To Adapt or to Fine-tune: A Case Study on Abstractive Summarization

Zheng Zhao* **Pinzhen Chen**
School of Informatics, University of Edinburgh
{zheng.zhao, pinzhen.chen}@ed.ac.uk

Abstract

Recent advances in the field of abstractive summarization leverage pre-trained language models rather than train a model from scratch. However, such models are sluggish to train and accompanied by a massive overhead. Researchers have proposed a few lightweight alternatives such as smaller adapters to mitigate the drawbacks. Nonetheless, it remains uncertain whether using adapters benefits the task of summarization, in terms of improved efficiency without an unpleasant sacrifice in performance. In this work, we carry out multifaceted investigations on fine-tuning and adapters for summarization tasks with varying complexity: language, domain, and task transfer. In our experiments, fine-tuning a pre-trained language model generally attains a better performance than using adapters; the performance gap positively correlates with the amount of training data used. Notably, adapters exceed fine-tuning under extremely low-resource conditions. We further provide insights on multilinguality, model convergence, and robustness, hoping to shed light on the pragmatic choice of fine-tuning or adapters in abstractive summarization.

1 Introduction

In the current era of research, using large pre-trained language models (PLM) and fine-tuning these models on a downstream task yields dominating results in many tasks (Devlin et al., 2019; Lewis et al., 2020; Raffel et al., 2020; Brown et al., 2020). The scope of our work is on abstractive summarization, which is the task of generating a concise and relevant summary given a long document. Recent works have demonstrated the success of fine-tuning PLMs on summarization (Liu and Lapata, 2019; Zhang et al., 2020; Rothe et al., 2020). Nonetheless, such a paradigm becomes increasingly expensive with the ever-growing sizes of PLMs, since both the training time and space requirement increase along with the number of parameters. The issue becomes more severe when multiple languages or domains are introduced, as separate models need to be trained and saved depending on the setup.

Houlsby et al. (2019) proposed lightweight adapters as an alleviation of the large overhead of fine-tuning PLM on a downstream task. While many researchers have followed and adopted their idea, experiments are rarely done on summarization; from both quantitative and qualitative perspectives, it remains a myth of which direction one should pick in practice. In this work, we perform a thorough exploration of using adapters with a PLM on the task of abstractive summarization by examining different scenarios.

Our experiments are designed along three dimensions: 1) languages involved: monolingual, cross-lingual, and multilingual; 2) data availability: high, medium, low, and scarce; 3) knowledge being transferred: languages, domains as well as tasks. Through comprehensive experimental results, we demonstrate that with a realistic availability of resources, fine-tuning a PLM is superior to using adapters for the purpose of obtaining the best text quality. However, the game changes under low-resource settings: adapters have shown better, if not, on par performances compared to fine-tuning, especially in domain adaption.

*Corresponding author

Proceedings of the 21st China National Conference on Computational Linguistics, pages 824-835, Nanchang, China, October 14 - 16, 2022.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

2 Related Work

Fine-tuning a PLM with downstream task-specific objectives is a useful paradigm. It not only speeds up training, but also transfers the knowledge from abundant pre-training data to lower-resourced tasks. Whilst it has been proven successful in the field of summarization (Ladhak et al., 2020; Zhang et al., 2020; Zou et al., 2020; Rothe et al., 2020), this strategy requires optimizing and updating all parameters in the fine-tuned model, and is particularly expensive when a number of (sub-)tasks need to be approached.

To mitigate these problems, Houlsby et al. (2019) proposed to insert small neural modules named "adapters" to each layer of the PLM sequentially, and only update the adapters during fine-tuning while freezing most of the PLM parameters. When dealing with different sub-tasks – languages, domains, etc. – it is especially storage-efficient as only adapter weights need to be saved instead of the whole fine-tuned model. Several adapter architectures have been designed since then. Pfeiffer et al. (2020b) suggested simply placing adapters after the feed-forward block in each layer of the PLM, instead of adding adapters after both the multi-head attention and feed-forward block as proposed in the original work. Apart from adding adapters sequentially, He et al. (2022) designed an adapter that is parallel to the PLM.

Recent research that had utilized adapters in the task of summarization, argued that the low availability of opinion summarization datasets often leads to the standard fine-tuning method overfitting on tiny datasets (Brazinskas et al., 2022). Thus, they presented an efficient few-shot fine-tuning method based on adapters for opinion summarization. They added adapters to pre-trained models, trained the adapters on a large unlabelled customer reviews dataset, then fine-tuned them on the human-annotated corpus. Their method outperformed standard fine-tuning methods on various datasets. In addition, they showed that the proposed method can generate better-organized summaries with improved coherence and fewer redundancies in the case of summary personalization. Chen and Shuai (2021) created a meta-transfer learning framework for low-resource abstractive summarization, aiming to leverage pre-trained knowledge to improve the performance of the target corpus with limited examples. They inserted adapter modules into their model to perform meta-learning and leverage pre-trained knowledge simultaneously. Their methods are particularly effective under manually constructed low-resource settings on various summarization datasets with diverse writing styles and forms.

In comparison, our work investigates fine-tuning and using adapters in summarization, by comparing the performance of models using the fine-tuning strategy with models using adapters in the case of language adaptability, data availability, and knowledge transfer. For language adaptability, we examine the case of monolingual, cross-lingual, and multilingual summarization. For data availability, we study models trained under low, medium, and high resource scenarios. Lastly, for knowledge transfer, we investigate several factors: languages, domains, and tasks. To the best of our knowledge, adapters have not been tested in these scenarios.

3 Methodology

3.1 Method overview

Our aim is to study two fine-tuning variants for summarization under several settings using a PLM: the *fine-tuning* paradigm, and the *adapter* strategy. Fine-tuning initializes a PLM from a pre-trained checkpoint, then trains and updates the whole model on a summarization dataset. On the other hand, the adapter strategy also initializes a PLM from a pre-trained checkpoint, with adapter modules then inserted into the model. During training, we only update the adapter, the layer normalization parameters, and the final output layer.

We use mBART (Liu et al., 2020) as our backbone PLM for settings involving non-English languages. It is a sequence-to-sequence model pre-trained on large-scale monolingual corpora in 25 languages, with a denoising autoencoding objective. The model is designed to do multilingual machine translation tasks. After training it on a summarization dataset, the model is capable of doing monolingual, cross-lingual, and multilingual summarization. For English-only settings, we use BART (Lewis et al., 2020)

Proceedings of the 21st China National Conference on Computational Linguistics, pages 824-835, Nanchang, China, October 14 - 16, 2022.

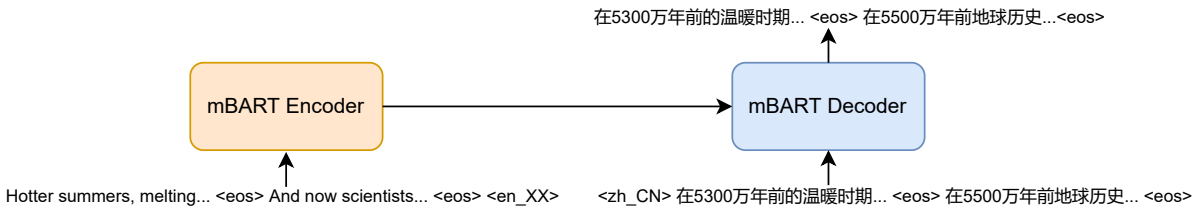


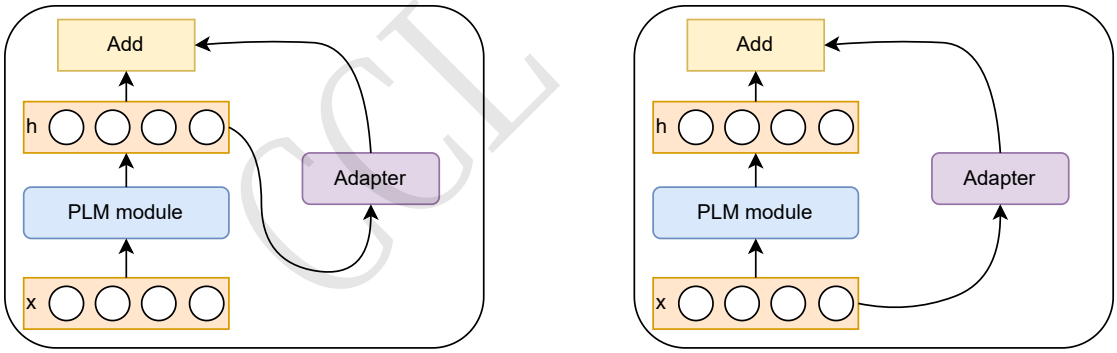
Figure 1: An illustration of our mBART based model for cross-lingual summarization from English to Chinese.

as the PLM. Similar to mBART, BART is also a sequence-to-sequence model pre-trained on large-scale corpora with denoising autoencoder architecture.

We have two kinds of models: mBART-FT which employs the fine-tune strategy, and mBART-Adapt which uses the adapt strategy. In order to recognise the source and target languages, following Liu et al. (2020), our models take a special separator token between each sentence, a language code token at the end of the source document, and at the beginning of the target summary. We provide a cross-lingual demonstration for our model in Figure 1. In addition, we propose BART-FT and BART-Adapt which use the fine-tune strategy and the adapt strategy, respectively.

3.2 Adapter variants

As mentioned earlier, there are various adapter variants. We experiment with two variants: one with sequential connections (Houlsby et al., 2019), and one with parallel connections (He et al., 2022). We display an illustration of these variants in Figure 2. After trying out different learning rates and reduction factors (the ratio between PLM's hidden dimension and adapter's bottleneck dimension), we discover that sequential adapters always outperform the parallel ones in our tasks. Thus we use Houlsby et al. (2019)'s sequential adapter for all of our mBART/BART-Adapt models.



(a) A sequential adapter

(b) A parallel adapter

Figure 2: An illustration of adapter variants, adapted from He et al. (2022). "PLM module" represents a certain sub-layer of the PLM (e.g. attention or feed-forward layer) that is frozen.

3.3 Evaluation

The evaluation metrics are F1 scores of ROUGE-1/2/L (Lin, 2004). Since we deal with multiple languages, we use the multilingual ROUGE implemented in a previous paper.[1] We stick to the toolkit's default settings, e.g., sentence segmentation and word stemming.

[1] https://github.com/csebuetnlp/xl-sum/tree/master/multilingual_rouge_scoring

Proceedings of the 21st China National Conference on Computational Linguistics, pages 824-835, Nanchang, China, October 14 - 16, 2022.

Dataset	Language	train / valid / test	Source
NCLS	zh→en	1.7m / 3.0k / 3.0k	Sina Weibo
	en→zh	365k / 3.0k / 3.0k	CNN/Daily Mail
Wiki-Lingua	en→ar	20.4k / 2.9k / 5.8k	wikiHow
	en→vi	13.7k / 2.0k / 3.9k	
	en↔ja	8.9k / 1.3k / 2.5k	
XL-Sum	gu	9.1k / 1.1k / 1.1k	BBC
	fr	8.7k / 1.1k / 1.1k	
	ne	5.8k / 0.7k / 0.7k	
	ko	4.4k / 0.6k / 0.6k	
	si	3.2k / 0.5k / 0.5k	

Table 1: Statistics of datasets and languages for the language adaption experiment.

4 Language Experiments

4.1 Experimental setup

We test our proposed paradigm on NCLS[2], WikiLingua[3], and XL-Sum[4] datasets particularly designed for cross-lingual and multilingual summarization (Zhu et al., 2019; Ladhak et al., 2020; Hasan et al., 2021). These datasets are either machine-translated or crawled from the web.

NLCS is built by machine-translating an existing English (en) dataset (CNN/Daily Mail, by Nallapati et al. (2016)) to Chinese (zh), and vice versa (Sina Weibo, by Hu et al. (2015)). A translated document is only kept if its round-trip translation reaches a certain threshold score. Plain translations and human-corrected translations are supplied as separate test sets; we use the human-corrected set in this work. WikiLingua is constructed by extracting and aligning article-summary pairs from wikiHow. We experiment with three languages that resemble medium and low-resource scenarios: Arabic (ar), Vietnamese (vi), and Japanese (ja).

Different from the cross-lingual datasets, XL-Sum is monolingual. It consists of professionally annotated article-summary pairs from BBC in many languages. The datasets come in various sizes for a number of languages, as shown in Table 1. This dataset allows for multilingual experiments since the data come from the same domain and are not centred on English. We experiment on five low-resource languages: Gujarati (gu), French (fr), Nepali (ne), Korean (ko), and Sinhala (si). For the monolingual scenario, we directly use the monolingual summarization data to train the model. For the cross-lingual setting, since machine translation is a cross-lingual task, we also directly train the model using the cross-lingual summarization data. Lastly, in a multilingual configuration, we simply mix summarization data in different languages, and train the model using the mixed data.

Our experiments are based on a public mBART checkpoint[5]. We use the adapter from Houlsby et al. (2019). Fine-tuning an mBART model updates around 610M parameters in total; the addition of adapters introduces 50M parameters, yet only this 8% are being optimized during training. We use the Adam optimizer for training (Kingma and Ba, 2015), with a learning rate of 1e-5 for mBART, and 1e-4 for mBART with adapters. We set the adapter reduction factor to 2, which means that the bottleneck dimension in an adapter is half of the hidden dimension in mBART. We perform hyperparameter searches on the following: learning rate and reduction factor, and monitor ROUGE scores on the validation set to select the best value. We provide further details of the grid search in Appendix A.

All models are trained on 4 NVIDIA A100 GPUs with a batch size of 12 on NCLS, and 4 on WikiLingua and XL-Sum. The model convergence time is from 1 to 30 hours depending on the dataset used. We use PyTorch (Paszke et al., 2019) for our model implementation. We use the Huggingface library (Wolf et al., 2020) and AdapterHub (Pfeiffer et al., 2020a) for mBART and adapter implementation.

[2] https://github.com/znlp/ncls-corpora
[3] https://github.com/esdurmus/wikilingua
[4] https://github.com/csebuetnlp/xl-sum
[5] https://huggingface.co/facebook/mbart-large-cc25

Proceedings of the 21st China National Conference on Computational Linguistics, pages 824-835, Nanchang, China, October 14 - 16, 2022.

Lang.	mBART-FT			mBART-Adapt		
	R1	R2	RL	R1	R2	RL
zh→en	**46.46**	**30.18**	**42.26**	41.41	22.73	36.56
en→zh	**45.22**	**22.49**	**34.38**	40.74	16.83	29.27

(a) High-resource, NCLS.

Lang.	mBART-FT			mBART-Adapt		
	R1	R2	RL	R1	R2	RL
en→ar	**25.85**	**7.35**	**21.01**	24.68	7.26	20.40
en→vi	**33.63**	**15.17**	**26.65**	30.98	13.94	24.59
en→ja	**35.70**	**12.34**	**28.34**	34.06	11.43	27.08
ja→en	**35.24**	**12.38**	**28.09**	33.14	11.54	26.46

(b) Medium and low-resource, WikiLingua.

Table 2: Results for cross-lingual summarization.

Lang.	Multilingual						Monolingual					
	mBART-FT			mBART-Adapt			mBART-FT			mBART-Adapt		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
gu	**20.18**	**6.96**	**18.09**	20.12	6.82	17.99	**20.23**	**6.43**	**17.67**	19.20	5.95	16.96
fr	**33.53**	**14.37**	**26.11**	33.44	14.01	25.63	**33.29**	**13.68**	**25.13**	32.37	13.02	24.73
ne	**24.70**	**9.52**	**22.23**	23.26	8.55	20.94	**24.06**	**9.05**	**21.62**	23.31	8.36	21.01
ko	17.73	**8.76**	16.27	**18.82**	8.12	**17.23**	**19.73**	9.12	**18.07**	19.05	**9.24**	17.73
si	**26.95**	**13.51**	**22.36**	25.68	12.69	21.80	**25.59**	12.25	**21.92**	24.99	**12.30**	21.44

Table 3: Results for low-resource multilingual and monolingual summarization on XL-Sum.

4.2 Results

We first provide results on high-recourse cross-lingual summarization on NCLS in Table 2a. We can see that mBART-FT achieves significantly higher ROUGE scores than mBART-Adapt in both Chinese-to-English as well as English-to-Chinese settings. We then list result numbers on medium and low-recourse cross-lingual summarization on WikiLingua in Table 2b. Similar to the behaviour under the high-resource setting, mBART-FT consistently achieves better ROUGE performance than mBART-Adapt, regardless of the source or target languages. However, we spot that the difference in ROUGE scores is smaller for language pairs with lower resources, which suggests a positive correlation between the gap in performance and training data availability.

In Table 3, we show results of both multilingual (left) and monolingual (right) summarization on XL-Sum. In a multilingual setup, a single model is trained on five languages. whereas in a monolingual setup, five individual models are trained on the five languages separately. We can first see that mBART-FT generally surpasses mBART-Adapt, in both multilingual and monolingual setups. In addition, multilingual models generally outperform monolingual models by a small margin. This behaviour is corroborated by Hasan et al. (2021)'s work that mixing multiple languages altogether during training can result in a positive transfer among them (Conneau et al., 2020).

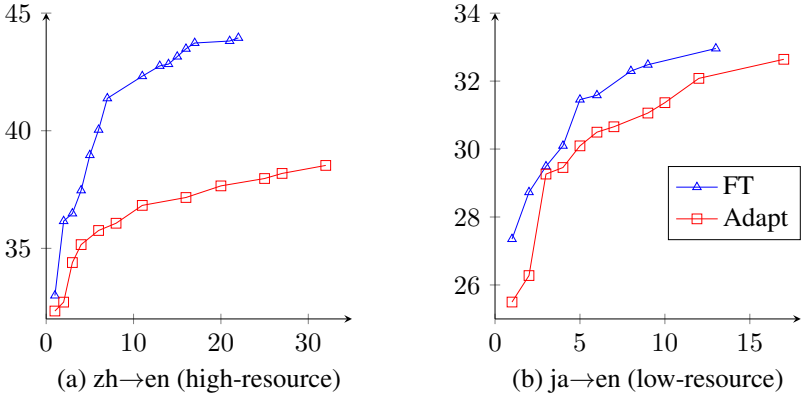
It is straightforward from our work, that, for summarization tasks with high data availability, it is not worth trading performance for efficiency with adapters. For low-resource scenarios, adapters achieve similar results as fine-tuning, and can therefore be a convenient choice for fast training and compact disk storage. When multiple low-resource languages are concerned, especially if they are related languages, it might be beneficial to build a multilingual model instead of individual monolingual models.

4.3 Convergence

To measure the convergence difference between mBART-FT and mBART-Adapt, we plot validation set ROUGE-1 scores against epochs for two previous experiments (high-resource zh→en and low-resource

Proceedings of the 21st China National Conference on Computational Linguistics, pages 824-835, Nanchang, China, October 14 - 16, 2022.

ja→en) in Figure 3. Plotting stops when validation does not improve. We measure convergence in terms of epochs, rather than wall-time. In our experiments, we find that wall-time per epoch for mBART-FT is about merely 1.5 times that for mBART-Adapt, since validation takes a large portion especially when the dataset is small.



(a) zh→en (high-resource) (b) ja→en (low-resource)

Figure 3: Validation ROUGE-1 (y-axis) against epochs (x-axis) for mBART-FT and mBART-Adapt in different data conditions.

As Figure 3(a) shows, with sufficient resources, mBART-FT and mBART-Adapt started with similar ROUGE scores, then the gap quickly increases, suggesting a faster and better convergence rate for fine-tuning. We also observe that mBART-FT converged within fewer epochs. Furthermore, Figure 3(b) suggests that, in a low-resource condition, even though mBART-FT surpasses mBART-Adapt in terms of ROUGE, they both have similar convergence rates with the gap reduced. These trends indicate that in a high-resource scenario fine-tuning is preferred, whereas in a low-resource scenario, adapters can be used to reduce overhead while maintaining performance.

5 Domain Adaptation Experiments

5.1 Experimental setup

In addition to multilinguality, we conduct extra experiments on domain adaptation, which is typically tackled using the same pre-training then fine-tuning paradigm. In our setting, we adapt CNN/Daily Mail to XL-Sum, both in English, with various data sizes. Although both datasets are news articles, they differ hugely in writing styles. We start with a BART model (Lewis et al., 2020) fine-tuned on the CNN/Daily Mail (Nallapati et al., 2016) dataset for summarization; it is available as a public model checkpoint.[6]

To further understand the impact of data availability, we artificially and iteratively make the training data 10 times smaller. This results in five data conditions with sizes ranging from merely 31 to 306.5k. We make sure that larger training splits are supersets of the smaller splits. The validation and test sets remain unchanged at 11.5k as provided in the original dataset. In addition to the XL-Sum dataset, which is in the news domain, we also experimented with adapting CNN/Daily Mail to the BookSum[7] (Kryscinski et al., 2021) dataset, a collection of narratives from the literature domain such as novels, plays, and stories. Their human written summaries have three levels of granularity, and we use the paragraph-level summaries for our experiment. Unlike the CNN/Daily Mail dataset, we only experiment on the full size of the BookSum dataset.

The English BART checkpoint has in total 139M parameters to be fine-tuned, while adapters have 14.2M parameters (10%). As an additional parameter-controlled fine-tuning variant, we choose to freeze the entire BART but the last decoder layer, which has 9.5M parameters. The final decoder layer makes up 7% of the entire model, and has a comparable amount of trainable parameters to an adapter. Similar to the previous setting, we use the Adam optimizer with a learning rate of 1e-5 for BART-FT, and 1e-4

[6] https://huggingface.co/ainize/bart-base-cnn
[7] https://github.com/salesforce/booksum

Domain	Data Size		BART-FT			BART-Adapt			BART-FT-LastLayer		
			R1	R2	RL	R1	R2	RL	R1	R2	RL
XL-Sum	original	306.5k	**34.48**	**14.73**	**28.93**	32.94	13.46	27.60	30.20	11.69	25.17
	medium	30.65k	**30.63**	**11.38**	**25.31**	30.15	11.10	25.05	26.70	8.67	21.94
	small	3065	27.27	**8.91**	**22.27**	**27.32**	8.79	22.20	23.06	6.21	18.76
	tiny	307	24.10	**6.52**	19.38	**24.29**	6.41	**19.50**	19.13	4.12	15.54
	micro	31	19.69	4.26	15.73	**20.74**	**4.65**	**16.45**	16.30	2.20	11.43
BookSum	111.6k		**20.27**	**4.01**	15.50	20.22	3.95	**15.57**	19.33	3.56	14.93

Table 4: Results for domain adaptation from CNN/Daily Mail to XL-Sum on English (top) with artificially constrained data sizes, and to BookSum (bottom) with full data size.

Article: Lewis Williams, 20, died on 11 January from a shotgun wound suffered in Wath Road, Mexborough. South Yorkshire Police said two men aged 20 and 49 were arrested on Friday in connection with his death, bringing the total number of arrests to eight . . . **Gold Summary:** Two more people have been arrested in connection with a fatal shooting. **BART-FT Summary:** Two more people have been arrested in connection with the fatal shooting of a man in South Yorkshire. **BART-Adapt Summary:** *Eight* more people have been arrested in connection with the death of a man in South Yorkshire.
Article: BBC News Officials say the country's Olympic Committee will "oversee participation of women athletes who can qualify". The decision will end recent speculation as to whether the entire Saudi team could have been disqualified on grounds of gender discrimination . . . For the desert kingdom, the decision to allow women to compete in the Olympics is a huge step, overturning deep-rooted opposition from those opposed to any public role for women . . . **Gold Summary:** Saudi Arabia is to allow its women athletes to compete in the Olympics for the first time. **BART-FT Summary:** Saudi Arabia is to allow women to compete in next year's Olympic and Paralympic Games. **BART-Adapt Summary:** Saudi Arabia is to allow women to take part in the *2012 Winter* Olympics, officials say.
Article: The vehicle was seen at about 03:45 BST at the fast food giant's branch in Catterick, North Yorkshire. A 19-year-old man was arrested at the site, a short distance from the local golf club, on suspicion of theft and driving while unfit through drink. Police said it was the "most unusual job" of the night but officers managed to "avoid a high-speed pursuit" . . . **Gold Summary:** A stolen golf buggy was seized after being spotted at a McDonald's drive-thru. **BART-FT Summary:** A suspected stolen car was spotted at a McDonald's drive-thru. **BART-Adapt Summary:** A man has been arrested after a car was seen driving into a McDonald's branch.

Table 5: Examples of gold and generated summaries (from models trained on the full dataset) with their corresponding articles selected from the XL-Sum (English) dataset. Summary phrases italicized and highlighted in red denote hallucinations.

for BART-Adapt. We use a batch size of 4 on XL-Sum, and 8 on BookSum. All other hyperparameter settings are identical to those in the language adaptation experiment.

5.2 Results

We report the experiment results in Table 4. The pattern is that for medium to large CNN/Daily Mail data sizes, BART-FT outperforms BART-Adapt significantly. The two methods tie at around 300-3000 training sizes. BART-Adapt wins notably when there are only a handful of examples. This implies that adapters only stand out when the amount of data is extremely limited. In this case, we doubt the importance of training efficiency in adapters when the data size is so small. Instead, we argue that a potential benefit of using adapters is to reduce overfitting. As for BookSum, we can observe that numbers are very similar for both models with BART-FT slightly outperforming BART-Adapt. We argue adapters can do well in domain adaption despite the domain difference as long as there are sufficient training data. Finally, we notice the performance of fine-tuning only the last decoder layer is nowhere near BART-FT or BART-Adapt; this implies the practicability of adapters in summarization.

5.3 Qualitative analysis

To understand the quality of generated summaries between BART-FT and BART-Adapt, we examined a set of randomly selected model outputs from the XL-Sum dataset. We show some examples in Table 5. We find that summaries generated by the two models are roughly the same in terms of informativeness,

Proceedings of the 21st China National Conference on Computational Linguistics, pages 824-835, Nanchang, China, October 14 - 16, 2022.

Task	Data Size	Model	R1	R2	RL
DialogSum	12.5k	BART-FT	**47.40**	**24.66**	**39.03**
		BART-Adapt	47.24	24.57	38.56
SAMSum	14.7k	BART-FT	**49.52**	**24.91**	40.64
		BART-Adapt	49.38	24.69	**40.99**

Table 6: Results for task adaption from CNN/Daily Mail to DialogSum and SAMSum.

Task	Data Size	Model	R1	R2	RL
DialogSum	12.5k	BART-FT*	35.60	16.59	29.69
		BART-Adapt*	**36.35**	**17.03**	**30.25**
SAMSum	14.7k	BART-FT**	**40.91**	**14.82**	**32.32**
		BART-Adapt**	40.42	14.65	32.28

Table 7: Results for robustness analysis of task adaption experiments. Results are directly obtained by using the trained model from the other task without any further training. *denotes the model trained on SAMSum, and **denotes the model trained on DialogSum.

grammaticality, and fluency. Despite summaries being similar in these aspects, we find that BART-Adapt summaries are more prone to hallucinations, which is a well-known problem in abstractive summarization that summaries are not factual with respect to the source or general knowledge.

6 Task Transfer Experiments

6.1 Experimental setup

In previous settings, we conduct experiments with the fine-tuning paradigm on the subject of language and domain adaption. We also conduct experiments on task adaption to further verify our findings. In particular, we experiment with adapting a news summarization model to dialogue summarization. Dialogue summarization is often considered a much different task from monologic texts (e.g. news in our case) summarization due to its unique challenges. Chen et al. (2021) point out that: information flow is reflected in the dialogue discourse structures, summaries are required to be objective, and dialogue is acted at the pragmatic level. For these reasons, we choose to work with the DialogSum (Chen et al., 2021) and the SAMSum (Gliwa et al., 2019) datasets. We follow the previous setting and start with a BART model already fine-tuned on the CNN/Daily Mail dataset, then further train the model on these two datasets separately. We use a batch size of 8 for both DialogSum and SAMSum. All other hyperparameter settings are identical to those in the domain adaptation experiment.

6.2 Results

We report the experiment results in Table 6. We can observe that despite the dataset, BART-FT almost always beats BART-Adapt. However, we can notice that the performance gap is rather small, possibly due to the small dataset sizes. This is consistent with our earlier findings that adapters are on par with fine-tuning when the amount of training data is limited.

6.3 Model robustness

In addition to model performance, we also examine the robustness of models with either fine-tuning or adapters. In particular, we evaluate the model in a zero-shot manner where we directly test the DialogSum model on the SAMSum dataset, and vice versa. We present the results in Table 7. We can first observe that performance drops significantly compared to those in Table 6. Moreover, BART-Adapt has better performance than BART-FT on the DialogSum dataset, and it achieves very similar results on the SAMSum dataset. This suggests that adapters are more robust in a zero-shot setup with fewer data; the reason could be less overfitting introduced by a limited number of parameters in adapters.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 824-835, Nanchang, China, October 14 - 16, 2022.

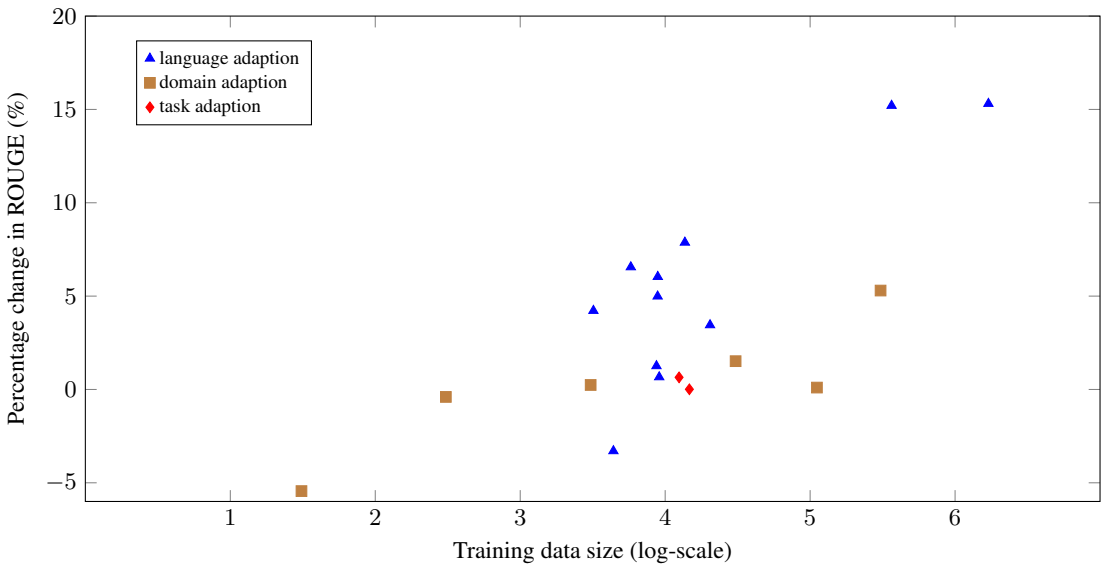


Figure 4: The effect of the training data size on ROUGE difference between the fine-tuning and adapter strategy. We display how much percent FT is better than using adapters. Note that data points from different tasks (with different shapes and colors) are not strictly comparable.

6.4 Effect of data availability on performance

Our results suggest that fine-tuning generally surpasses adapters under all three settings (language, domain, and task adaption). In addition, we observe that the amount of training data affects the performance gap between the two methods. To further validate this observation, we plot the percentage change in ROUGE performance (between those of fine-tuning and those of adapters) against the training size (log-scale) and we provide the visualization in Figure 4. We use the average number of ROUGE-1/2/L to represent the performance. From the plot, we can see that percentage change in ROUGE has an obvious positive relationship with the training data size which means that as the amount of training data increases, the performance gap between BART-FT and BART-Adapt increases as well. Looking at the tasks individually, we can see that for language adaption tasks with relatively small amounts of data, this trend is not very notable. The trend is most salient on domain adaption tasks since we manually controlled the data size for the experiment for adapting CNN/Daily Mail to XL-Sum.

7 Conclusions and Future Work

With large PLMs coming to light, we investigate fine-tuning and adapter strategies for transfer learning in abstractive summarization. We demonstrated that the performance gap between the two strategies is positively correlated with the availability of training resources, despite the languages being tested. Further analysis on domain adaptation and task adaption produces agreeing observations. We conclude that for realistically large summarization datasets, full fine-tuning will guarantee the best output quality. On the other hand, when resources are scarce, the advantages of adapters emerge in the niche market.

Most summarization datasets are web-crawled or machine-translated, resulting in non-optimal data quality. We plan to perform more qualitative analysis on the model outputs such as linguistic interpretation and human evaluation. In addition, we only experimented with fine-tuning and using adapters on mBART and BART for abstractive summarization, so there is room for research on other large PLMs, as well as other NLP tasks in the future.

Acknowledgements

We thank the reviewers of the paper for their feedback. Zheng Zhao is supported by the UKRI Centre for Doctoral Training in Natural Language Processing (grant EP/S022481/1). Pinzhen Chen is supported by a donation to Kenneth Heafield. This work does not necessarily reflect the opinion of the funders.

References

Arthur Brazinskas, Ramesh Nallapati, Mohit Bansal, and Markus Dreyer. 2022. Efficient few-shot fine-tuning for opinion summarization. *Findings of the Association for Computational Linguistics: NAACL 2022*.

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. *34th Conference on Neural Information Processing Systems*.

Yi-Syuan Chen and Hong-Han Shuai. 2021. Meta-transfer learning for low-resource abstractive summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yulong Chen, Yang Liu, and Yue Zhang. 2021. DialogSum challenge: Summarizing real-life scenario dialogues. *Proceedings of the 14th International Conference on Natural Language Generation*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. *Proceedings of the 2nd Workshop on New Frontiers in Summarization*.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. *International Conference on Learning Representations*.

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. *Proceedings of the 36th International Conference on Machine Learning*.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A large scale Chinese short text summarization dataset. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2021. Booksum: A collection of datasets for long-form narrative summarization. *arXiv*.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. *In Findings of the Association for Computational Linguistics: EMNLP 2020*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 824-835, Nanchang, China, October 14 - 16, 2022.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *33rd Conference on Neural Information Processing Systems*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. AdapterHub: A framework for adapting transformers. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. *Proceedings of the 37th International Conference on Machine Learning*.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. NCLS: Neural cross-lingual summarization. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Yanyan Zou, Xingxing Zhang, Wei Lu, Furu Wei, and Ming Zhou. 2020. Pre-training for abstractive document summarization by reinstating source text. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 824-835, Nanchang, China, October 14 - 16, 2022.

A Model Configurations

We tuned the hyperparameters using the validation set. We list the hyperparameters in Table 8, and highlight the selected ones in bold if multiple values are tried out. Instead of an expensive grid search on all combinations, we searched for the best configurations one by one. We performed a single run for each experiment.

Configuration	**Value**
training toolkit	PyTorch (Paszke et al., 2019)
stopping criterion	validation ROUGE
learning rate	1e-3, 5e-3, **1e-4** (mBART+Adapt), 5e-4, **1e-5** (mBART-FT), 5e-5
optimizer	Adam (Kingma and Ba, 2015)
beta1, beta2	0.9, 0.999
weight decay	1e-6
loss function	cross-entropy
decoding batch size	1
decoding beam size	5
decoding len. penalty	1.0
adapter reduction factor	1, **2**, 8, 16
trainable parameters	mBART-FT: 610M mBART-Adapt: 50M

Table 8: Model and training configurations.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 824-835, Nanchang, China, October 14 - 16, 2022.