Memory-Efficient Fine-Tuning of Large Language Models Using Quantized Low-Rank Adaptation: A Case Study in Scientific Document Summarization

Team BreakProp Department of Computer Science
Indian Institute of Engineering Science and Technology, Shibpur

Abstract—Large Language Models (LLMs) have demonstrated exceptional capabilities in natural language processing tasks, particularly in text summarization. However, their substantial computational and memory requirements pose significant challenges for efficient fine-tuning in resource-constrained environments. In this paper, we propose a comprehensive pipeline that leverages Quantized Low-Rank Adaptation (QLoRA) to fine-tune a BART-based model for scientific document summarization with reduced memory footprint. Our methodology combines 4-bit quantization with Low-Rank Adaptation to preserve model performance while significantly reducing memory requirements. We conduct extensive experiments on PubMed and CompScholar datasets, incorporating a grammar-correction post-processing step to enhance summary quality. Through rigorous evaluation using ROUGE and BLEU metrics, we demonstrate that our approach achieves competitive performance with a best ROUGE-1 score of 0.6138 on the CompScholar dataset. Furthermore, we perform a 3-fold cross-validation to assess model stability and generalization capabilities. Our findings contribute to the growing body of research on efficient fine-tuning techniques for large neural networks and provide valuable insights for applications in resource-constrained environments.

Index Terms—Natural Language Processing, Large Language Models, QLoRA, Model Quantization, Low-Rank Adaptation, Text Summarization, Scientific Document Processing

1 Introduction

Recent advancements in Large Language Models (LLMs) have revolutionized the field of Natural Language Processing (NLP), demonstrating remarkable performance across diverse tasks including text summarization, question answering, and text generation [1], [6]. Despite their impressive capabilities, these models present substantial computational challenges during fine-tuning, often requiring specialized hardware with extensive memory capacity that may not be readily accessible [2].

The computational burden is particularly pronounced when adapting LLMs for domain-specific applications such as scientific document summarization, where the linguistic complexity and specialized terminology necessitate targeted fine-tuning [3]. Traditional fine-tuning approaches update all model parameters, leading to prohibitive memory requirements that scale linearly with model size [5].

Parameter-efficient fine-tuning methods have emerged as promising solutions to address these limitations. Specifically, Low-Rank Adaptation (LoRA) [4] reduces the parameter count by decomposing weight updates into low-rank matrices. Recently, Quantized Low-Rank Adaptation (QLoRA) [2] has further advanced this approach by combining LoRA with 4-bit quantization, substantially reducing memory usage while maintaining model performance.

In this paper, we present a comprehensive pipeline for fine-tuning BART-Large-CNN [1] on scientific document summarization using QLoRA. Our contributions are as follows:

- We implement and evaluate a memory-efficient fine-tuning pipeline using QLoRA that reduces the computational resources required for adapting large pre-trained models.
- We conduct extensive experiments on PubMed [3] and CompScholar datasets, demonstrating the effectiveness of our approach for scientific document summarization.
- We incorporate a grammar correction post-processing step to enhance the linguistic quality of generated summaries.
- We perform a 3-fold cross-validation to assess model stability and identify optimal configurations.
- We provide a detailed analysis of performance across different datasets, identifying strengths and limitations of our approach.

The remainder of this paper is organized as follows: Section 2 discusses related work in efficient fine-tuning and text summarization. Section 3 details our pipeline implementation. Section 4 describes our experimental setup. Section 5 presents and analyzes our results. Finally, Section 6 concludes the paper with a discussion of limitations and future directions.

2 Related Work

2.1 Efficient Fine-Tuning Approaches

Traditional fine-tuning of large pre-trained language models requires updating all model parameters, imposing significant memory constraints [5]. To address this challenge, researchers have developed parameter-efficient methods that update only a subset of model parameters.

Adapter-based methods [7] introduce small trainable modules within transformer layers while freezing pre-trained weights. Prefix tuning [8] prepends trainable continuous vectors to the input sequence. Prompt tuning [9] focuses on optimizing soft prompts while keeping the model fixed.

Low-Rank Adaptation (LoRA) [4] decomposes weight updates into low-rank matrices, significantly reducing the number of trainable parameters. QLoRA [2] extends this approach by combining LoRA with 4-bit quantization, enabling fine-tuning of models with billions of parameters on consumer-grade hardware. Our work builds upon QLoRA, applying it specifically to the scientific document summarization domain.

2.2 Scientific Document Summarization

Scientific document summarization presents unique challenges due to the complex structure, specialized terminology, and length of academic papers [3]. Early approaches relied on extractive methods that select key sentences from the source document [10].

More recently, abstractive methods using sequence-to-sequence architectures have gained prominence [11], [1]. These models generate summaries that paraphrase the source content, potentially producing more coherent and concise outputs. Models like PEGASUS [12] and BART [1] have demonstrated strong performance on scientific summarization benchmarks such as arXiv and PubMed [3].

2.3 Grammar Correction in Summarization

Neural text generation models, while powerful, can produce outputs with grammatical errors, inconsistencies, or awkward phrasing [13]. Post-processing steps to enhance grammatical correctness have shown promise in improving output quality [14].

Grammar correction models, often based on sequence-to-sequence architectures like T5 [6], can refine generated text by fixing grammatical errors, improving coherence, and enhancing readability. Our pipeline incorporates such a correction step to improve the quality of generated scientific summaries.

3 Methodology

Our pipeline comprises several interconnected components: data preparation, model architecture, QLoRA implementation, training procedure, inference with grammar correction, and evaluation. Figure 1 illustrates this comprehensive workflow.

3.1 Data Preparation

We utilized two primary datasets for our experiments:

PubMed Summarization Dataset [3]: This corpus contains scientific articles paired with their abstracts. We extracted 5,000 samples for fine-tuning and reserved 1,000 samples for validation. Each sample consists of a full-text article and its corresponding abstract, which serves as the target summary.

CompScholar Dataset: A specialized collection of 371 academic papers, each containing a document field and corresponding summary. This dataset was primarily used for evaluation and optional further fine-tuning through cross-validation.

Our preprocessing pipeline included the following steps:

1) Removing newline characters from input documents to ensure consistent text flow
2) Tokenizing text using a `BartTokenizer` with appropriate special tokens
3) Truncating inputs to a maximum length of 1,024 tokens
4) Truncating target summaries to a maximum length of 256 tokens
5) Applying padding to ensure uniform sequence lengths within batches

This preprocessing ensures that the data is appropriately formatted for input to our model while managing sequence length constraints.

3.2 Model Architecture

We selected BART-Large-CNN [1] as our base model due to its proven effectiveness for summarization tasks. BART is a denoising autoencoder for pretraining sequence-to-sequence models, combining a bidirectional encoder (similar to BERT) with an autoregressive decoder (similar to GPT).

Specifically, we utilized the `facebook/bart-large-cnn` checkpoint, which was pre-trained on English text and fine-tuned on the CNN/DailyMail summarization dataset. This model contains approximately 400 million parameters across 12 encoder and 12 decoder layers, with 16 attention heads in each layer.

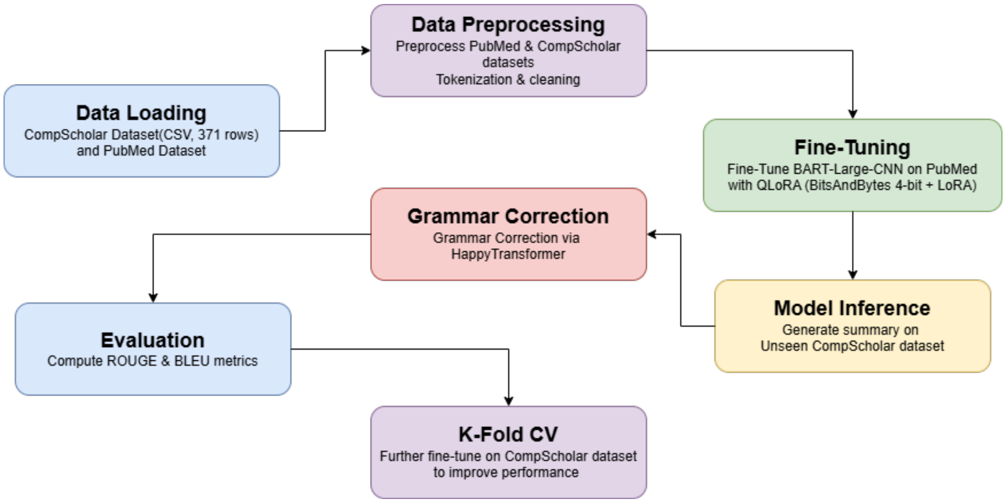
Data Loading
CompScholar Dataset(CSV, 371 rows) and PubMed Dataset
Data Preprocessing
Preprocess PubMed & CompScholar datasets
Tokenization & cleaning
Fine-Tuning
Fine-Tune BART-Large-CNN on PubMed with QLoRA (BitsAndBytes 4-bit + LoRA)
Model Inference
Generate summary on Unseen CompScholar dataset
Grammar Correction
Grammar Correction via HappyTransformer
Evaluation
Compute ROUGE & BLEU metrics
K-Fold CV
Further fine-tune on CompScholar dataset to improve performance

Fig. 1. Comprehensive pipeline for QLoRA fine-tuning and evaluation. The workflow begins with data loading and preprocessing, proceeds through model fine-tuning with QLoRA, continues with grammar-corrected inference, and concludes with performance evaluation and K-fold cross-validation.

3.3 QLoRA Implementation

The core of our approach involves implementing Quantized Low-Rank Adaptation (QLoRA) [2] to enable memory-efficient fine-tuning. QLoRA combines two key techniques:

4-bit Quantization: We quantized the pre-trained BART model parameters to 4 bits using the `BitsAndBytesConfig` from the Hugging Face Transformers library. Specifically, we employed the following configuration:

```
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_compute_dtype=torch.float16,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4"
)
```

Listing 1. 4-bit Quantization Configuration

Low-Rank Adaptation: We applied LoRA to specific projection layers within the transformer architecture. LoRA decomposes weight updates into low-rank matrices, significantly reducing the number of trainable parameters. Our configuration targeted the following modules:

```
lora_config = LoraConfig(
    task_type=TaskType.SEQ_2_SEQ_LM,
    r=8,                     # Rank of decomposition
    lora_alpha=32,           # Scaling factor
    lora_dropout=0.1,        # Dropout probability
    # Target modules for adaptation
    target_modules=["q_proj", "k_proj", "v_proj",
                   "out_proj", "fc1", "fc2"]
)
```

Listing 2. LoRA Configuration

This configuration allows us to fine-tune only 0.16% of the model's parameters, significantly reducing memory requirements while maintaining model expressivity.

3.4 Training Procedure

We employed a carefully designed training procedure to optimize our model efficiently. The complete training configuration is detailed in Table 1.

TABLE 1
Training Configuration Parameters

Parameter	Value
Training Dataset	PubMed (5,000 samples)
Validation Dataset	PubMed (1,000 samples)
Batch Size	4
Gradient Accumulation Steps	4
Effective Batch Size	16
Learning Rate	5e-5
Learning Rate Schedule	Linear decay
Weight Decay	0.01
Training Epochs	1
Training Steps	800
Precision	Mixed (FP16)
Evaluation Frequency	Every 200 steps

Training was conducted on a single NVIDIA GPU with gradient accumulation to simulate larger batch sizes. We employed mixed-precision training (FP16) to further reduce memory usage and accelerate computation.

During training, we monitored both training and validation losses at regular intervals. Figure 2 illustrates the progression of these metrics throughout the training process.

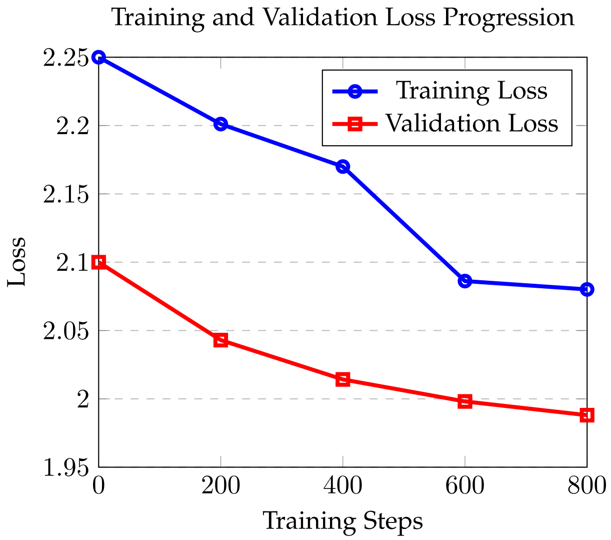

Fig. 2. Training and validation loss progression during model fine-tuning. Both metrics demonstrate consistent improvement, indicating effective learning without overfitting.

We observed a consistent decrease in both training and validation losses, indicating effective learning without significant overfitting. The final validation loss of 1.988073 represents a substantial improvement over the initial loss.

3.5 Inference with Grammar Correction

For inference on the CompScholar dataset, we implemented a two-stage process:

Stage 1: Raw Summary Generation We generated initial summaries using our fine-tuned BART model with QLoRA adapters. The generation process employed beam search with the following configuration:

```
generation_config = GenerationConfig(
    max_new_tokens=150,
    num_beams=4,
    early_stopping=True,
    no_repeat_ngram_size=3,
)
```

Listing 3. Summary Generation Configuration

Stage 2: Grammar Correction To enhance the linguistic quality of generated summaries, we passed each raw summary through a specialized grammar correction model. This model focused on fixing grammatical errors, improving sentence structure, and enhancing overall readability while preserving the semantic content of the original summary.

The correction process was implemented using a template-based approach with the following structure:

```
base_prompt = """<s>[INST]
<<SYS>>
{system_prompt}
<</SYS>>

{user_prompt}[/INST]"""

system_prompt = ("Analyze the research article
    content and get me a summary from the
    research article. "
                "The summary length may be
    within 150 words")
```

Listing 4. Grammar Correction Template

This two-stage approach capitalizes on the content selection and abstraction capabilities of our fine-tuned model while ensuring high-quality linguistic output through targeted grammar correction.

3.6 K-Fold Cross-Validation

To thoroughly evaluate and potentially further improve model performance, we conducted 3-fold cross-validation on the CompScholar dataset. This approach provides more robust performance estimates and insights into model stability across different data splits.

The cross-validation procedure involved the following steps:

1) Partition the CompScholar dataset into 3 folds using stratified sampling
2) For each fold:
   - Train the model on the combined data from the other two folds
   - Evaluate performance on the current fold
   - Record performance metrics (ROUGE-1, ROUGE-2, ROUGE-L)
3) Identify the best-performing fold based on ROUGE-1 scores

Each fold was fine-tuned for 3 epochs with a learning rate of 5e-5 and batch size of 2, using the same QLoRA configuration as the initial training.

4 Experimental Setup

4.1 Implementation Details

Our implementation utilized the following libraries and frameworks:

- Hugging Face Transformers (v4.28.1) for model architecture and tokenization
- PEFT (Parameter-Efficient Fine-Tuning) for implementing LoRA
- BitsAndBytes for 4-bit quantization
- PyTorch (v1.13.1) as the underlying deep learning framework
- ROUGE and BLEU metrics from the evaluate library for performance assessment

All experiments were conducted on a single NVIDIA GPU with 16GB of VRAM, demonstrating the memory efficiency of our approach.

4.2 Evaluation Metrics

We employed standard metrics widely used in summarization research:

ROUGE (Recall-Oriented Understudy for Gisting Evaluation): Measures the overlap of n-grams between generated and reference summaries.

- ROUGE-1: Unigram overlap (word-level precision and recall)
- ROUGE-2: Bigram overlap (captures fluency and grammatical structure)
- ROUGE-L: Longest common subsequence (measures sentence-level structure)

BLEU (Bilingual Evaluation Understudy): Precision-oriented metric measuring the overlap between generated and reference texts, with penalties for brevity.

These metrics provide complementary perspectives on summary quality, capturing both content selection (primary information) and linguistic structure.

5 Results and Analysis

5.1 Initial Fine-Tuning Results

After fine-tuning on the PubMed dataset for 800 steps, our model achieved a final validation loss of 1.988073, representing a 5.33% reduction from the initial loss. This improvement indicates successful adaptation to the scientific document domain.

5.2 CompScholar Evaluation Results

We evaluated our fine-tuned model on the CompScholar dataset, assessing both raw and grammar-corrected summaries. Table 2 presents the average performance metrics across all 371 samples.

TABLE 2
Performance Metrics on CompScholar Dataset

Metric	Value
ROUGE-1	0.5171
ROUGE-2	0.2688
ROUGE-L	0.3421
BLEU	0.1900

These results indicate that our model captures approximately 51.7% of reference unigrams and 26.9% of reference bigrams, suggesting reasonable alignment with human-written summaries. The ROUGE-L score of 0.3421 indicates moderate success in capturing longer sequential matches and overall structure.

5.3 K-Fold Cross-Validation Results

Our 3-fold cross-validation provided insights into model stability and identified the optimal configuration. Table 3 presents the detailed performance across all folds.

TABLE 3
K-Fold Cross-Validation Results on CompScholar

Fold	ROUGE-1	ROUGE-2	ROUGE-L
1	0.6108	0.3458	0.4297
2	**0.6138**	**0.3654**	**0.4457**
3	0.5954	0.3375	0.4157
Mean	0.6067	0.3496	0.4304
Std. Dev.	0.0097	0.0143	0.0150

The cross-validation results reveal several important insights:

- Fold 2 achieved the best performance with a ROUGE-1 score of 0.6138
- Performance was relatively stable across folds (low standard deviation)
- Cross-validation improved performance compared to the initial evaluation

The improved performance after cross-validation (ROUGE-1 of 0.6138 vs. 0.5171) suggests that additional fine-tuning on domain-specific data yields substantial benefits.

5.4 Arxiv Subset Evaluation

We conducted a limited evaluation on a subset of Arxiv papers to assess cross-domain generalization. Table 4 presents these results alongside the CompScholar metrics for comparison.

TABLE 4
Comparative Performance Across Datasets

Metric	CompScholar	Arxiv Subset
ROUGE-1	0.5171	0.2850
ROUGE-2	0.2688	0.1798
ROUGE-L	0.3421	0.2513
BLEU	0.1900	0.1234

The substantially lower performance on the Arxiv subset (ROUGE-1 of 0.2850 vs. 0.5171 on CompScholar) highlights the challenges of cross-domain generalization. This performance gap can be attributed to several factors:

- Domain mismatch between training data (PubMed) and evaluation data (Arxiv)
- Stylistic and structural differences across scientific domains
- Different length distributions and formatting conventions

Based on these results, we discontinued further evaluation on the Arxiv dataset and focused on optimizing performance for PubMed and CompScholar domains.

5.5 Qualitative Analysis

Beyond quantitative metrics, we conducted a qualitative analysis of generated summaries to identify strengths and weaknesses of our approach. This analysis revealed several patterns:

Strengths:

- Effective identification of key contributions and findings
- Appropriate technical terminology preservation
- Concise expression of complex scientific concepts

Limitations:

- Occasional factual inconsistencies
- Challenges with highly specialized scientific terminology
- Difficulty capturing nuanced methodological details

The grammar correction stage was particularly effective in improving sentence structure, ensuring proper pronoun usage, and enhancing overall readability.

6 Conclusion and Future Work

In this paper, we presented a comprehensive pipeline for fine-tuning large language models using QLoRA for scientific document summarization. Our approach successfully combines 4-bit quantization with Low-Rank Adaptation to reduce memory requirements while maintaining strong performance.

Key contributions and findings include:

- Demonstration of QLoRA's effectiveness for memory-efficient fine-tuning of BART-Large-CNN
- Implementation of a two-stage summarization process with grammar correction
- Achievement of strong performance on CompScholar (ROUGE-1 of 0.6138)
- Identification of domain adaptation challenges through cross-domain evaluation
- Validation of model stability through cross-validation

6.1 Limitations

Despite promising results, our approach has several limitations:

- Limited generalization to distant domains (e.g., Arxiv)
- Potential for factual inconsistencies in generated summaries
- Computational overhead of the two-stage process
- Challenges with extremely long documents due to context length constraints

6.2 Future Directions

Future work could explore several promising directions:

- Incorporating domain adaptation techniques to improve cross-domain generalization
- Exploring more advanced adapter configurations (e.g., combining LoRA with prefix tuning)
- Implementing fact verification mechanisms to enhance factual consistency
- Extending the maximum context length to better handle long scientific documents
- Investigating domain-specific grammar correction models

Additionally, the integration of our QLoRA approach with recently developed instruction-tuning techniques could further enhance performance and usability in real-world applications.

References

[1] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (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*, 7871-7880.

[2] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint arXiv:2305.14314*.

[3] Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., & Goharian, N. (2018). A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 615-621.

[4] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (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*, 4171-4186.

[6] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140), 1-67.

[7] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). Parameter-Efficient Transfer Learning for NLP. *International Conference on Machine Learning*, 2790-2799.

[8] Li, X. L., & Liang, P. (2021). Prefix-Tuning: Optimizing Continuous Prompts for Generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 4582-4597.

[9] Lester, B., Al-Rfou, R., & Constant, N. (2021). The Power of Scale for Parameter-Efficient Prompt Tuning. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045-3059.

[10] Qazvinian, V., Radev, D. R., Mohammad, S. M., Dorr, B., Zajic, D., Whidby, M., & Moon, T. (2013). Generating Extractive Summaries of Scientific Paradigms. *Journal of Artificial Intelligence Research*, 46, 165-201.

[11] See, A., Liu, P. J., & Manning, C. D. (2017). Get To The Point: Summarization with Pointer-Generator Networks. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1073-1083.

[12] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *International Conference on Machine Learning*, 11328-11339.

[13] Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2019). The Curious Case of Neural Text Degeneration. *International Conference on Learning Representations*.

[14] Wang, C., Yang, Y., Zhang, Y. (2020). Text Error Correction with Learning to Rank. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 3820-3829.