

LARGE-SCALE ASR DOMAIN ADAPTATION USING SELF- AND SEMI-SUPERVISED LEARNING

Dongseong Hwang, Ananya Misra, Zhouyuan Huo, Nikhil Siddhartha, Shefali Garg, David Qiu, Khe Chai Sim, Trevor Strohman, Françoise Beaufays, Yanzhang He

Google LLC, USA

ABSTRACT

Self- and semi-supervised learning methods have been actively investigated to reduce labeled training data or enhance model performance. However, these approaches mostly focus on in-domain performance for public datasets. In this study, we utilize the combination of self- and semi-supervised learning methods to solve unseen domain adaptation problems in a large-scale production setting for online ASR model. This approach demonstrates that using the source domain data with a small fraction of the target domain data (3%) can recover the performance gap compared to a full data baseline: 13.5% relative WER improvement for target domain data.

Index Terms— speech recognition, domain adaptation, self-supervised learning, semi-supervised learning, RNN-T

1. INTRODUCTION

It is natural to have poor speech recognition accuracy with small labeled dataset. There are 2 major approaches to resolving this problem: self-supervised learning (self-sup) and semi-supervised learning (semi-sup). Self-sup pre-trains the audio encoder to learn meaningful representations. Contrastive Predictive Coding (CPC) [1], Wav2vec [2] and Wav2vec2.0 [3] show that contrastive loss can be good pre-training objectives. Autoregressive Predictive Coding (APC) [4] shows that mean squared error (MSE) also can be a good objective. Semi-sup trains the ASR model using the pseudo labels generated by a teacher model. Noisy student training (NST) [5] is a popular semi-supervised learning approach, which also works for ASR [6, 7]. Recently, it is also shown that combining self- and semi-sup methods improves the performance of automatic speech recognition (ASR) [8] significantly, leading to the state-of-art (SOTA) Librispeech Word Error Rate (WER). However, the current methods focused on maximizing the performance for in-domain data. The largest dataset of the studies is labeled Librispeech data (1k hours) [9] and unlabeled LibriLight data (60k hours) [10].

In large-scale production ASR systems, there is often the challenge of mismatched domain between the training data (source domain) and the real world data (target domain) [11]. It is also common to have an order of magnitude bigger

dataset with various contexts (e.g. command, chat, caption) than the public dataset like Librispeech. To the best of our knowledge, there are no extensive studies to tackle large-scale domain adaptation using both self- and semi-supervised learning.

In this paper, we propose a combined self- and semi-sup approach for domain adaptation. Our method completely recovers target domain accuracy using only a small fraction of labeled target data. In addition, the improved generalization power enhances source domain accuracy as well. The other main contributions of our work is that we analyze how much self- and semi-sup contribute on domain adaptation. Self-supervised learning improves overall model generalization and semi-sup directly closes out-of-domain generalization gap, which means both are complementary.

2. RELATED WORKS

We use the three popular self-supervised learning methods in this work: Wav2vec [1, 2], and Wav2vec2.0 [3], APC [4].

The Wav2vec loss maximizes the mutual information between the context latent vector and the future inputs [2]. Instead of the convolution layers, we use conformer blocks [12] for the context network. As we directly use the log-mel features instead of the waveform, we exclude the feature network. The output of the audio encoder directly predicts the future log-mel features.

The Wav2vec2.0 loss maximizes the mutual information between the context latent vector and the masked input features [3]. We exclude the feature network for the same reason. We do not quantize the input features as we do not observe any performance difference in our experiments.

The APC loss minimizes the MSE between the input log-mel feature and the predicted log-mel feature by the audio encoder [4]. APC paper uses LSTM or transformer blocks but we use conformer blocks.

Noisy student training (NST) [6] is one of the most popular semi-sup methods. It trains a model with both labeled data and unlabeled data. The teacher model produces pseudo labels of unlabeled data from non-augmented input feature. The method trains the student model with augmented input feature and the pseudo labels as the ground truth. The current

Librispeech SOTA paper [8] uses NST with Librispeech data (labeled data) and LibriLight data (unlabeled data). Unlike image domain, it's hard to use soft label for sequence model. NST in speech domain [6, 8] uses hard label.

3. METHODS

3.1. Self-Supervised learning

All of the self-supervised methods are used to pre-train the audio encoder of the RNN-T model [13] using all the source and target domain data. This is followed by supervised training of the entire model using only the labeled source domain data. However, with this 2-stage training approach, it is difficult to determine the optimal pre-training hyper parameters and checkpoints because the frame accuracy metrics in self-sup are not always a good indicator of the final WER performance. To resolve this issue, we propose a joint train for both RNN-T and self-sup loss:

$$\mathcal{L} = \mathcal{L}_{\text{RNN-T}} + \lambda \mathcal{L}_{\text{self-sup}} \quad (1)$$

where λ is a hyper-parameter. We find $\lambda = 0.5$ works well.

Wav2vec2.0 is a bi-directional method. When we train it with a causal audio encoder, it does not converge. Our workaround is to pre-train with right context and subsequently fine-tune with left context only. Joint Wav2vec2.0 and RNN-T train does not have the issue. The joint train works on a causal audio encoder, because RNN-T loss can guide Wav2vec2.0 objective. As our RNN-T model is online, we use a causal audio encoder for all joint experiments. We add an additional MLP layer between the audio encoder and RNN-T joint layer, and between the audio encoder and self-supervised loss. The representation of the last audio encoder layer is specialized by 2 MLP layers for RNN-T loss and self-supervised loss respectively.

3.2. Semi-supervised learning

After self-supervised pre-training, we train the ASR model using RNN-T loss with both source domain data (labeled data) and target domain data (unlabeled data) [14]. NST produces pseudo labels for target domain data.

The teacher model is trained with source domain data (same for the student model). As a result, the pseudo labels generated for the target domain data is error-prone, which is harmful for domain adaptation. When pseudo labels are generated, the teacher model filters out low confidence utterances by Confidence Estimation Module (CEM) [15]. When the teacher model is trained by RNN-T loss, we add CEM whose inputs are the audio encoding and the beam search labels from the RNN-T model. The CEM is trained to minimise the binary cross entropy between the estimated confidence p and the binary target sequence c . The target sequence c contains a 1 when the prediction word is correct and 0 otherwise.

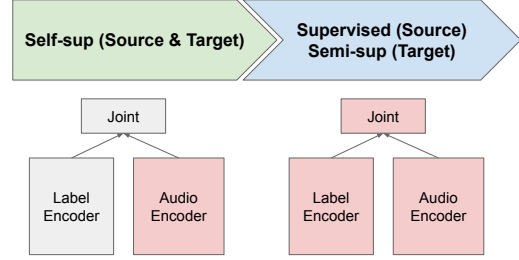


Fig. 1: Domain adaptation: Self-sup pre-trains the audio encoder. Supervised and semi-sup train all the modules.

The average word-level confidence is used to filter utterances. The teacher's multi-task training objective consists of both the RNN-T loss and confidence loss.

3.3. Domain adaptation approach

Figure 1 visualizes the proposed domain adaptation method. First of all, self-sup trains the audio encoder with source and target domain data. Then, RNN-T model is trained by RNN-T loss with source and target domain data. NST produces pseudo label for target domain data.

4. EXPERIMENTAL SETUP

4.1. Model Architecture Details

We use a Conformer RNN-T [13] model in the experiments. The audio encoder has 17 Conformer blocks with model dimension 512. As the model is online ASR, we restrict the model from using any future information [16]. Each conformer block [12, 16] uses causal convolution and left-context attention layers. The convolution kernel size is 15 and the self-attention layer consists of 8 heads with 65 left context length. The RNN-T decoder consists of a label encoder with 2 LSTM layers with 2048 units projected down to 640 output units, and a joint network with a single feed-forward layer with 640 units, as shown in Fig 1. The architecture and training procedure is similar to the LibriSpeech SOTA WERs model [8]. The total number of weights is 137M, while that of the SOTA model is 1B.

The model input is a vector of size 528, consisting of 4 contiguous frames of 128-dimension log-mel features [17] sub-sampled by a factor of 3 and one-hot domain-id vector of size 16. The model label uses 4,096 word pieces as sub-word label.

4.2. Data Sets

We use large multi-domain (MD) data sets [18] in English. MD utterances include multi domain data such as search, farfield, telephony and YouTube. All datasets are anonymized and hand-transcribed except for YT which uses YouTube video transcription with confidence filter. We divide MD data into

Table 1: Overview of training data sets. *MF* denotes the target domain utterances. *MD* denotes all domain utterances including *MF*, *SF* and *YT*. *MD_{src}* (*MD*−*MF*) denotes the source domain utterances.

Data set	Domain	Hours
Medium-form (MF)	Target	26k
Short-form (SF)	Source	27k
Youtube (YT)	Source	226k
Multi-domain (MD)	Source & Target	400k
MD−MF (MD _{src})	Source	374k
3% MF + MD _{src} (MD _{3p})	Source & Target	375k

source domain data and target domain data. We pre-train the model using source domain data by self-sup and transfer it to target domain data by semi-sup. As shown in Table 1, we use Medium-form (MF) utterances as target domain and MD−MF as the source domain. MF is average 10.4 secs length in natural conversation domain. Short-form (SF) is average 4.6 secs length in voice command domain.

For evaluation metrics, we calculate the WER of MF to measure the performance on target domain and the WER of SF for the performance on the source domain. The goal of domain adaptation is to minimize the amount of required transcriptions of MF while maintaining WERs for both MF and SF. In the final experiments, we use 3% MF as labeled data, because 800 hours data is manageable amount for hand-transcription. NST produces pseudo labels for rest of 97% MF data.

5. EXPERIMENTAL RESULTS

In this section, we conduct extensive domain adaptation experiments using self- and semi-supervised learning.

5.1. Self-supervised Learning

In Table 2, we compare the WERs of supervised and self-supervised learning. The first block lists supervised experiments with MD, MD_{src}, and MD_{3p} data. MD has full MF data. MD_{src} doesn't have any MF data. MD_{3p} has 3% MF data. As expected, SF WERs are the same but MF WERs are better when more MF data is used.

The second block lists three different self-sup experiments. One of 3 self-sup methods pre-trains the audio encoder with MD, and RNN-T loss trains the RNN-T model with MD_{src}. Wav2vec and APC have better WERs on both MF and SF than Wav2vec2.0, unlike what Wav2vec2.0 paper reported [3]. The downstream ASR model is online RNN-T, which is a causal model. Wav2vec and APC are causal models like GPT-3, but Wav2vec2.0 is full context (non-causal) model like BERT. It shows causal self-sup has better performance for causal downstream task. Even though Wav2vec

Table 2: Comparisons of Word Error Rate (WER) between supervised baselines, three self-sup algorithms.

Algorithms	Data	Word Error Rate (%)	
		MF (Target)	SF (Source)
Supervised	MD	3.2	6.0
Supervised	MD _{src}	6.2	6.0
Supervised	MD _{3p}	3.7	6.0
Wav2vec	MD _{src}	4.5	5.8
Wav2vec2.0	MD _{src}	4.6	5.9
APC	MD _{src}	4.5	5.8
Wav2vec	MD	3.2	5.8
Wav2vec	MD _{3p}	3.6	5.8

and APC have the same WERs, we use Wav2vec for rest of experiments. In our experience, APC is more sensitive to checkpoint fluctuations. When we choose a pre-trained checkpoint, Wav2vec works between 50k and 1.2M steps, but APC works only near 100k steps. In addition, APC requires total variation auxiliary loss to stabilize it [19].

In the third block of Table 2, Wav2vec MD_{3p} enhances WERs for both source domain (SF, 6.0 to 5.8) and target domain (MF, 3.7 to 3.6), compared to Supervised MD_{3p}. However, there is huge gap between Supervised MD and Wav2vec MD_{3p}. Self-sup enhances overall model generalization, but cannot reduce gap of out-of-domain (OOD) generalization.

5.2. Semi-Supervised Learning

The first block in Table 3 shows semi-sup results. Semi-sup MD_{src} denotes RNN-T training using MD_{src} for labeled data and MF for unlabeled data. Semi-sup MD_{3p} denotes RNN-T training using MD_{3p} for labeled data and 97% MF for unlabeled data. NST produces the pseudo labels for unlabeled data. Compared to Supervised MD_{src}, Semi-sup MD_{src} improves MF (target domain) WER from 6.2 to 3.4. There is only 0.2 % gap between Supervised MD. Semi-sup can reduce most of gap of OOD generalization, even without target domain data. Semi-sup MD_{3p} closes the all the gap, whose MF WER is same to the full data baseline: 3.2%. All semi-supervised experiments have the same SF WER to the baseline as all of them uses same amount of source domain data.

The teacher model for semi-sup has almost the same architecture as the student model. One difference for the teacher model is that it is non-causal. The self-attention right context is 64 and convolution layer is not causal. The teacher model uses the same amount of data as the student model. The teacher model is trained with MD_{src} for Semi-sup MD_{src} and MD_{3p} for Semi-sup MD_{3p}.

5.3. Self + Semi-Supervised Learning

The second block in Table 3 shows self + semi-sup WERs. In Table 2, Supervised MD uses all the labels of both source

Table 3: WERs for semi-supervised learning.

Algorithms	Data	Word Error Rate (%)	
		MF (Target)	SF (Source)
Semi-sup	MD _{src}	3.4	6.0
Semi-sup	MD _{3p}	3.2	6.0
Self + Semi-sup	MD _{3p}	3.1	5.7

domain (MD_{src}) and target domain (MF) data. Wav2vec MD_{3p} has better SF (source domain) WER than the supervised baseline because self-sup improves overall generalization as mentioned in Section 5.1. Semi-sup MD_{3p} has the same MF (target domain) WER to the baseline because semi-sup resolves OOD generalization as mentioned in Section 5.2. Combined both self- and semi-sup are complementary. In Table 3, Self + Semi-sup MD_{3p} show even better WERs for both MF (target domain) and SF (source domain). We are actually surprised that that Self + Semi-sup with 3% target domain has better WERs than supervised learning with 100% target domain. Our domain adaptation method not only reduces all the OOD generalization gap but also improves source domain performance. Semi-sup plays a much more critical role to close the OOD gap.

5.4. Confidence filter for Semi-Supervised Learning

In Table 3, all semi-supervised experiments use a confidence filter [15] that filters out target domain (MF) data when the utterance-level confidence score is less than 0.9. This results in dropping 24% of MF utterances and improves the teacher WER on the target domain by 54%. Without confidence filter, both MF and SF WERs are slightly worse.

Using only high-confidence utterances results in higher quality pseudo labeling, which enhances MF (target domain) WER from 3.2 to 3.1. As the teacher is trained with MD_{3p}, it tends to be more confident to source-domain-like SF data. This results in additional SF (source domain) WER improvement from 5.8 to 5.7. It indicates that confidence score threshold should be tuned per fraction of target label data.

Table 4: WERs of combined self- and semi-sup with/without confidence filter on MD_{3p}.

Algorithms	Word Error Rate (%)	
	MF (Target)	SF (Source)
Self + Semi-sup w/o filter	3.2	5.8
Self + Semi-sup w filter	3.1	5.7

5.5. Joint RNNT and Self-Sup train

In Fig 2, Joint W2V2 (Green line) reaches WER 4.6 at 200k step, while W2V2 requires 900k steps for both pre-training and supervised learning. Total training steps is reduced

significantly. In addition, Joint W2V2 reaches WER 4.3, which W2V2 never reach. It's because Joint W2V2 utilizes Wav2vec2.0 loss on the causal encoder, so there is no transition gap from non-causal to causal unlike the Wav2vec2.0 experiment. It brings out the full potential of Wav2vec2.0 objective. However, Joint W2V (Purple line) requires the same steps to reach the converging point to W2V (Blue line), as both utilize a causal audio encoder.

In Table 5, Joint W2V + NST and Joint W2V2 + NST has additional semi-sup stage. They have the same MF (target domain) WER to Self + Semi-sup MD_{3p} in Table 3. However, their SF (source domain) WER is worse than Self + Semi-sup MD_{3p} (5.7%). We leave the weaker in-domain generalization power of Joint train for future study.

**Fig. 2: Comparison of WER convergence with and without Joint train for wav2vec and wav2vec2.0.****Table 5: WERs of Joint train with MD_{3p}.**

Algorithms	Word Error Rate (%)	
	MF (Target)	SF (Source)
Joint W2V	4.5	5.8
Joint W2V2	4.3	5.8
Joint W2V + NST	3.1	5.9
Joint W2V2 + NST	3.1	5.8

6. CONCLUSION

In this paper, we investigated a domain adaptation method by combining self- and semi-supervised learning. Self-sup improves overall generalization and semi-sup closes out-of-domain generalization gap. Both methods complement each other. Extensive experimental results demonstrate that the proposed methods using only 3% of the labeled target domain data obtain better WERs on both the target and source domains than the baseline.

7. REFERENCES

- [1] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [2] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised Pre-Training for Speech Recognition,” in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.
- [3] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12449–12460, 2020.
- [4] Y. Chung and J. Glass, “Generative pre-training for speech with autoregressive predictive coding,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3497–3501.
- [5] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, “Self-training with noisy student improves imagenet classification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10687–10698.
- [6] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, “Improved noisy student training for automatic speech recognition,” *Interspeech 2020*, Oct 2020.
- [7] A. Xiao, C. Fuegen, and A. Mohamed, “Contrastive semi-supervised learning for asr,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3870–3874.
- [8] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, “Pushing the limits of semi-supervised learning for automatic speech recognition,” *arXiv preprint arXiv:2010.10504*, 2020.
- [9] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
- [10] J. Kahn, M. Rivi re, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazar , J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, et al., “Libri-light: A benchmark for asr with limited or no supervision,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7669–7673.
- [11] P. Bell, J. Fainberg, O. Klejch, J. Li, S. Renals, and P. Swietojanski, “Adaptation algorithms for neural network-based speech recognition: An overview,” *IEEE Open Journal of Signal Processing*, vol. 2, pp. 33–66, 2021.
- [12] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented transformer for speech recognition,” in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, H. Meng, B. Xu, and T. F. Zheng, Eds. 2020, pp. 5036–5040, ISCA.
- [13] A. Graves, “Sequence transduction with recurrent neural networks,” in *International Conference on Machine Learning: Representation Learning Workshop*, 2012.
- [14] A. Misra, D. Hwang, Z. Huo, S. Garg, N. Siddhartha, A. Narayanan, and K. C. Sim, “A comparison of supervised and unsupervised pre-training of end-to-end models,” in *Interspeech*, 2021.
- [15] D. Qiu, Q. Li, Y. He, Y. Zhang, B. Li, L. Cao, R. Prabhavalkar, D. Bhatia, W. Li, K. Hu, et al., “Learning word-level confidence for subword end-to-end asr,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6393–6397.
- [16] B. Li, A. Gulati, J. Yu, T. N. Sainath, C.-C. Chiu, A. Narayanan, S.-Y. Chang, R. Pang, Y. He, J. Qin, et al., “A better and faster end-to-end model for streaming asr,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5634–5638.
- [17] A. Narayanan, R. Prabhavalkar, C.-C. Chiu, D. Rybach, T. N. Sainath, and T. Strohman, “Recognizing long-form speech using streaming end-to-end models,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 920–927.
- [18] A. Narayanan, A. Misra, K. C. Sim, G. Pundak, A. Tripathi, M. Elfeky, P. Haghani, T. Strohman, and M. Bacchiani, “Toward domain-invariant speech recognition via large scale training,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 441–447.
- [19] Z. Huo, D. Hwang, K. C. Sim, S. Garg, A. Misra, N. Siddhartha, T. Strohman, and F. Beaufays, “Incremental layer-wise self-supervised learning for efficient speech domain adaptation on device,” *arXiv preprint arXiv:2110.00155*, 2021.