DRAFT: A Novel Framework to Reduce Domain Shifting in Self-supervised Learning and Its Application to Children's ASR

INTERSPEECH 2022
Sep 18 - 22 • Incheon Korea

Ruchao Fan, Abeer Alwan

fanruchao@g.ucla.edu

Department of Electrical and Computer Engineering, University of California Los Angeles, USA

UCLA Samueli Electrical & Computer Engineering

Introduction

- One of the challenges for children's speech recognition is the lacking of large, publicly-available and annotated databases
- Self-supervised learning has shown its potential in improving low-resource tasks, e.g. children's ASR [25].
- However, there is a domain mismatch between the pretraining and finetuning data, causing a domain shifting of the pretrained models.

- **In this paper (novel contributions),**
  - We develop a domain responsible adaptation and finetuning (DRAFT) to reduce the aforementioned domain shifting problem with only the funetuning data.
  - Residual adapters (RAs) are inserted after each block of the backbone model to learn domain related information during an adaptation stage.
  - Backbone model is updated at the finetuning stage. RAs are updated at both the adaptation and finetuning stage.
  - The proposed method is universal to all self-supervised learning methods.

DRAFT Framework

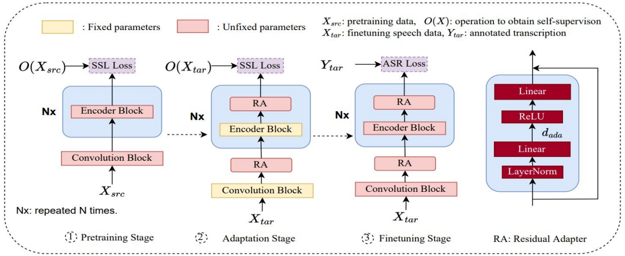

Figure 1: *An overview of DRAFT. d_{ada} is the output dimension of the first linear layer in the residual adapter (RA).*

- Pretraining stage: source domain data + SSL loss
- Adaptation stage: target domain data + SSL loss
- Finetuning stage: target domain data + ASR loss

- Notations:
  - θ_{ada}: parameters in residual adapters
  - θ_f : parameters in the backbone model (without residual adapters)
  - θ_g : parameters in the last embedding mapping layer for the self-supervised task
  - $\theta_{g'}$: parameters in the last linear layer for the ASR task
  - S_{src}: source domain data; S_{tgt}: target domain data
- SAFT: simple adaptation and finetuning
  - adaptation stage: update θ_f and θ_g directly using S_{tgt} and a SSL loss
  - Finetuning stage: update θ_f and $\theta_{g'}$ using S_{tgt} and an ASR loss
- DRAFT: domain responsible adaptation and finetuning
  - Initialize a model θ_f^0 and θ_g^0 update the parameters using S_{src} and a SSL loss, and obtain a pretrained model θ_f^1 and θ_g^1.
  - From model θ_f^1 θ_g^1, insert residual adapters after each block initialized with θ_{ada}^0, freeze and update θ_f^1 θ_g^1 using S_{tgt} and the same SSL loss, and obtain an adapted model θ_f^1 θ_g^1 θ_{ada}^1
  - From model { θ_f^1 θ_g^1 θ_{ada}^1}, replace θ_g^1 with a new generator that can map the embedding space to token space denoted as $\theta_{g'}^0$, update the entire model with S_{src} and an ASR loss, and obtain the final ASR model { θ_f^2 θ_{ada}^2 $\theta_{g'}^1$}.
- Since the backbone model is frozen during stage 2, catastrophic forgetting is prevented.

Results and Discussion

1. Experimental Setup

- Dataset
  - Pre-training task: Lirispeech adult dataset (960 hours)
  - Fine-tuning task: child speech datasets
    - OGI: 50 hours, scripted speech, 70:15:15 split for train, dev and test sets
    - MyST: 240 hours, spontaneous speech
- Training Details:
  - APC for causal transformers:
    - Input: 80-dim log-mel filter bank features
    - 12 transformer encoder blocks + a causal mask in self-attention
    - Output: 320-dim because of a 4x sub-sampling in the convolution block
    - Model size: ~ 39M
  - Wav2vec2.0 and Hubert for non-causal transformers:
    - We use pretrained models in the Fairseq toolkit [47].
    - Model size: ~ 95M
    - Code for this part: https://github.com/Diamondfan/fairseq

Acknowledgement

This paper was supported in part by the NSF and the UCLA-Amazon Science Hub!

References numbers are the same as those in the paper.

2. Effect of Adapter Size

Table 1. WER results of different values of dada in residual adapters using APC. SAFT: sample adaptation and finetuning. DRAFT: the proposed domain responsible adaptation and finetuning. The total number of updated parameters are also shown in absolute and relative values (compared to the SAFT). All DRAFT performance improvements are statistically significant.

	d_{ada}	OGI		MyST		Updated Params	
		dev	test	dev	test	total	relative
Baseline	0	5.9	7.0	36.7	36.3	-	-
Finetune	0	5.0	6.1	32.2	31.6	-	-
SAFT	0	5.0	5.9	33.4	32.9	39.2M	-
DRAFT	64	4.9	5.7	31.9	31.0	0.9M	2%
	128	4.7	5.6	31.6	30.9	1.7M	4%
	256	4.6	5.3	31.1	30.4	3.4M	9%
	512	4.4	5.2	30.9	30.2	6.8M	17%
	1024	4.4	4.9	30.1	29.4	13.7M	35%
	2048	4.4	4.9	30.0	29.3	27.3M	70%

- The WER drops when we increase the number of parameters in the RAs, while computational cost increases.
- One can use a small value of d_{ada} to achieve a fast adaptation of the self-supervised model when the computational resources are limited.
- A large value of d_{ada} can be used to achieve a better performance for the finetuning task.
- 1024 is used for all subsequent experiments.

3. Overall Results

Table 2. WER results of SAFT and DRAFT for APC, Wav2vec2.0 and HuBERT on the OGI and MyST datasets. NC: no convergence

	APC				Wav2vec2.0				HuBERT			
	OGI		MyST		OGI		MyST		OGI		MyST	
	dev	test	dev	test	dev	test	dev	test	dev	test	dev	test
Baseline (w/o SSL)	5.9	7.0	36.7	36.3	-	-	-	-	-	-	-	-
Finetune	5.0	6.1	32.2	31.6	2.27	2.70	17.84	17.16	2.07	2.48	17.40	16.71
Adapter Finetune [38]	8.6	10.1	47.4	47.3	NC	NC	NC	NC	NC	NC	NC	NC
Self-Transfer												
SAFT	5.0	5.9	33.4	32.9	2.22	2.67	17.85	17.28	2.02	2.43	17.52	16.89
DRAFT	**4.4**	**4.9**	**30.1**	**29.4**	**2.11**	**2.51**	**17.21**	**16.70**	**1.85**	**2.05**	**16.79**	**16.53**
Cross-Transfer												
SAFT	5.2	6.2	37.8	37.3	2.33	2.85	21.24	20.28	2.11	2.30	17.67	17.20
DRAFT	4.7	5.5	31.4	30.8	2.13	2.63	17.95	17.36	2.03	2.28	17.13	16.65

- Adapter finetuning [38] does not work well in our case.
- Compared to SAFT, DRAFT prevents overfitting during the adaptation stage,
- We achieve relative WER improvements of 19.7%/7.0%, 7.4%/2.7%, and 16.0%/1.1% on the OGI/MyST test sets for APC, Wav2vec2.0, and HuBERT, respectively.
- For cross-transfer experiments, we observe that the RAs learned from MyST data can help the finetuning on OGI data, while the RAs learned from OGI data did not provide improvements on MyST data.