ADScAI
Applied Data Science & Artificial Intelligence Symposium

Adapter-based Fine-tuning for PRIMERA

Kushan Hewapathirana*[1,2], Nisansa de Silva†[1], C.D. Athuraliya‡[2]
[1]*Department of Computer Science & Engineering, University of Moratuwa, Sri Lanka*
{*kushan.22,†nisansa}@cse.mrt.ac.lk
[2]*ConscientAI, Sri Lanka*
‡cd@conscient.ai

***Keywords*—Multi-document Summarisation, Natural Language Processing, Pre-trained Models, Adapters**

I. Introduction

Multi-document summarisation (MDS) involves generating concise summaries from clusters of related documents. PRIMERA (Pyramid-based Masked Sentence Pre-training for Multi-document Summarisation) is a pre-trained model specifically designed for MDS, utilizing the LED architecture to handle long sequences effectively [1–4]. Despite its capabilities, fine-tuning PRIMERA for specific tasks remains resource-intensive. To mitigate this, we explore the integration of adapter modules—small, trainable components inserted within transformer layers—that allow models to adapt to new tasks by updating only a fraction of the parameters, thereby reducing computational requirements [5–8].

II. Adapter Integration into the LED Architecture of PRIMERA

Integrating adapters into the LED architecture of PRIMERA necessitates modifications to the adapter-transformers library to support the unique aspects of LED. The following components were updated to facilitate this integration:

- `src/transformers/__init__.py`: Ensured that the LED model is recognized within the transformers library's initialization process.
- `src/transformers/adapters/__init__.py`: Included adapter support for the LED model within the adapters module.
- `adapters/composition.py`: Allowed for the composition of multiple adapters within the LED model, enabling modular task-specific adaptations.
- `adapters/head_utils.py`: Adjusted utility functions to accommodate the architecture of the LED model when managing adapter heads.
- `adapters/models/auto/adapter_model.py`: Enabled automatic integration of adapters into the LED model.
- `adapters/wrappers/configuration.py`: Ensured that adapter configurations are compatible with the settings of the LED model.
- `transformers/models/led/modeling_led.py`: Incorporated adapter modules directly into the forward pass of the LED model, facilitating their functionality during training and inference.

These modifications allow PRIMERA to utilize multiple adapters, each fine-tuned for specific summarisation tasks, without altering the core model parameters. This modular approach enhances the adaptability and efficiency of the model, which is illustrated by Figure 1.

III. Types of Adapters

Several adapter architectures have been proposed for efficient fine-tuning of transformer models:

- **Bottleneck Adapters**: Introduce bottleneck feed-forward layers in each layer of a Transformer model. These adapters consist of a down-projection matrix that projects the layer hidden states into a lower dimension, a non-linearity function, an up-projection back to the original hidden dimension, and a residual connection [5].
- **AdapterFusion**: Combines multiple pre-trained adapters to leverage knowledge from various tasks, enhancing the performance of the model on new tasks by fusing information from related adapters [6].
- **LoRA (Low-Rank Adaptation)**: Injects trainable low-rank decomposition matrices into the layers of a pre-trained model, allowing adaptation to new tasks with minimal parameter updates [7].
- **Prefix Tuning**: Introduces trainable prefix tokens to the input of each Transformer layer, enabling the model to adapt to new tasks by learning task-specific prefixes while keeping the original model parameters frozen [8].

IV. Benefits of Adapter-Based Fine-Tuning

Implementing adapters within the LED architecture of PRIMERA offers several advantages:

- **Parameter Efficiency**: Adapters require tuning only a small subset of parameters, significantly reducing the computational resources needed compared to full model fine-tuning [5].
- **Modularity**: Each adapter can be trained for a specific task and activated as needed, promoting a modular ar-

https://doi.org/10.31705/ADScAI.2025.57

chitecture that simplifies multi-task learning and model management [6].

- **Scalability**: The reduced resource requirements enable the scaling of fine-tuning processes across numerous tasks, making it feasible to deploy the model in diverse application scenarios[5, 7].

V. Experimental Evaluation

To evaluate the effectiveness of adapter-based fine-tuning on PRIMERA, we utilize the Multi-News dataset [9], a large-scale benchmark for multi-document summarization. This dataset consists of news articles paired with human-written summaries, making it well-suited for assessing summarization performance. The performance of the model was evaluated using ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [10] scores, which are standard metrics for summarisation tasks[1–3].

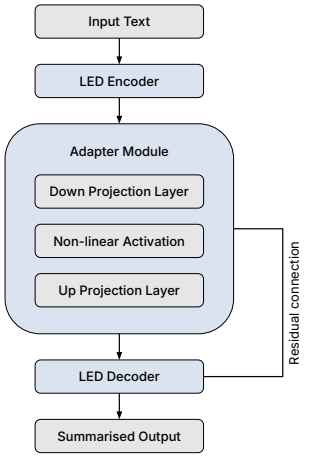

Fig. 1. Adapter Integration into PRIMERA

TABLE I
ROUGE Scores on Multi-News Dataset

Method	ROUGE-1	ROUGE-2	ROUGE-L
PRIMERA (Vanilla Model) [1]	42.0 [1]	13.6 [1]	20.8 [1]
PRIMERA (Fully Fine-Tuned) [1]	**49.9** [1]	**21.1** [1]	**25.9** [1]
Bottleneck Adapter [5]	47.1	17.4	21.6
AdapterFusion [6]	48.5	18.2	22.7

The results indicate that adapter-based fine-tuning achieves performance comparable to traditional full-model fine-tuning while utilizing significantly fewer trainable parameters. Specifically, the AdapterFusion method approaches the ROUGE scores of the fully fine-tuned PRIMERA model, confirming the efficacy of the adapter integration.

VI. Conclusion

The integration of adapter modules into the LED architecture of PRIMERA presents a viable solution for efficient and scalable fine-tuning across multiple MDS tasks. This approach maintains high performance while substantially reducing the computational burden associated with training large-scale models. The modular nature of adapters further enhances the adaptability of the model, allowing for seamless transitions between tasks without necessitating extensive retraining. Future work may explore the application of this methodology to other transformer-based architectures and investigate the potential of adapter fusion techniques to further enhance model performance.

References

[1] W. Xiao, I. Beltagy, G. Carenini, and A. Cohan, "Primera: Pyramid-based masked sentence pre-training for multi-document summarization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 5245–5263.

[2] K. Hewapathirana, N. De Silva, and C. Athuraliya, "Multi-document summarization: a comparative evaluation," in *2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS)*. IEEE, 2023, pp. 19–24.

[3] C. Ma, W. E. Zhang, M. Guo, H. Wang, and Q. Z. Sheng, "Multi-document summarization via deep learning techniques: A survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, 2022.

[4] K. Hewapathirana, N. de Silva, and C. Athuraliya, "M2ds: Multilingual dataset for multi-document summarisation," in *International Conference on Computational Collective Intelligence*. Springer, 2024, pp. 219–231.

[5] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*. PMLR, 2019, pp. 2790–2799.

[6] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, "Adapterfusion: Non-destructive task composition for transfer learning," in *EACL*, 2021, pp. 487–503.

[7] E. J. Hu, Wallis *et al.*, "Lora: Low-rank adaptation of large language models," in *ICLR*, 2021.

[8] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4582–4597.

[9] A. R. Fabbri, I. Li, T. She, S. Li, and D. Radev, "Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model," in *ACL*, 2019, pp. 1074–1084.

[10] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.