | 19.2 | 17.30 | 21.06 | 7.11 | 7.65 |
| | LS | 1.77 | 3.84 | 2.08 | 4.15 | 13.0 | 9.8 | 14.6 | 12.36 | 15.27 | 5.06 | 5.38 |
| | SB-10h | 3.83 | 7.76 | 4.12 | 7.96 | 9.8 | 6.3 | 13.3 | 20.32 | 24.69 | 7.01 | 6.82 |
| | SB | 3.03 | 6.74 | 3.30 | 7.08 | 7.7 | 4.9 | 9.9 | 18.02 | 22.48 | 5.62 | 5.18 |
| None [8] | LS | 2.0 | 5.3 | 2.5 | 5.6 | 27.5 | 19.3 | 26.4 | 18.8 | 22.5 | 7.8 | 9.4 |
| | SF | 7.1 | 19.1 | 7.9 | 20.4 | 10.4 | 6.5 | 10.3 | 31.7 | 36.0 | 8.5 | 8.8 |
| | Joint (LS+SF+CV+TD+WS) | 2.0 | 5.2 | 2.5 | 5.6 | 9.8 | 5.9 | 9.5 | 10.5 | 12.6 | 5.4 | 5.7 |
| *Sup. SOTA: LS [44] / SB [45] / CV [8] / TD [46]* | | - | - | *1.9* | *3.9* | *11.4* | *6.3* | *13.3* | *10.7* | *13.6* | *5.1* | *5.6* |

Table 6: *LS dev-other validation WER with exactly 450h of pre-training data and varying domain similarity.*

| Pre-train Size | | *LS dev-other WER*, FT on | | |
|---|---|---|---|---|
| TD | LS | TD-10h | LS-10h | SB-10h |
| 450.0h | None | 23.06 | 16.08 | 28.72 |
| 337.5h | 112.5h | 15.97 | 12.61 | 20.83 |
| 225.0h | 225.0h | 14.46 | 11.52 | 18.27 |
| 112.5h | 337.5h | 14.15 | 11.00 | 18.83 |
| None | 450.0h | 15.74 | 10.75 | 18.86 |

Table 7: *Effect of in-domain pre-training size in terms of LS dev-other WER with joint (Joint) and continual (Cont.) training.*

| Pre-train Size | | Num. of Cont. PT Steps on LS | *LS dev-other WER* | | | |
|---|---|---|---|---|---|---|
| | | | FT on TD-10h | | FT on SB-10h | |
| TD | LS | | Joint | Cont. | Joint | Cont. |
| 450h | None | - | 23.06 | | 28.72 | |
| | 10m | 10k | 22.93 | 22.75 | 29.24 | 29.35 |
| | 1h | 25k | 22.09 | 22.25 | 28.89 | 28.77 |
| | 10h | 50k | 21.83 | 21.77 | 26.78 | 27.59 |
| | 100h | 100k | 18.80 | 18.82 | 23.55 | 24.18 |
| | 960h | 400k | 13.19 | 13.91 | 16.75 | 17.03 |

### 4.6. Effect of in-domain pre-training data size

In this section, we study the relation between the amount of in-domain data used during pre-training and performance. Two strategies are considered: (1) *joint training*, which pre-trains on TD + LS of size $T$ for 400k steps, and (2) *continual training*, which first pre-trains on TD for 400k steps, and then pre-trains only on unlabeled LS of size $T$ for the numbers of steps specified in Table 7. In practice, it is convenient to pre-train a model on a large dataset, and then adapt it to a new domain of interest by running additional pre-training steps on that domain.

Table 7 shows that more in-domain unlabeled data improves performance for both joint training and continual training, and both achieve similar performance. Compared to Table 6, using all in-domain unlabeled data still performs well when fine-tuned on TD-10h, because pre-training always includes all TD data which matches the domain of labeled data for fine-tuning.

### 4.7. Larger model, more pre-training and fine-tuning data

Finally, we pre-train a single large wav2vec 2.0 model with 300M parameters [6] on three domains (LV, SF and CV) for 800K steps and fine-tune it on 10 hours as well as all data of the LS and SB datasets. We evaluate each model on the validation and test splits for LS, SB, CV and TD, showing in-domain and OOD performance. We compare our model to supervised models trained on single or multiple domains reported in [8] with 270M parameters (a deeper but narrower transformer model) (Table 5).

On in-domain data (LS/SB), our model achieves superior performance to all single- and multi-domain models in [8] except on the CallHome (H-CH) set. Pre-training on multiple domains is particularly effective when testing on domains different from fine-tuning: when fine-tuned on LS, WER is reduced by a relative 35% to 50% on SB, CV, TD compared to the single-dataset baseline trained on full LS in [8]. Moreover, we even achieve better OOD performance compared to that baseline when fine-tuning on just 10 hours of LS data (LS-10h). The same trend holds when comparing fine-tuning on SB-10h and for the baseline in [8] trained on all of SF (200x more labeled data). Our model is not pre-trained or fine-tuned on any TD data, and when fine-tuning it with LS, it achieves better performance on TD than the supervised SOTA trained on TD [46] as well as the joint RASR model trained on five labeled datasets including TD [8].

Finally, we quantify how much gain pre-training on in-domain unlabeled data achieves compared to the ideal setting with access to labeled in-domain data. We measure WERs on the SB test sets (H-SB/H-CH) for our large model fine-tuned on SB (ideal setup with in-domain labeled), contrast this to the same model fine-tuned on LS (with only in-domain unlabeled) and the competitive supervised model of [8] trained on labeled LS only (OOD labeled) as the *baseline*. Pre-training including in-domain unlabeled data closes 66%-73% of the gap between the baseline and the in-domain labeled setup. This bodes very well for practitioners who would like to build a model for a new domain since it is generally much easier to obtain unlabeled data for a new domain compared to transcribed data.

## 5. Conclusion

We present the first controlled study to better understand domain-shift in self-supervised learning for ASR. Results show that adding unlabeled in-domain data improves performance, even when the fine-tuning data does not match the test domain. On a large-scale competitive setup, we show that pre-training on unlabeled in-domain data reduces the gap between models trained on in-domain and out-of-domain labeled data by 66%-73%. Moreover, self-supervised representations trained on a variety of domains are robust and lead to better generalization performance on domains completely unseen during pre-training and fine-tuning. Retaining some unlabeled data from the same domain as the fine-tuning data is beneficial though.

## 6. References


[1] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv*, vol. abs/1807.03748, 2018.

[2] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech*, 2019.

[3] D. Harwath, W.-N. Hsu, and J. Glass, "Learning hierarchical discrete linguistic units from visually-grounded speech," in *Proc. ICLR*, 2020.

[4] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. R. Glass, "An unsupervised autoregressive model for speech representation learning," *arXiv*, vol. abs/1904.03240, 2019.

[5] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," *arXiv*, 2019.

[6] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020.

[7] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP*, 2013.

[8] T. Likhomanenko, Q. Xu, V. Pratap, P. Tomasello, J. Kahn, G. Avidov, R. Collobert, and G. Synnaeve, "Rethinking evaluation in asr: Are our models robust enough?" *arXiv*, 2020.

[9] R. M. Stern, N. Morgan, T. Virtanen, B. Raj, and R. Singh, "Features based on auditory physiology and perception." *Techniques for Noise Robustness in Automatic Speech Recognition*, vol. 193227, 2012.

[10] W.-N. Hsu and J. Glass, "Extracting domain invariant features by unsupervised learning for robust automatic speech recognition," in *Proc. ICASSP*, 2018.

[11] B. E. Kingsbury, N. Morgan, and S. Greenberg, "Robust speech recognition using the modulation spectrogram," *Speech communication*, vol. 25, no. 1-3, pp. 117–132, 1998.

[12] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," *arXiv preprint arXiv:1709.07902*, 2017.

[13] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani, "Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home," 2017.

[14] H. Tang, W.-N. Hsu, F. Grondin, and J. Glass, "A study of enhancement, augmentation, and autoencoder methods for domain adaptation in distant speech recognition," *arXiv preprint arXiv:1806.04841*, 2018.

[15] S. Khurana, N. Moritz, T. Hori, and J. L. Roux, "Unsupervised domain adaptation for speech recognition via uncertainty driven self-training," *arXi*, 2020.

[16] S. Sun, B. Zhang, L. Xie, and Y. Zhang, "An unsupervised deep domain adaptation approach for robust speech recognition," *Neurocomputing*, vol. 257, pp. 79–87, 2017.

[17] K. Kawakami, L. Wang, C. Dyer, P. Blunsom, and A. v. d. Oord, "Learning robust and multilingual speech representations," *arXiv*, 2020.

[18] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," *arXiv*, vol. abs/2002.02848, 2020.

[19] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv*, 2020.

[20] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *Proc. ICLR*, 2020.

[21] A. Baevski, M. Auli, and A. Mohamed, "Effectiveness of self-supervised pre-training for speech recognition," *arXiv*, vol. abs/1911.03912, 2019.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, and et al., "Attention is all you need," in *Proc. NIPS*, 2017.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv*, vol. abs/1810.04805, 2018.

[24] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. on PAMI*, 2011.

[25] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv*, vol. abs/1611.01144, 2016.

[26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. ICASSP*, 2015.

[27] J. Kahn and et al., "Libri-light: A benchmark for asr with limited or no supervision," in *Proc. ICASSP*, 2020.

[28] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *International Conference on Speech and Computer*, 2018.

[29] J. Godfrey and E. Holliman, "Switchboard-1 Release 2 LDC97S62," *Web Download. Linguistic Data Consortium, Philadelphia*, 1993.

[30] Cieri, Christopher, et al. , "Fisher English training speech parts 1 and 2 LDC200{4,5}S13," *Web Download. Linguistic Data Consortium, Philadelphia*, 2004,2005.

[31] ——, "Fisher English training speech parts 1 and 2 transcripts LDC200{4,5}T19." *Web Download. Linguistic Data Consortium, Philadelphia*, 2004,2005.

[32] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[33] J. Garofalo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) complete LDC93S6A," *Web Download. Linguistic Data Consortium, Philadelphia*, 1993.

[34] Linguistic Data Consortium, NIST Multimodal Information Group, "CSR-II (WSJ1) Complete LDC94S13A," *Web Download. Linguistic Data Consortium, Philadelphia*, 1994.

[35] C. Wang, M. Rivière, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," *arXiv preprint arXiv:2101.00390*, 2021.

[36] Fiscus, Jonathan G., et al., "2003 NIST Rich Transcription Evaluation Data LDC2007S10," *Web Download. Philadelphia: Linguistic Data Consortium*, 2007.

[37] Linguistic Data Consortium, "2000 HUB5 English Evaluation Speech LDC2002S09," *Web Download. Philadelphia: Linguistic Data Consortium*, 2007.

[38] A. Fan, E. Grave, and A. Joulin, "Reducing transformer depth on demand with structured dropout," in *Proc. ICLR*, 2020.

[39] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv*, 2016.

[40] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[41] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proc. WMT*, 2011.

[42] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert, "Wav2letter++: A fast open-source speech recognition system," in *Proc. ICASSP*, 2019.

[43] "Adaptive experimentation platform," https://github.com/facebook/Ax, accessed: 2020-03-27.

[44] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, and et al., "Conformer: Convolution-augmented transformer for speech recognition," *arXiv*, 2020.

[45] W. Wang, Y. Zhou, C. Xiong, and R. Socher, "An investigation of phone-based subword units for end-to-end speech recognition," *arXiv preprint arXiv:2004.04290*, 2020.

[46] W. Zhou, W. Michel, K. Irie, M. Kitza, R. Schlüter, and H. Ney, "The rwth asr system for ted-lium release 2: Improving hybrid hmm with specaugment," in *Proc. ICASSP*, 2020.