Incremental Neural Language Model Learning with Adapters

Aditya Kurniawan
Charles University, MFF UFAL
akurniawan.cs@gmail.com

Deniz Gunceler
Amazon Alexa
denizg@amazon.com

Anna Piunova
Amazon Alexa
piunova@amazon.com

Abstract

The objective of incremental learning is to adapt a pre-trained model towards a new domain with minimal degredation of its performance on past domains and with the smallest retraining cost. However, fine-tuning pre-trained models on new data leads to catastrophic forgetting where the newly adapted models' performance degrades on past data. One promising approach is to use adapter modules. In this work we conduct the first extensive study on catastrophic forgetting in an adapter module for neural language models (LMs). We show that adapters are more robust against catastrophic forgetting (as much as 20.8% better perplexity) and have better training speed (up to 43% more samples processed per second) in contrast to fine-tuning the whole model. We reduce the catastrophic forgetting further by incorporating past data during fine-tuning and show that the perplexity on past data improves by up to 79% with only 10% of past data included compared to training the adapter on target domain only. Additionally, we propose a new gating mechanism to automatically turn the adapters on/off depending on the input data, that reduces the perplexity on past data to be on-par with the model before adaptation. With this gating mechanism, we almost recover the performance on past data while sacrificing only a modest amount of perplexity (≈4%) on the new data.

1 Introduction

The ability to continually learn new knowledge is crucial for humans to understand new concepts. This is also important for machines faced with stream of data so that they can continuously learn and improve (Parisi et al., 2018). In a voice assistant application, frequent adaptation of the statistical language understanding system and it's components, such as an automatic speech recognition (ASR) model, is essential to keep up with trending topics. In this work we investigate improving a neural language model with previously unseen data, while limiting the degradation on past data.

A common adaptation approach in language modeling is fine-tuning all model parameters on the new domain data (Howard and Ruder, 2018; Radford and Narasimhan, 2018). While being efficient, this approach comes with challenges and limitations. First, adapting all model parameters is time consuming and computationally expensive, often incurring a cost comparable to training a new model from scratch (Biesialska et al., 2020; Devlin et al., 2019; Brown et al., 2020; Radford and Narasimhan, 2018; Radford et al., 2019). Second, fine tuning often results in catastrophic forgetting, where fine-tuned models forget what they have learned on the past data in order to improve on the new domain (McCloskey and Cohen, 1989; Yogatama et al., 2019). This is undesired if the purpose is to increase the model's knowledge coverage (Biesialska et al., 2020; Parisi et al., 2018; Delange et al., 2021).

Adapters are proposed as an alternative to standard fine-tuning (Houlsby et al., 2019; Bapna and Firat, 2019). This approach keeps the original network unchanged, and trains additional layers that are inserted into the network. By leveraging adapters, one trains a much smaller number of parameters compared to the standard fine-tuning, making the process computationally less expensive. It was also found that adapters are more robust against catastrophic forgetting than fine-tuning (Han et al.). Another interesting factor about adapters is that it has perfect memory of past data and past behavior can be recovered exactly by disabling the adapter. However, this holds only if we know the domain of the input data. In most practical applications the input data distribution is unknown, so turning the adapter off manually is not practical and a domain classifier is needed.

In this work, we perform an extensive study on

using adapters for incremental learning of neural language models efficiently and reducing the impact of catastrophic forgetting. Specifically, our contributions are the following:

- We demonstrate that we can increase training speed (up to 43%) compared to fine-tuning by using adapters.
- We show that adapters can achieve on-par or better performance on past data while also improving performance on new data.
- We show that adapters are more robust to catastrophic forgetting than standard fine-tuning (as much as 20.8%).
- We show that including past data during adaptation (as little as 10%) reduces catastrophic forgetting up to 79%.
- We introduce a novel end-to-end gating technique to control the usage of adapter module in the residual network. With this gating mechanism, we managed to reduce perplexity on the past data by 7.4 - 39.3% while also improving by 0.3 - 14.8% on the new data on top of the mixing strategy. The goal of the gating is to act as an automatic switch that decides whether to use adapter for a given input.

2 Related Work

Pre-trained language models such as the work of Devlin et al. (2019) has become the backbone of many state-of-the-art natural language processing (NLP) models. In transfer-learning, the language model is first trained on a large size corpus that covers broad domain and text varieties, and then fine-tuned to a specific application (Howard and Ruder, 2018).

Incremental learning refers to extending a model's knowledge by continuously training it on new data (Thrun, 1998). Unfortunately, straightforward fine-tuning often results in catastrophic forgetting (French, 1999), which causes the model performance to degrade on past data. The simplest way of avoiding forgetting is by freezing all parameters, using the pre-trained model as a feature extractor, and fine-tuning only the prediction layer. However, this leads to an inferior performance compared to fine-tuning all parameters (Rosenfeld and Tsotsos, 2018). There is recent work to mitigate forgetting by incorporating past data to fine-tuning (Zheng et al., 2021; Li et al., 2020) or by regularizing using techniques such as Elastic Weight Consolidation (Kirkpatrick et al., 2017).

Adapters were proposed by Houlsby et al. (2019) as an alternative to fine-tuning. It is often assumed that an adapter has a perfect memory as the original model's parameters are unchanged. Furthermore, adapters also improve the fine-tuning speed by training significantly fewer parameters than fine tuning. The most common setup is appending new adapter modules in between the layers of a pre-trained model. There are two different adapter placements as proposed by Bapna and Firat (2019) and Houlsby et al. (2019). The former leverages the adapters by incorporating them in two different parts of the sub-layers. The latter only appends the adapters on top of each layer with layer normalization added within the adapter architecture. As of this writing, there are no direct comparisons between these two techniques. However, the work of Bapna and Firat (2019) is simpler to implement and has been adopted in other works such as Pfeiffer et al. (2020); Rücklé et al. (2020); Pfeiffer et al. (2021)

There have been several applications of adapters in NLP as well as in ASR. Bapna and Firat (2019) show the benefit of using adapters when adapting machine translation models to new domains and languages; Houlsby et al. (2019) propose to use adapters in various natural language understanding (NLU) domains such as commonsense reasoning, sentiment analysis, and natural language inference (NLI) (Pfeiffer et al., 2021); Pfeiffer et al. (2020) show that adapters can be extended to work efficiently in a multi-language setup where they are incorporated with new languages in multi-lingual models such as Devlin et al. (2019) as their main model; and finally Lee et al. (2021) show that incorporating adapters in multi-domain language models can help to improve the Word Error Rate (WER) of an ASR model while increasing the number of parameters by only 2%.

Gating mechanism in adapters was studied by Rochan et al. (2021); Pham et al. (2020) as a regularization effect. Rochan et al. (2021) propose to learn a lightweight gating mechanism in Computer Vision domain on a residual network to allow the network to gradually learn to regulate the contributions from adapter transformations. Alternatively, Pham et al. (2020) incorporate a classifier that was trained separately to regularize adapter when the adaptation data is small. They investigate the ef-

fectiveness on a machine translation task and find that incorrectly choosing between the generic and adapted models can negatively impact the translation result.

Adapter efficiency was studied in Rücklé et al. (2020) by disabling adapters on lower transformer layers during training either statically or dynamically. The work investigates dropping the adapters on lower layers and sharing the parameters across all layers. The work shows that one can get speed up during inference by pruning the adapters from lower layers with minimal decrease in task performance.

3 Methodology

3.1 Base model architecture

We use the Transformer architecture (Vaswani et al., 2017) as our base model in the following configuration: We use the embedding dimension of size 512, 6 encoder layers with 8 attention heads and 2048 hidden units. The output layer dimensionality and the corresponding vocabulary size is 30K, that corresponds to the most likely subword segmentation, generated by the unigram wordpiece model (Kudo and Richardson, 2018). We omit the decoder component of transformer and only focus on the encoder component.

3.2 Adapter

3.2.1 Adapter Module

We closely follow the adapter implementation from Bapna and Firat (2019); Pfeiffer et al. (2020) as depicted in Figure 1. The adapter module is defined as a simple bottleneck architecture that is added on top of a transformer layer.

The adapter A_l at layer l consists of a down-projector D_l followed by a non-linear activation $ReLU$ (Agarap, 2018), and an up-projection layer U_l. Let h_l be the output of current transformer layer l. It is firstly passed through a layer normalization (Ba et al., 2016) LN prior to being transformed by a down-projection layer D_l of the adapter module. The output of up-projection is again forwarded through layer normalization:

$$A_l(h_l) = U_l(LN(ReLU(D_l(LN(h_l))))) + h_l$$

3.2.2 Adapter Efficiency

It's not immediately obvious what the optimal configuration of adapters is and different approaches have been proposed in the literature (Bapna and Firat, 2019; Houlsby et al., 2019; Pfeiffer et al., 2020; Rücklé et al., 2020). As part of our experiments, we evaluate four different adapter configurations to understand the most optimal choice:

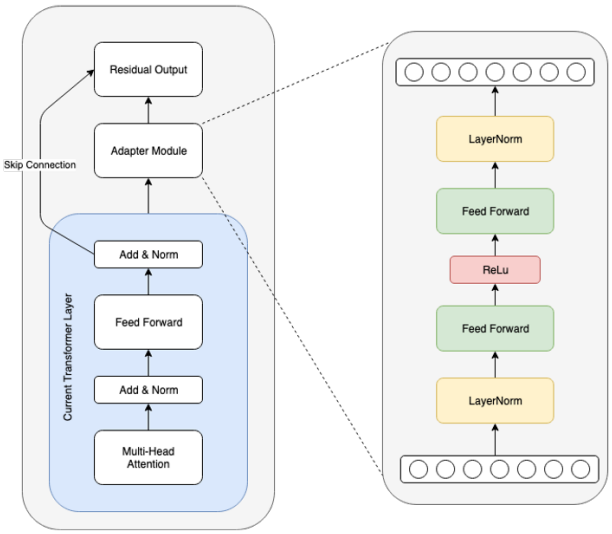


Figure 1: Adapter architecture

- Full adapters: We employ adapters on top of all transformers layers
- Upper-half adapters: We employ adapters only on the 3 top layers of transformers
- Bottom-half adapters: We employ adapters only on the first 3 layers of transformers
- Shared adapter: We employ only a single adapter that is shared across different transformer layers

3.2.3 Data mixing strategy

It is assumed that adapters should circumvent catastrophic forgetting as the original model's parameters are kept intact (Houlsby et al., 2019; Bapna and Firat, 2019). This assumption only holds when the input domain is defined prior to the prediction. We argue that this is not the case in a real-word scenario where the domain of input data is not always known in advance.

Zheng et al. (2021); Li et al. (2020) show that incorporating past data during the fine-tuning can help to mitigate the catastrophic forgetting. In this work, we investigate the effect of including different ratios of source and target domain data during adaptation. Similar to Zheng et al. (2021), we sample our data on-the-fly instead of directly combining both data sources from the beginning.

3.2.4 Gating mechanism

To automatically switch the adapter on/off depending on the input, we introduce gating mechanism on the residual connection between of the adapter. In contrast to Pham et al. (2020) that employs the gating mechanism at token level features, we operate on the sentence level features by leveraging average embedding of the current input tokens. We believe that providing the full sentence representation into the gating mechanism is helpful to classify the domain of the input, as the same token may qualify for different semantics and domain depending on the context. The gating mechanism G comprises of a single feed-forward layer L with $sigmoid$ as it's activation function. We average the current token embeddings emb_t as the input features of the feed-forward layer:

$$G(emb_{1..t}) = sigmoid(L(avg(emb_{1..t})))$$

4 Results and Discussion

4.1 Datasets

We conduct language model pre-training on Librispeech-LM corpus (Panayotov et al., 2015), that consists of texts from 14500 publicly available books. We split the dataset into 75M/380K/380K sentences for train, dev, and eval sets, respectively.

We use two different corpora for the fine-tuning experiments: PubMed (Canese and Weis, 2013) and Switchboard (Stolcke et al., 1998). PubMed consists of 31 million citations for biomedical literature from MEDLINE, life science journals, and online books. For this work, we only extract the abstracts from PubMed and do not leverage any of the metadata. Similar to LibriSpeech, we further split the data into 185M/2.2M/3.8M sentences-long partitions for train, dev, and eval sets. Switchboard comprises of 2,400 two-sided telephone conversations among 543 speakers. For language model training, we managed to populate around 250K sentences from the corpus. In the end, we generated 199K/25K/25K sentences-long splits for train, dev, and eval partitions.

4.2 Baseline results

Training only a small number of parameters such as adapters can achieve similar, if not better, performance than fine-tuning of the whole model according to Houlsby et al. (2019); Bapna and Firat (2019). To provide better clarity on the total number of trained parameters, we refer to Table 1. In it's full configuration the adapter only has ≈3% of the number of parameters that the pre-trained language model has (other configurations have even fewer parameters).

	Adapters Only	**Adapters + Embedding + Output Layer**
Trainable parameters	1,590,784	32,311,808
Total parameters	51,226,112	

Table 1: The total number of trainable and all parameters in different experimental setups.

In this study we pre-trained the model on LibriSpeech transcriptions and adapted on Switchboard or PubMed corpora. PubMed is a highly domain-specific corpus full of medical terms that were never seen during training on the LibriSpeech. Switchboard, on the other hand, is more similar to LibriSpeech in data distribution. This can be seen by looking at the perplexities in Table 2. The model pre-trained on LibriSpeech has a perplexity of 253 on Switchboard corpus, but an extremely high perplexity of 18896 on the PubMed corpus.

Switchboard The perplexity results for Switchboard are displayed in Table 2. In addition to the pre-trained and adapted models, we also train a model on the Switchboard from scratch as an "upper bound" on adaptation performance. We see that the adapted model even slightly outperforms the model trained only on Switchboard data (38.72 vs 40.23). This indicates that there is sufficient similarity between Switchboard and LibriSpeech for the pre-training on LibriSpeech to benefit the performance on Switchboard. However, this comes at a significant cost as the perplexity on LibriSpeech data is roughly four times the original value after adaptation (from 76.97 to 318.15).

PubMed The perplexity results for PubMed are also displayed in Table 2. We note again that the model trained just on LibriSpeech data from scratch performs poorly on the PubMed corpus, with a perplexity value of ≈18k. In contrast to the Switchboard, where we saw that adapting from LibriSpeech beats a model trained only on Switchboard, this does not happen for the PubMed corpus. We believe this is the result of the huge domain mismatch between LibriSpeech and Pubmed. However, the adapter can still recover most of the per-

Model	LibriSpeech (LS)	PubMed (PM)	Switchboard (SWB)
Adapter FULL trained on PM	1171.73	44.88	N/A
Full model fine-tuned on PM	1072.07	14.41	N/A
Adapter FULL trained on SWB	318.15	N/A	38.72
Full model fine-tuned on SWB	402.09	N/A	35.31
Baselines			
From Scratch trained on LS	76.97	18896.72	253.46
From Scratch trained on PM	1229.03	14.00	N/A
From Scratch trained on SWB	1195.86	N/A	40.23

Table 2: Perplexity results comparing adapted models to those trained from scracth on a particular domain. PM stands for PubMed, SWB and LS stand for LibriSpeech and Switchboard respectively.

Model	Perplexity	
	Pre-training Data	**Domain Data**
Switchboard		
Adapters	209.35	219.74
Full fine-tuning	258.72	78.83
PubMed		
Adapters	455.96	167.81
Full fine-tuning	996.80	78.75

Table 3: Perplexity results when first pre-training on Switchboard or PubMed corpus and then adapting to the LibriSpeech corpus.

formance gap, with perplexity improving to 44.88. As before, this improvement comes with a severe perplexity cost on past data, with perplexity on LibriSpeech degrading from 76.97 to 1171.73.

These results highlight that while adapters achieve good domain adaptation performance, catastrophic forgetting severely regresses performance on past domains. Mitigating this catastrophic forgetting will be the focus of section 4.5.

Fine-tuning Adapters + Embedding + Output layer We also investigate the impact of fine-tuning the embedding and output layers as well as the adapter as a possible mitigation strategy for the issues discussed earlier (Table 4). Fine-tuning these additional layers along with the adapter brings substantial improvement on the new domain data. The difference is more pronounced for PubMed, as we gain almost 52% relative improvement in perplexity when compared to the adapter-only model. In contrast to PubMed, we do not see the same benefit on past data for Switchboard. The perplexity even increases by ≈26% when we fine-tune the embedding and output layers. One reason for this could be the following: when the adaptation data is small (as is the case with Switchboard), the model easily overfits the much larger embedding and output layers, resulting in a performance degradation on past data. Adapter experiments in follow-up sections on PubMed train embedding and output layers along with adapters, while Switchboard adapter experiments only train the adapter modules.

4.3 The effect of the training sequence

For completeness, we also report results on the "reverse sequence", i.e. pre-training first on Switchboard or PubMed and then adapting to LibriSpeech (Table 3). Interestigly, the model adapted to LibriSpeech from Switchboard has a higher perplexity than the one that was adapted from PubMed; despite the fact that Switchboard corpus is closer in data distribution to LibriSpeech. This tells us that the size of the pre-training dataset influences significantly the adapter's capability (PubMed corpus is roughly 900 times larger than Switchboard).

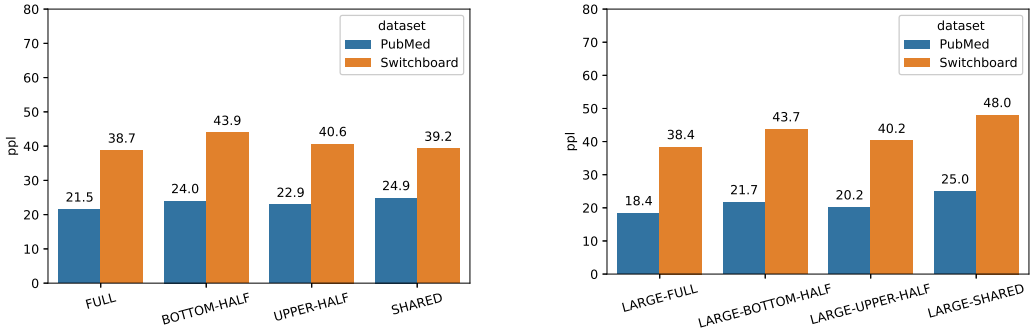
4.4 Performance of different adapters

Adapters configurations In this section, we discuss the experiments conducted to see the efficiency of the different adapter configurations. We can see from Figure 2 (a) that the performances of bottom-half, upper-half, and shared setups slightly degrade perplexity compared to the full configuration. We hypothesize that this is due to the fact that the full configuration has more trainable parameters. The difference is more pronounced on Switchboard than on PubMed. We also observe in Table 5, that compared to the full model fine-tuning, training adapters can be up to ≈43% faster, as fewer weights are trained.

Larger adapters In order to see the impact of having larger adapters, we experimented with a modification in the bottleneck architecture. Instead of scaling down the dimensionality, we scaled it up

Model	Adapters Only		Adapters +Embedding +Output Layer	
	LibriSpeech	Domain	LibriSpeech	Domain
Adapters fine-tuned on PubMed	1171.73	44.88	781.49	21.53
Adapters fine-tuned on Switchboard	318.15	38.72	402.05	35.15

Table 4: Perplexity results comparing models that train adapters only against models that train adapters as well as the embedding and output layers of the base model.



(a) (b)

Figure 2: Comparison of perplexities on different setup of adapters. (a) Shows the performance of the normal size adapter and (b) Shows the performance of the large size adapter

Methods	Average samples per second
Full fine-tuning	375
Full adapter	455
Upper-half adapter	535
Bottom-half adapter	490
Shared adapter	475

Table 5: Training speeds comparison between different type of adapter and full fine-tuning

to twice the original dimension size. We see in Figure 2 (b) that using larger adapters does not consistently improve perplexity, with results being mixed and varying with the configuration used. We see that increasing the adapter size benefits PubMed more than Switchboard (≈14% relative improvement over the standard adapter in full setup). We will also see in Section 4.5 that larger adapters suffer more from catastrophic forgetting, as having more parameters in the adapter encourages the model to overfit to the domain data.

4.5 Mitigating Catastrophic Forgetting

Our results so far on adapting neural language models are similar to and in agreement with what Bapna and Firat (2019) observed for machine translation models. We now turn our focus to controlling and mitigating catastrophic forgetting. For this purpose, we employ mixed training and a new gating mechanism to automatically turn the adapter on and off. For Switchboard dataset, we performed the experiments with all adapter configurations; but in the interests of frugality, we only performed PubMed experiments on the two best performing configurations.

4.5.1 Mixed training

Switchboard From Figure 3, we can see that introducing even small amounts of past data (as little as 10%) to adapter training can reduce the negative impact of catastrophic forgetting by more than 53.3% relative compared to training the adapter on domain data only. However, including too much past data results in modest gradual but consistent increase in the target domain perplexity.

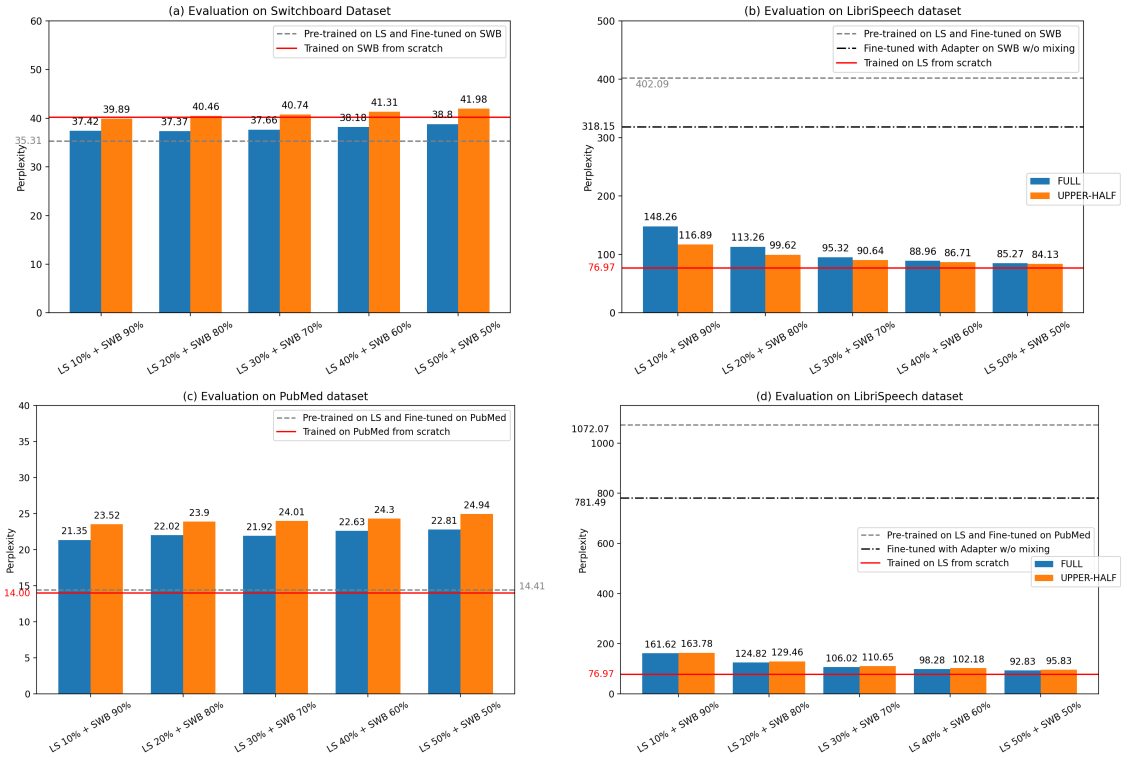


Figure 3: Perplexities of FULL and UPPER-HALF adapter configurations trained in data-mixing setup for models pre-trained on LibriSpeech. The first row (a, b) shows the performances of models adapted to Switchboard and the second (c, d) row shows the performance for models adapted to PubMed.

PubMed Similar to Switchboard, we see a significant ≈79% relative reduction in perplexity on LibriSpeech by incorporating past data, compared to the adapter that was only fine-tuned with domain data. Likewise, we also observe an increase in perplexity on the domain data when we add more past data.

4.5.2 Mixed training with gating

We now study the impact of the gating on the model performance. In Figure 4, we see that gating improves perplexity on both domain and past data (up to 29.1% on past data along with 11.9% on domain data for Switchboard). The perplexity improvements on domain data are largely insensitive to the data mixing ratio, but the perplexity improvements on past data are stronger in low mixing ratios.

In Table 6, we can see that after applying both data mixing and automatic gating, we're able to mitigate catastrophic forgetting almost completely. The model trained just on LibriSpeech has a pre-adaptation perplexity of 76.97 on the LibriSpeech evaluation set, while models adapted on PubMed and Switchboard have perplexities of 79.36 and 78.39 respectively (down from 1171.73 and 318.15 as shown in Table 2).

4.5.3 Comparison with fine-tuning

In this section, we compare adapters to fine-tuning all the weights of the base model on domain data. In Table 6 we see the trade-offs between improving on target domain data versus degradation on past data as a result of model adaptation via both adapter approach and whole model fine-tuning. On the Switchboard corpus, the performance of the two approaches on the domain data is almost on-par, but adapters experience smaller degradation on the original LibriSpeech data. However, on the PubMed corpus adapters are better at mitigating catastrophic forgetting while fine-tuning all weights achieves a lower perplexity on the domain data. These results suggest that using adapters is a better approach for incremental training on similarly distributed data (Switchboard scenario), whereas fine-tuning all weights can be more effective when adapting the model towards domain, whose data distribution is further away from the source data (PubMed scenario).

5 Conclusion

In this work, we present a study on using adapters to train neural language models, demonstrating that they are effective in mitigating catastrophic forget-

Mix		Perplexity			
Libri	Target	LibriSpeech		Domain Data	
		Fine-tuning all weights	Adapters	Fine-tuning all weights	Adapters
PubMed					
10	90	143.96	116.05	14.57	21.29
20	80	119.13	97.37	14.70	21.34
30	70	104.34	87.26	14.94	21.85
40	60	97.69	81.57	15.40	21.59
50	50	92.47	78.39	15.46	22.30
Switchboard					
10	90	126.70	114.82	31.90	33.43
20	80	107.85	92.42	31.38	33.06
30	70	97.71	84.23	31.79	33.13
40	60	93.32	81.36	32.04	33.65
50	50	89.88	79.36	32.37	33.81

Table 6: Perplexity results comparing fine-tuning all base model parameters and adapters with automatic gating.

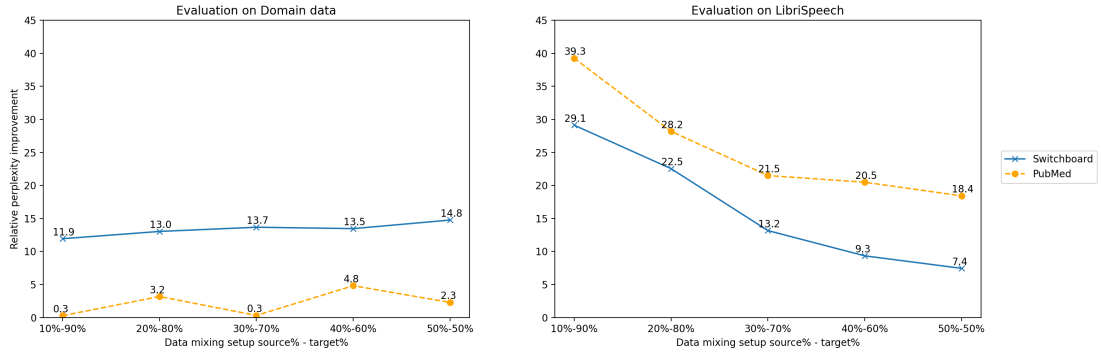


Figure 4: Relative perplexity improvements between adapters trained with gating mechanism and mixed data setup and adapters that were trained only with mixed setup. X-axes show the sample percentage in the data mixing strategy, 10%-90% means we include 10% of the source data and 90% of domain data. The left image shows the relative improvements on both Switchboard and Pubmed. The right image shows the adapters performance on the source LibriSpeech data after adaptation either on Switchboard or PubMed

ting in an incremental learning scenario. We first show that fine-tuning a pre-trained model using adapters increases the robustness to catastrophic forgetting by up to 20.8% compared to standard fine-tuning of the entire model.

Although the problem is not entirely solved just by fine-tuning the adapters, there are several approaches to help alleviate the problem further. By adopting a data mixing strategy, we show that incorporating a small amount of past data can help to reduce the perplexity up to 79%.

We also introduce a novel gating mechanism to automatically switch the adapters on and off depending on the input and train it in an end-to-end fashion. Even though we only use a very simple feature as input to the gating layer, this approach manages to reduce perplexity on the past data by 7.4-39.3% while also improving 0.3-14.8% on the domain data relative to the naive mixing strategy. Furthermore, when combined with data mixing, automatic gating can produce adapted models whose perplexity on past data is almost as good as the model before adaptation.

Furthermore, we also study the impact of different adapter setups. We find that we can reduce the number of trainable parameters by a factor of 6 by sharing adapter parameters across all layers while suffering only a modest absolute perplexity degradation. This could be a viable option in resource-constrained scenarios.

References

Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Kathi Canese and Sarah Weis. 2013. Pubmed: the bibliographic database. *The NCBI Handbook*, 2:1.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Wenjuan Han, Bo Pang, and Yingnian Wu. Robust transfer learning with pretrained language models through adapters.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Taewoo Lee, Min-Joong Lee, Tae Gyoon Kang, Seokyeoung Jung, Minseok Kwon, Yeona Hong, Jungin Lee, Kyoung-Gu Woo, Ho-Gyeong Kim, Jiseung Jeong, et al. 2021. Adaptable multi-domain language model for transformer asr. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7358–7362. IEEE.

Jinyu Li, Rui Zhao, Zhong Meng, Yanqing Liu, Wenning Wei, S. Parthasarathy, Vadim Mazalov, Z. Wang, Lei He, Sheng Zhao, and Y. Gong. 2020. Developing rnn-t models surpassing high-performance hybrid models with customization capability. *INTERSPEECH 2020*, abs/2007.15188.

M. McCloskey and N. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

G. I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and S. Wermter. 2018. Continual lifelong learning with neural networks: A review.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Minh Quang Pham, Josep M Crego, François Yvon, and Jean Senellart. 2020. A study of residual adapters for multi-domain neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 617–628.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Mrigank Rochan, Shubhra Aich, Eduardo R Corral-Soto, Amir Nabatchian, and Bingbing Liu. 2021. Unsupervised domain adaptation in lidar semantic segmentation with self-supervision and gated adapters. *arXiv preprint arXiv:2107.09783*.

Amir Rosenfeld and John K Tsotsos. 2018. Incremental learning through deep adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 42(3):651–663.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, N. Reimers, and Iryna Gurevych. 2020. Adapterdrop: On the efficiency of adapters in transformers. *ArXiv*, abs/2010.11918.

Andreas Stolcke, Elizabeth Shriberg, Rebecca Bates, Noah Coccaro, Daniel Jurafsky, Rachel Martin, Marie Meteer, Klaus Ries, Paul Taylor, Carol Van Ess-Dykema, et al. 1998. Dialog act modeling for conversational speech. In *AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, pages 98–105.

Sebastian Thrun. 1998. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome T. Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, A. Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and P. Blunsom. 2019. Learning and evaluating general linguistic intelligence. *ArXiv*, abs/1901.11373.

Xianrui Zheng, Yulan Liu, Deniz Gunceler, and Daniel Willett. 2021. Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems.