DURF Project Report - Efficient Transfer Learning for Unsupervised Domain Adaptation

Yilun Kuang
New York University
yk2516@nyu.edu

1 Introduction

In recent years, pretrained language models [1][2][3][4] finetuned on downstream tasks have achieved state-of-the-art accuracy scores on a variety of benchmarks [5][6]. The pretrain-finetune paradigm has become the one of the de facto procedures for model training and inference.

Underlying the success lies the problem of robustness and out-of-distribution (OOD) generalization [7]. A model trained on a source distribution might fail during inference if the target distributions shift [8] [9], preventing real world safe AI deployments [7]. To address the OOD detection and generalization problem, a variety of approaches have been proposed, including the Invariant Risk Minimization (IRM) [10], softmax probability score detection [8], multi-task learning [11], and so on. This study proposes a new adapter-based method for unsupervised domain adaptation [12]. The results are comparable with standard baselines. The codebase for this project is available in Github [1].

2 Related Work

2.1 Out-of-Distribution Error for the Pretrain-Finetune Paradigm

For the pretrain-finetune paradigm, it has been shown that pretraining improves robustness to distribution shifts [13] [14]. Tu et al. [15] further shows that trained neural networks usually reply on spurious correlations in the in-distribution datasets and thus perform poorly on adversarial or challenging sets. Pretraining uses a few training examples without the spurious patterns and therefore causes the model to be more robust against challenging sets [15].

Finetuning, on the other hand, generally increases in-distribution performance but hurts out-of-distribution performance [16] [17]. Specifically, Andreassen et al. [16] defines the effective robustness (ER) of a neural network f as

$$\rho(f) = \text{acc}_{\text{out}}(f) - \beta(\text{acc}_{\text{in}}(f)),$$

where $\beta(\cdot)$ is a linear function that outputs the predicted out-of-distribution accuracy of a model given the in-distribution accuracy. This linear relationship assumption is based on the empirically shown linear fit between the in-distribution accuracy and the out-of-distribution accuracy, as analyzed theoretically in [18]. It is shown that pretrained models sometimes exhibit ER, and ER diminishes with the time step of finetuning [16]. Kumar et al. [17] claims that finetuning distorts pretrained features and provides a lower bound on the OOD error of finetuning (See Theorem 3.1 in [17]).

[1] https://github.com/YilunKuang/UDA_Experiments

2.2 Alternative Training Methods

Given the limitations of the pretrain-finetune paradigm in OOD generalization, it's natural to think of other transfer learning alternatives for downstream tasks based on the pretrained LM. This section reviews two of the parameter-efficient learning approach without changing the pretrained features.

2.2.1 Prompt Tuning

GPT-3 uses task demonstrations and discrete prompts to achieve state-of-the-art few shot learning performance [19] [20]. Li and Liang [21] and Lester et al. [22] develops soft prompt-tuning which keeps the pretrained feature frozen and optimizes for prepended continuous vector representations in the input text. Liu et al. [23] develops P-Tuning v2 optimizes continuous soft prompts at every pretrained layers. Gao et al. [19] uses prompt-tuning with demonstrations and automatic template generation for improved few-shot learning performance.

2.2.2 Adapter

Adapter [12] is proposed as a plugin-layer for the Transformer block as a parameter-efficient finetuning alternatives. Pretrained model weights are frozen and only the adapter layer is trained [12]. The AdapterFusion is also proposed to integrate an ensemble of adapters based on the attention mechanism with respect to the context [24].

Adapter is used to learn task representation in neural machine translation (NMT) tasks [25]. Pfeiffer et al. [26] proposes composing a language adapter trained via the Mask Language Modeling (MLM) objective on the target language and a task adapter trained on the classification objective on the source domain for transfer learning. Stickland et al. [27] studies several compositions of the domain adapter and the task adapter in the encoder and decoder layers of pretrained language models and obtained improved BLEU scores for translation performance.

2.3 Unsupervised Domain Adaptation

Domain adaptation is a special case of the OOD generalization problems involving shifts in the test distribution due to different domain [9]. It can be either supervised or unsupervised [9].

3 Method

This study is inspired by the line of work by Bapna et al. [25], Pfeiffer et al. [26], Pfeiffer et al. [24], and Stickland et al. [27] in using adapter for domain adaptation. Specifically, we consider the unsupervised domain adaptation settings where the pretrained LM has a source domain with labels and a target domain without labels. The goal is to use adapter to decouple the domain representation and the task representation to achieve better OOD generalization performance. The algorithm is given as follows

3.1 Algorithm

Let $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}, \mathcal{S} \subset P(\mathbb{R}^{n\times d} \times \mathcal{C})$ and $\mathcal{T} = \{x_i\}_{i=1}^{M}$, $\mathcal{T} \subset P(\mathbb{R}^{n\times d})$ be respectively the source domain and target domain training datasets. Here n is the length of the input sequence. $\mathcal{C}$ is the label space. Let $f : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}^n \times \mathbb{R}^h$ be the pretrained language model, where h is the dimension of the output hidden states. $\phi \circ f : \mathbb{R}^n \times \mathbb{R}^d \to [0, 1]^{|\mathcal{C}|}$ is the classification head on top of the pretrained LM.

First, we train the pretrained LM f on $\mathcal{S}$ with the Mask Language Model objective $\mathcal{L}_m$ for learning document representation. Then, the adapter module are attached to the model and trained using the cross entropy loss $\mathcal{L}_c$ with the pretrained weights frozen on $\mathcal{S}$. The third step is to train f on $\mathcal{T}_{\text{train}}$ with the MLM objective $\mathcal{L}_m$ to learn the target domain representation. Finally, we evaluate the model with adapters on the target domain dataset $\mathcal{T}_{\text{test}}$.

3.2 Experiment Setup

For pretrained LM, bert-base-uncased from the Huggingface implementation is used [2] [28]. The adapter implementation is from the AdapterHub [29]. The algorithm is then given in the algorithm block. Default configurations and an early stopping criteria are set up for model training.

Algorithm 1 Experiment Prototype

Require: Bert-base-uncased

1: Continue the pretraining of the BERT encoder on the source domain dataset using the Mask-Language-Modeling (MLM) objective (learn document representations)
2: Freeze BERT model parameters and finetune the adapter module on the source domain dataset using Cross Entropy (CE) loss (learn task representations)
3: Continue the pretraining of the BERT encoder on the target domain dataset using the Mask-Language-Modeling (MLM) objective (learn document representations)
4: Evaluate BERT with adapters on the target domain dataset

Both classification and natural language inference (NLI) tasks are used to evaluated the proposed method. The pair of in-distribution vs. out-of-distribution classification datasets used are IMDB-SST2, IMDB-Yelp, SST2-Yelp [30] [31] [32]. The pair of in-distribution NLI datasets used are SNLI-MNLI and MNLI-SNLI [33] [34].

Three baselines are used to compare against our method. They are given as follows

Algorithm 2 Baseline Design

Require: Bert-base-uncased

1: Baseline 1
2: Standard fine-tuning on source domain,
3: zero-shot test on target domain.
4: Baseline 2
5: Standard fine-tuning on source domain,
6: MLM fine-tuning on target domain,
7: then test on target domain.
8: Baseline 3 (Self Training)
9: Standard fine-tuning on source domain,
10: zeroshot test on target domain.
11: Use the confident labels as pseudo label
12: then standard finetune the model on the target domain with pseudo labels,
13: test on target domain

All of the algorithm and baseline results are averaged over three seeds. The experiments are done using the NVIDIA Tesla V100-SXM2 GPUs and NVIDIA Tesla Quadro RTX8000 GPUs on the NYU Greene Cluster.

4 Results

4.1 Classification Task

The results for the classification task is given as follows

Table 1: Evaluation Accuracy of the Adapter Method and the Baselines

METHOD	IMDB-SST2	IMDB-YELP	SST2-YELP
Adapter	0.86208	0.91267	0.90382
Baseline 1	0.86582	0.91348	0.87095
Baseline 2	0.53784	0.60673	0.50286
Baseline 3	0.85435	0.93021	0.903

Table 2: Evaluation Loss of the Adapter Method and the Baselines

METHOD	IMDB-SST2	IMDB-YELP	SST2-YELP
Adapter	0.34954	0.22094	0.24731
Baseline 1	0.71760	0.40717	0.51086
Baseline 2	0.67634	0.68292	0.69226
Baseline 3	1.12710	0.55087	0.56499

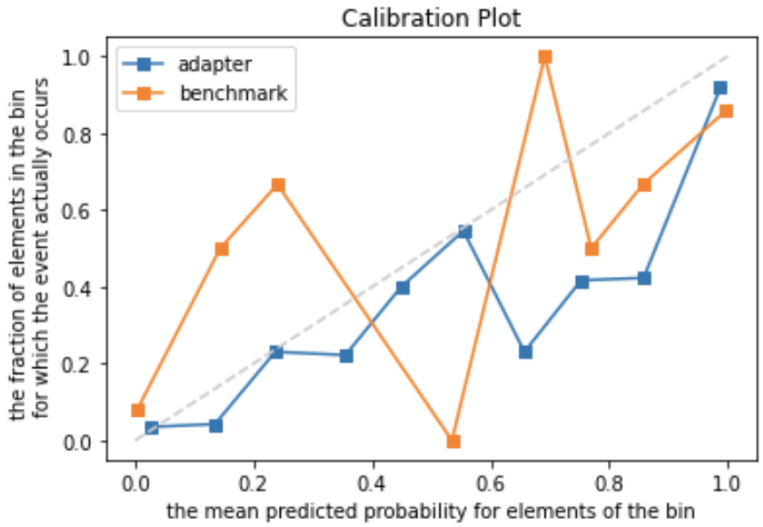
Table 3: Evaluation Perplexity of the Adapter Method and the Baselines

METHOD	IMDB-SST2	IMDB-YELP	SST2-YELP
Adapter	1.41860	1.24730	1.28073
Baseline 1	2.05054	1.50258	1.67353
Baseline 2	1.96667	1.97965	1.99823
Baseline 3	3.08671	1.73477	1.75943

The proposed adapter method has very close evaluation accuracy performance compared to baseline 1 and baseline 3. Baseline 2 does not have a good enough evaluation accuracy, even if it differs from baseline 1only by one additional MLM finetuning procedure. The adapter method does generally seems to have a lower evaluation loss and evaluation perplexity compared to all three other baselines.

4.2 Calibration Analysis

To further dissect the causes of the the lower evaluation loss, calibration analysis is performed for the IMDB-SST2 pair. The difference in calibration between the adapter method and baseline 1 is given in the figure below.



It looks like for the classification task, the adapter method is generally more calibrated in the low confidence region.

4.3 Natural Language Inference Task

To see if the results shown in the classification tasks are transferrable to other tasks, experiments on NLI tasks are given in the table below.

Table 4: Evaluation Accuracy of the Adapter Method and the Baselines

METHOD	SNLI-MNLI	MNLI-SNLI	MNLI-MNLI	SNLI-SNLI
Adapter	0.6166	0.6408	0.7787	0.8336
Baseline 1	0.63171	0.67780	0.75702	0.83011

Table 5: Evaluation Loss of the Adapter Method and the Baselines

METHOD	SNLI-MNLI	MNLI-SNLI	MNLI-MNLI	SNLI-SNLI
Adapter	0.924	0.8301	0.5574	0.4315
Baseline 1	0.91385	0.74415	0.60417	0.46036

Table 6: Evaluation Perplexity of the Adapter Method and the Baselines

METHOD	SNLI-MNLI	MNLI-SNLI	MNLI-MNLI	SNLI-SNLI
Adapter	2.5194	2.2936	1.7461	1.5396
Baseline 1	2.49392	2.10466	1.82974	1.58465

Compared to baseline 1, the adapter method does not have a consistent lower loss and perplexity values. Adapter outperforms the baseline in the in-domain settings but underpforms in the out of domain setting.

5 Discussion and Conclusion

The results for the adapter method are not ideal. For the classification task, it is shown that baseline 2 underperforms compared to baseline 1. Baseline 2 differs from baseline 1 by adding an additional MLM finetuning on the target domain. It's likely that the additional MLM perturbs the pretrained features and thus leads to catastrophic forgetting [35]. Based on this observation, it's likely that our adapter method based on the MLM objective also suffers from catastrophic forgetting due to additional MLM finetuning. This study also does not strictly follow the setup from Pfeiffer et al. [26] and Stickland et al. [27], where the domain adapter is trained using MLM to decouple domain and task information.

To prevent catastrophic forgetting or other issues, future study can decouple the task and domain representations using adapter only. The text generation setup can also be more accurate to the original formulation in Pfeiffer et al. [26] and Stickland et al. [27]'s work. A candidate setup is given as follows.

Algorithm 3 Text Generation Experiment Prototype

Require: BART-Large

1: train the encoder of the BART-Large Model using pretraining objective on CNN/Dailymail (for learning document representation in domain A).
2: attach adaptors for the decoder of BART-Large with all pretraining parameter fixed. Finetune the adaptor parameters with on CNN/Dailymail for summarization tasks (this is using adaptor to do summarization on domain A based on the document representation of domain A). For prompt-tuning, use prompt-tuning instead of adapters.
3: Now fine-tune the encoder of BART-Large Model on Wikihow using pretraining objective with adaptors parameters frozen (for learning document representation in domain B).
4: Then test on Wikihow with adaptors for summarization task (do summarization on domain B).

Overall, this study proposes a novel adapter-based method for unsupervised domain adaptation. For future work, it's also worth evaluating the OOD generalization performance of adapter, prompt tuning, and other training alternatives in a more comprehensive way.

6 Acknowledgements

This study is supported by NYU Dean's Undergraduate Rsearch Fund and is submitted as the project report to CSCI-UA 521 Undergraduate Research course at New York University. I would like to thank Professor He He for thoughtful advice and guidance along the way.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL https://arxiv.org/abs/1810.04805.

[3] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2020. URL https://arxiv.org/abs/2006.03654.

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019. URL https://arxiv.org/abs/1910.10683.

[5] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2018. URL https://arxiv.org/abs/1804.07461.

[6] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems, 2019. URL https://arxiv.org/abs/1905.00537.

[7] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016. URL https://arxiv.org/abs/1606.06565.

[8] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. 2016. doi: 10.48550/ARXIV.1610.02136. URL https://arxiv.org/abs/1610.02136.

[9] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey, 2021. URL https://arxiv.org/abs/2108.13624.

[10] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2019. URL https://arxiv.org/abs/1907.02893.

[11] Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017. URL https://arxiv.org/abs/1706.05098.

[12] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

[13] Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness, 2020. URL https://arxiv.org/abs/2004.06100.

[14] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2712–2721. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hendrycks19a.html.

[15] Lifu Tu, Garima Lalwani, Spandana Gella, and He He. An empirical study on robustness to spurious correlations using pre-trained language models, 2020. URL https://arxiv.org/abs/2007.06778.

[16] Anders Andreassen, Yasaman Bahri, Behnam Neyshabur, and Rebecca Roelofs. The evolution of out-of-distribution robustness throughout fine-tuning, 2021. URL https://arxiv.org/abs/2106.15831.

[17] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution, 2022. URL https://arxiv.org/abs/2202.10054.

[18] Horia Mania and Suvrit Sra. Why do classifier accuracies show linear trends under distribution shift?, 2020. URL https://arxiv.org/abs/2012.15483.

[19] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners, 2020. URL https://arxiv.org/abs/2012.15723.

[20] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners, 2020. URL https://arxiv.org/abs/2012.15723.

[21] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation, 2021. URL https://arxiv.org/abs/2101.00190.

[22] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021. URL https://arxiv.org/abs/2104.08691.

[23] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks, 2021. URL https://arxiv.org/abs/2110.07602.

[24] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. 2020. doi: 10.48550/ARXIV.2005.00247. URL https://arxiv.org/abs/2005.00247.

[25] Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. Simple, scalable adaptation for neural machine translation, 2019. URL https://arxiv.org/abs/1909.08478.

[26] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. Mad-x: An adapter-based framework for multi-task cross-lingual transfer, 2020. URL https://arxiv.org/abs/2005.00052.

[27] Asa Cooper Stickland, Alexandre Bérard, and Vassilina Nikoulina. Multilingual domain adaptation for nmt: Decoupling language and domain information with adapters, 2021. URL https://arxiv.org/abs/2110.09574.

[28] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2019. URL https://arxiv.org/abs/1910.03771.

[29] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers, 2020. URL https://arxiv.org/abs/2007.07779.

[30] Imdb movie reviews datset. 2011. URL https://ai.stanford.edu/~amaas/data/sentiment.

[31] Sst: Stanford sentiment treebank. 2013. URL https://nlp.stanford.edu/sentiment/treebank.html.

[32] Yelp open dataset. 2012. URL https://www.yelp.com/dataset.

[33] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2017. URL https://arxiv.org/abs/1704.05426.

[34] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference, 2015. URL https://arxiv.org/abs/1508.05326.

[35] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.