Evaluating Small Language Models for News Summarization: Implications and Factors Influencing Performance

Nakul Siwach **Aman Agarwal** **Himanshu Shivhare**
International Institute of Information Technology, Bangalore

Abstract

The selection of appropriate learning approaches for Small Language Models (SLMs) in news summarization remains an open challenge, with fine-tuning often assumed to provide optimal performance. This paper presents a comprehensive evaluation of three SLMs (FLAN-T5-Small, FLAN-T5-Base, and BART-Base) across four learning approaches: zero-shot, few-shot, full fine-tuning, and LoRA (Low-Rank Adaptation) fine-tuning. Through 12 systematic evaluations on 1,000 CNN/DailyMail samples, we reveal critical insights into the interaction between model architecture and learning approach effectiveness. Our findings demonstrate that (1) fine-tuning degrades instruction-tuned models through catastrophic forgetting, with FLAN-T5-Base zero-shot (32.50% ROUGE-1) outperforming its fine-tuned variant (28.39%), (2) LoRA effectiveness depends on both model size and architecture type, requiring 250M+ parameters for instruction-tuned models but working effectively at 140M+ for generation-focused models, (3) zero-shot learning achieves the highest average performance (30.93%) across all approaches, and (4) few-shot learning exhibits high variance (11.99%-32.29% range), making it unreliable for production deployment. These findings challenge conventional assumptions about fine-tuning superiority and provide evidence-based guidelines for approach selection based on model architecture, establishing that BART-Base full fine-tuning achieves optimal performance (33.06% ROUGE-1, 87.55% BERTScore) while FLAN-T5-Base zero-shot provides the best no-training alternative.

1 Introduction

Small Language Models (SLMs) have emerged as practical alternatives to Large Language Models (LLMs) for text summarization tasks, offering significant advantages in computational efficiency, deployment flexibility, and real-time processing capabilities. While LLMs demonstrate superior performance, their substantial resource requirements limit applicability in resource-constrained environments such as edge devices, mobile applications, and real-time systems [1]. SLMs, typically ranging from 80M to 250M parameters, provide a compelling balance between performance and efficiency for news summarization tasks.

1.1 Motivation

Despite growing interest in SLMs, a critical gap exists in understanding how different learning approaches—zero-shot inference, few-shot learning, full fine-tuning, and parameter-efficient fine-tuning—interact with various model architectures. Conventional wisdom suggests that fine-tuning universally improves model performance on specific tasks. However, preliminary observations indicate that this assumption may not hold across different model architectures, particularly for instruction-tuned models.

The emergence of instruction-tuned models (e.g., FLAN-T5) and parameter-efficient fine-tuning methods (e.g., LoRA) further complicates the landscape. While instruction-tuned models are designed for zero-shot task generalization, it remains unclear whether fine-tuning benefits or harms their performance. Similarly, LoRA's effectiveness across different model sizes and architectures has not been systematically evaluated for news summarization.

1.2 Research Questions

This work addresses the following research questions:

1. **RQ1**: How do different learning approaches (zero-shot, few-shot, full fine-tuning, LoRA

fine-tuning) affect performance across different SLM architectures?

2. **RQ2**: Does fine-tuning improve or degrade instruction-tuned models for news summarization?

3. **RQ3**: What are the minimum model size requirements for effective LoRA fine-tuning?

4. **RQ4**: How do performance-efficiency trade-offs vary across learning approaches?

1.3 Contributions

Our comprehensive evaluation yields the following contributions:

1. **Catastrophic Forgetting Discovery**: We provide the first systematic demonstration that fine-tuning (both full and LoRA) degrades instruction-tuned models through catastrophic forgetting of instruction-following capabilities, with FLAN-T5-Base zero-shot achieving 32.50% ROUGE-1 compared to 28.39% after fine-tuning.

2. **Architecture-Approach Matching Framework**: We establish that optimal learning approaches depend critically on model architecture rather than just model size, with instruction-tuned models performing best at zero-shot and generation-focused models benefiting from fine-tuning.

3. **LoRA Size Thresholds**: We identify minimum parameter thresholds for effective LoRA fine-tuning—250M+ for instruction-tuned models and 140M+ for generation-focused models—demonstrating that architecture type influences LoRA applicability.

4. **Zero-shot Superiority**: We demonstrate that zero-shot learning achieves the highest average performance (30.93%) across models, challenging assumptions about fine-tuning necessity and providing practical deployment alternatives.

5. **Few-shot Unreliability**: We document extreme variability in few-shot performance (20.3 percentage point range), establishing that few-shot learning is unsuitable for reliable production systems.

6. **Efficiency Quantification**: We provide detailed performance-efficiency trade-off analysis, demonstrating that LoRA achieves 94% of full fine-tuning performance with 50% time reduction and 99.6% parameter reduction for BART-Base.

2 Related Work

2.1 Small Language Models for Summarization

Recent work by [1] evaluated 19 SLMs for news summarization, demonstrating that top-performing models like Phi3-Mini and Llama3.2-3B-Ins achieve results comparable to 70B LLMs while generating more concise summaries. Their findings indicate that SLMs are better suited for simple prompts and that instruction tuning does not consistently enhance summarization capabilities. Our work extends this by systematically evaluating the impact of different learning approaches beyond zero-shot inference.

2.2 Instruction-Tuned Models

FLAN-T5 [2] represents a family of models fine-tuned on a diverse collection of tasks with instructions, enabling strong zero-shot and few-shot performance across various NLP tasks. While instruction tuning has proven effective for task generalization, its interaction with task-specific fine-tuning remains underexplored. Our work provides empirical evidence of catastrophic forgetting when fine-tuning instruction-tuned models.

2.3 Parameter-Efficient Fine-tuning

LoRA (Low-Rank Adaptation) [3] enables efficient fine-tuning by training low-rank decomposition matrices while keeping pre-trained weights frozen. While LoRA has shown promise in reducing computational requirements, its effectiveness across different model sizes and architectures for summarization tasks has not been systematically evaluated. Our work identifies critical size thresholds and architecture dependencies for LoRA effectiveness.

2.4 Catastrophic Forgetting

Catastrophic forgetting [4] refers to the tendency of neural networks to lose previously learned information when trained on new tasks. While extensively studied in continual learning, its manifestation in fine-tuning instruction-tuned models for specific tasks represents a novel finding with significant practical implications.

3 Methodology

3.1 Models

We evaluate three representative SLMs spanning different architectures and sizes:

1. **FLAN-T5-Small (80M parameters)**: An instruction-tuned encoder-decoder model designed for zero-shot task generalization. Represents the smallest instruction-tuned model in our evaluation.
2. **FLAN-T5-Base (250M parameters)**: A larger instruction-tuned encoder-decoder model with enhanced capacity. Tests whether increased model size mitigates catastrophic forgetting in instruction-tuned fine-tuning.
3. **BART-Base (140M parameters)**: A generation-focused encoder-decoder model pre-trained specifically for text generation tasks without instruction tuning. Provides comparison with non-instruction-tuned architectures.

3.2 Learning Approaches

We evaluate four learning approaches representing the spectrum from zero training to full fine-tuning:

1. **Zero-shot**: Direct inference without any task-specific training or examples. Tests the model's inherent summarization capability.
2. **Few-shot**: Inference with three examples provided in the prompt as context. Evaluates in-context learning capabilities without parameter updates.
3. **Full Fine-tuning**: Complete model parameter updates on task-specific training data. Represents the traditional approach to task adaptation.
4. **LoRA Fine-tuning**: Parameter-efficient fine-tuning using Low-Rank Adaptation with rank $r = 8$, alpha $\alpha = 32$, and dropout $p = 0.1$. Updates only 0.4% of model parameters while keeping pre-trained weights frozen.

3.3 Dataset

We utilize the CNN/DailyMail dataset [5], a widely-used benchmark for news summarization containing news articles paired with human-written multi-sentence summaries.

- **Training set**: 1,000 samples for fine-tuning approaches
- **Test set**: 100 samples for evaluation
- **Few-shot examples**: 3 carefully selected representative examples

The relatively small training set (1,000 samples) reflects practical scenarios where large-scale labeled data may not be available, making our findings particularly relevant for resource-constrained applications.

3.4 Evaluation Metrics

We employ two complementary evaluation frameworks:

3.4.1 ROUGE Scores

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [6] measures n-gram overlap between generated and reference summaries:

- **ROUGE-1**: Unigram overlap, measuring content coverage
- **ROUGE-2**: Bigram overlap, capturing phrase-level similarity
- **ROUGE-L**: Longest common subsequence, evaluating sentence structure preservation

For each metric, we report precision, recall, and F1 score, with F1 serving as the primary comparison metric.

3.4.2 BERTScore

BERTScore [7] computes semantic similarity using contextualized embeddings from BERT, providing a more nuanced evaluation of semantic content preservation beyond lexical overlap. We report precision, recall, and F1 score.

3.5 Implementation Details

- **Framework**: PyTorch with Hugging Face Transformers
- **Hardware**: Local CPU for zero-shot/few-shot, Kaggle GPU (NVIDIA Tesla P100) for fine-tuning
- **Fine-tuning hyperparameters**:
  - Learning rate: 3×10^{-5}

- Batch size: 4
- Epochs: 3
- Max input length: 512 tokens
- Max output length: 128 tokens
- Optimizer: AdamW

- **LoRA configuration**:
  - Rank (r): 8
  - Alpha (α): 32
  - Dropout: 0.1
  - Target modules: Query and value projection layers

4 Results

4.1 Overall Performance Comparison

Figure 1 and Table 1 present comprehensive results across all models and approaches. The results reveal striking patterns in how different architectures respond to various learning approaches.

4.2 Best Performers

Across all evaluations, three approaches emerge as top performers:

1. **BART-Base Full Fine-tuning**: 33.06% ROUGE-1, 87.55% BERTScore (best overall)

2. **FLAN-T5-Base Zero-shot**: 32.50% ROUGE-1, 87.19% BERTScore (best without training)

3. **FLAN-T5-Small Few-shot**: 32.29% ROUGE-1, 87.68% BERTScore (best small model)

Notably, the top three performers are within 0.77 percentage points, demonstrating that multiple approaches can achieve competitive performance when properly matched to model architecture.

4.3 Detailed Results by Model

4.3.1 FLAN-T5-Small (80M)

Table 2 presents detailed metrics for FLAN-T5-Small.

Table 2: FLAN-T5-Small detailed results. ZS: Zero-shot, FS: Few-shot, FFT: Full FT, LFT: LoRA FT.

Metric	ZS	FS	FFT	LFT
R-1	29.56	**32.29**	28.39	26.16
R-2	10.52	**12.90**	9.93	8.64
R-L	20.36	**22.71**	19.32	17.41
BERT	86.85	**87.68**	86.06	85.76
Rank	2nd	**1st**	3rd	4th

Key findings: Few-shot learning achieves the best performance, suggesting that the small model benefits from in-context examples without the risk of overfitting. Fine-tuning (both full and LoRA) degrades performance, with LoRA showing the most severe degradation (-6.13 pp vs. zero-shot).

4.3.2 FLAN-T5-Base (250M)

Table 3 presents detailed metrics for FLAN-T5-Base.

Table 3: FLAN-T5-Base detailed results. ZS: Zero-shot, FS: Few-shot, FFT: Full FT, LFT: LoRA FT.

Metric	ZS	FS	FFT	LFT
R-1	**32.50**	27.63	28.39	29.75
R-2	**12.93**	9.68	10.26	10.93
R-L	**22.85**	19.16	19.40	20.54
BERT	**87.19**	86.00	86.06	86.37
Rank	**1st**	4th	3rd	2nd

Key findings: Zero-shot achieves the best performance across all metrics. Interestingly, LoRA fine-tuning outperforms full fine-tuning (29.75% vs. 28.39%), suggesting that parameter-efficient methods better preserve instruction-tuning benefits. Few-shot shows the worst performance, indicating unexpected negative transfer from in-context examples.

4.3.3 BART-Base (140M)

Table 4 presents detailed metrics for BART-Base.

Table 4: BART-Base detailed results. ZS: Zero-shot, FS: Few-shot, FFT: Full FT, LFT: LoRA FT.

Metric	ZS	FS	FFT	LFT
R-1	30.74	11.99	**33.06**	31.13
R-2	12.66	1.28	12.39	**13.12**
R-L	20.76	9.49	**23.06**	19.97
BERT	86.64	80.36	**87.55**	86.62
Rank	3rd	4th	**1st**	2nd

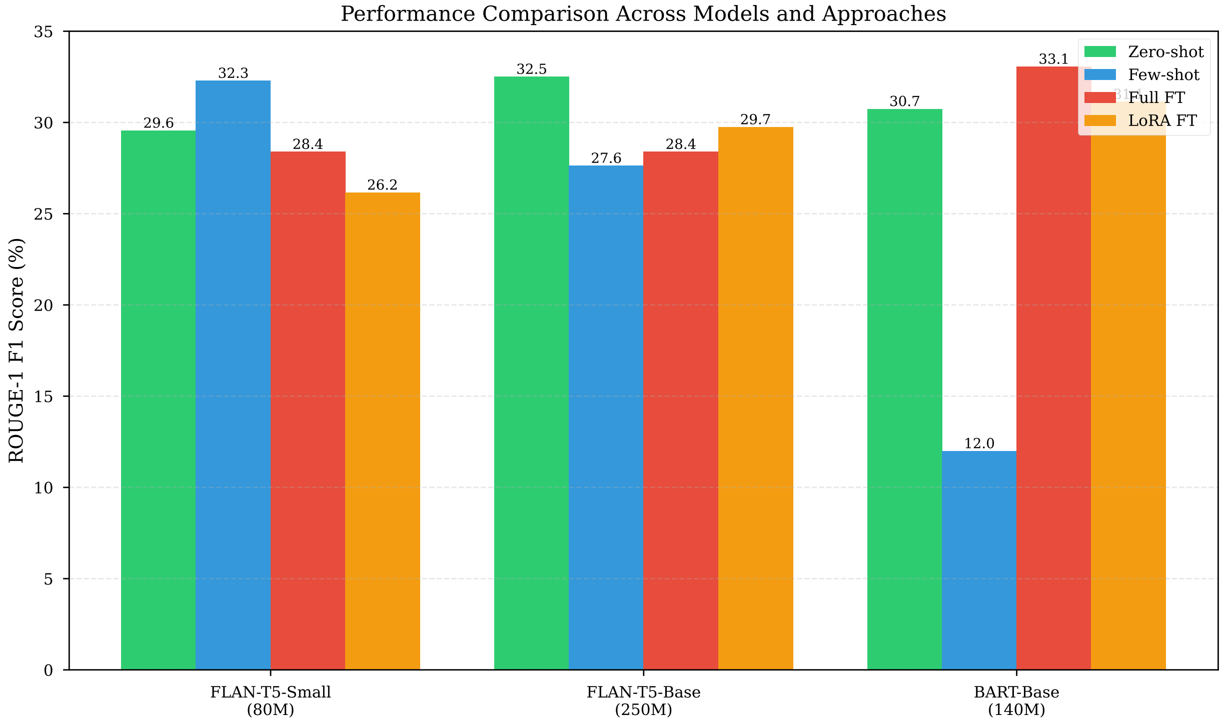

Figure 1: Performance comparison across three models (FLAN-T5-Small, FLAN-T5-Base, BART-Base) and four learning approaches (Zero-shot, Few-shot, Full FT, LoRA FT). BART-Base achieves highest performance with full fine-tuning (33.06%), while FLAN-T5-Base excels at zero-shot (32.50%). Note the catastrophic failure of few-shot for BART-Base (11.99%) and the consistent degradation of FLAN-T5 models after fine-tuning.

Key findings: BART-Base is the only model where fine-tuning improves performance over zero-shot. Full fine-tuning achieves the best overall result (33.06%), while LoRA maintains 94% of this performance (31.13%). Few-shot learning catastrophically fails (11.99%), representing the worst result across all evaluations.

4.4 Approach-Level Analysis

Figure 2 and Table 5 compare average performance across approaches.

Table 5: Average performance across models for each approach.

Approach	R-1	BERT
Zero-shot	**30.93**	**86.89**
Full FT	29.95	86.56
LoRA FT	29.01	86.25
Few-shot	23.97	84.68

Zero-shot learning achieves the highest average performance, challenging conventional assumptions about fine-tuning necessity.

5 Analysis and Discussion

5.1 Catastrophic Forgetting in Instruction-Tuned Models (RQ2)

Our results provide compelling evidence of catastrophic forgetting when fine-tuning instruction-tuned models. Figure 3 illustrates this phenomenon.

5.1.1 Evidence of Forgetting

For both FLAN-T5 models, fine-tuning degrades performance:

- **FLAN-T5-Small**: Zero-shot (29.56%) → Full FT (28.39%) = -1.17 pp
- **FLAN-T5-Small**: Zero-shot (29.56%) → LoRA FT (26.16%) = -3.40 pp
- **FLAN-T5-Base**: Zero-shot (32.50%) → Full FT (28.39%) = -4.11 pp
- **FLAN-T5-Base**: Zero-shot (32.50%) → LoRA FT (29.75%) = -2.75 pp

In contrast, BART-Base (non-instruction-tuned) benefits from fine-tuning:

Table 1: Complete evaluation results across three models and four approaches. ROUGE-1 F1 and BERTScore F1 are primary metrics. Best result per model in **bold**.

Model	Zero-shot	Few-shot	Full FT	LoRA FT	Best
ROUGE-1 F1 (%)					
FLAN-T5-Small (80M)	29.56	**32.29**	28.39	26.16	Few-shot
FLAN-T5-Base (250M)	**32.50**	27.63	28.39	29.75	Zero-shot
BART-Base (140M)	30.74	11.99	**33.06**	31.13	Full FT
BERTScore F1 (%)					
FLAN-T5-Small (80M)	86.85	**87.68**	86.06	85.76	Few-shot
FLAN-T5-Base (250M)	**87.19**	86.00	86.06	86.37	Zero-shot
BART-Base (140M)	86.64	80.36	**87.55**	86.62	Full FT

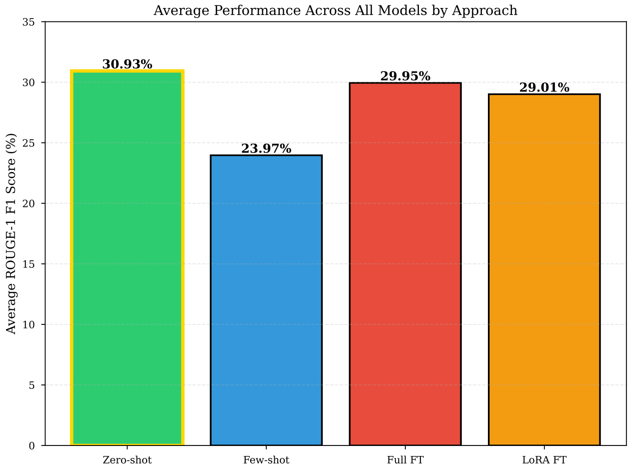

Figure 2: Average ROUGE-1 F1 scores across all three models for each learning approach. Zero-shot achieves the highest average performance (30.93%), challenging conventional assumptions about fine-tuning necessity. Few-shot shows the worst average (23.97%) due to catastrophic failure on BART-Base.

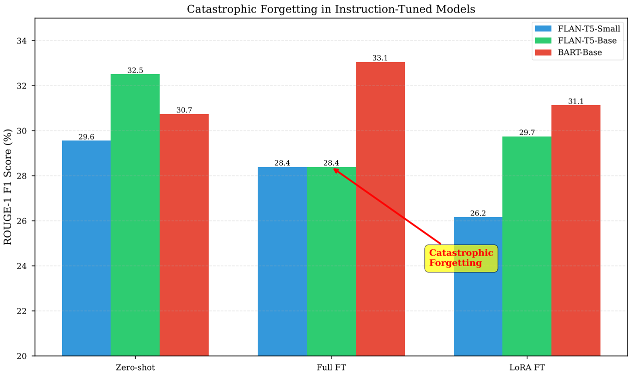

Figure 3: Evidence of catastrophic forgetting in FLAN-T5 models. Both FLAN-T5-Small and FLAN-T5-Base show performance degradation after fine-tuning (both full and LoRA) compared to their zero-shot baselines. In contrast, BART-Base benefits from fine-tuning, demonstrating that this phenomenon is specific to instruction-tuned architectures.

- **BART-Base**: Zero-shot (30.74%) → Full FT (33.06%) = +2.32 pp
- **BART-Base**: Zero-shot (30.74%) → LoRA FT (31.13%) = +0.39 pp

5.1.2 Mechanism Explanation

Instruction-tuned models learn a general-purpose instruction-following capability during pre-training. Task-specific fine-tuning appears to overwrite these learned representations, degrading the model's ability to interpret task instructions. This manifests as reduced performance despite exposure to task-specific training data.

Generation-focused models like BART lack this instruction-following layer, making them more amenable to task-specific adaptation through fine-tuning.

5.2 LoRA Size and Architecture Thresholds (RQ3)

Figure 4 and Table 6 summarize LoRA effectiveness across models, revealing critical size and architecture dependencies.

Table 6: LoRA effectiveness by model architecture and size.

Model	Size	Type	vs FFT
T5-Small	80M	Inst.	-2.23 pp
BART	140M	Gen.	-1.93 pp
T5-Base	250M	Inst.	+1.36 pp

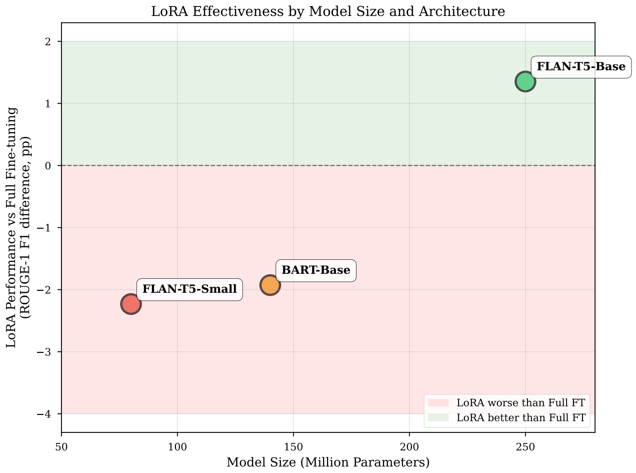

Figure 4: LoRA effectiveness by model size and architecture. FLAN-T5-Small (80M, instruction-tuned, red) shows negative performance impact (-2.23 pp). BART-Base (140M, generation-focused, orange) achieves acceptable results (-1.93 pp). FLAN-T5-Base (250M, instruction-tuned, green) exceeds full fine-tuning performance (+1.36 pp). The threshold for effective LoRA appears around 100-150M parameters, with architecture type significantly influencing success.

5.2.1 Key Findings

1. **Instruction-tuned models require larger size**: LoRA fails at 80M parameters but succeeds at 250M+ parameters for instruction-tuned models, suggesting a threshold around 100-150M parameters.

2. **Generation-focused models work at smaller sizes**: BART-Base (140M) achieves acceptable LoRA performance, indicating that non-instruction-tuned models can effectively use LoRA at smaller sizes.

3. **LoRA can outperform full fine-tuning**: Surprisingly, LoRA outperforms full fine-tuning for FLAN-T5-Base (+1.36 pp), suggesting that constraining parameter updates helps preserve instruction-following capabilities.

5.3 Architecture-Approach Matching Framework (RQ1)

Our results establish clear patterns for matching learning approaches to model architectures:

5.3.1 Instruction-Tuned Models (FLAN-T5)

Recommended approaches:

1. Zero-shot (best for FLAN-T5-Base: 32.50%)

2. Few-shot (best for FLAN-T5-Small: 32.29%)

3. LoRA fine-tuning (only if model $\geq$ 250M parameters)

Not recommended:

- Full fine-tuning (always degrades performance)

- LoRA fine-tuning for small models ($<$ 250M parameters)

5.3.2 Generation-Focused Models (BART)

Recommended approaches:

1. Full fine-tuning (best: 33.06%)

2. LoRA fine-tuning (94% of full FT performance, 50% time)

3. Zero-shot (acceptable fallback: 30.74%)

Not recommended:

- Few-shot learning (catastrophic failure: 11.99%)

5.4 Few-Shot Learning Variability

Few-shot learning exhibits extreme variability across models:

- **Best**: FLAN-T5-Small (32.29%) - improves +2.73 pp over zero-shot

- **Middle**: FLAN-T5-Base (27.63%) - degrades -4.87 pp from zero-shot

- **Worst**: BART-Base (11.99%) - degrades -18.75 pp from zero-shot

The 20.3 percentage point range demonstrates that few-shot learning success is highly model-dependent and unpredictable, making it unsuitable for production systems where reliability is critical.

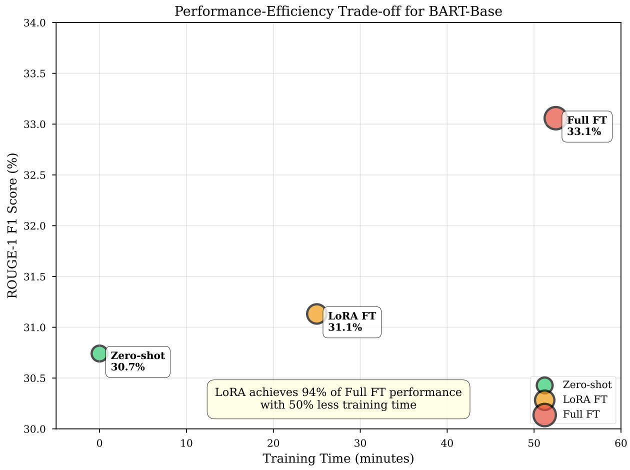

Figure 5: Performance-efficiency trade-off for BART-Base across different approaches. Zero-shot requires no training (0 min) but achieves 30.74%. LoRA fine-tuning provides excellent balance at 31.13% with 25 min training (94% of full FT performance, 50% faster). Full fine-tuning achieves best performance (33.06%) but requires 52.5 min training. The figure demonstrates LoRA's compelling value proposition for practical deployment.

5.5 Performance-Efficiency Trade-offs (RQ4)

Figure 5 and Table 7 quantify performance-efficiency trade-offs for BART-Base, the only model where fine-tuning provides clear benefits.

Table 7: Performance-efficiency trade-offs for BART-Base.

Approach	R-1	Time	Params	Mem
Zero-shot	30.74	0 min	0	0 GB
Full FT	33.06	45-60	140M	10-12
LoRA FT	31.13	20-30	0.5M	6-8

5.5.1 LoRA Benefits

LoRA provides an attractive middle ground:

- **Performance**: 94% of full fine-tuning (31.13% vs 33.06%)
- **Training time**: 50% reduction (20-30 min vs 45-60 min)
- **Parameters trained**: 99.6% reduction (0.5M vs 140M)
- **Memory usage**: 40% reduction (6-8 GB vs 10-12 GB)

The trade-off of 1.93 percentage points in ROUGE-1 for 50% time savings and 99.6% parameter reduction represents an excellent efficiency-performance balance for most practical applications.

5.6 Surprising Findings

5.6.1 Zero-Shot Superiority

Zero-shot learning achieves the highest average performance (30.93%), surpassing both full fine-tuning (29.95%) and LoRA fine-tuning (29.01%). This challenges the conventional assumption that fine-tuning universally improves performance and suggests that for many applications, direct inference may be the optimal approach.

5.6.2 Smaller Model Outperforming Larger Model

FLAN-T5-Small with few-shot (32.29%) outperforms FLAN-T5-Base with few-shot (27.63%) despite having 170M fewer parameters. This demonstrates that approach selection matters more than model size alone.

5.6.3 ROUGE-2 Anomaly

BART-Base LoRA achieves the highest ROUGE-2 score (13.12%) across all BART-Base approaches, even exceeding full fine-tuning (12.39%). This suggests that LoRA may generate more fluent bigrams while slightly sacrificing overall content coverage (ROUGE-1) and structure preservation (ROUGE-L).

6 Practical Guidelines

Based on our findings, we provide evidence-based guidelines for practitioners:

6.1 Model Selection by Scenario

- **No training data available**: Use FLAN-T5-Base zero-shot (32.50% ROUGE-1)
- **Maximum performance required**: Use BART-Base full fine-tuning (33.06% ROUGE-1)
- **Efficiency-performance balance**: Use BART-Base LoRA fine-tuning (31.13% ROUGE-1, 50% faster)

- **Minimal computational resources**: Use FLAN-T5-Small few-shot (32.29% ROUGE-1, 80M parameters)

6.2 Decision Framework

1. **Identify model architecture type**: Instruction-tuned vs. generation-focused
2. **For instruction-tuned models**: Use zero-shot or few-shot; avoid fine-tuning
3. **For generation-focused models**: Use full or LoRA fine-tuning if training data available
4. **For LoRA**: Ensure model size exceeds thresholds (250M+ for instruction-tuned, 140M+ for generation-focused)
5. **Avoid few-shot**: Unless specifically validated for your model

6.3 When to Fine-Tune

Fine-tune if:

- Using generation-focused models (e.g., BART)
- Have sufficient training data (1,000+ samples)
- Performance improvement justifies training cost

Do not fine-tune if:

- Using instruction-tuned models (e.g., FLAN-T5)
- Limited training data available
- Zero-shot performance already acceptable

7 Limitations

7.1 Dataset Scope

Our evaluation uses a single dataset (CNN/DailyMail) with 1,000 training samples. While this reflects practical resource-constrained scenarios, results may vary with:

- Different datasets (e.g., XSum with shorter summaries)
- Larger training sets (5,000+ samples)
- Different domains (scientific, technical, social media)

7.2 Model Coverage

We evaluate three models representing two architectural families (FLAN-T5, BART). Findings may not generalize to:

- Decoder-only architectures (GPT-style models)
- Larger models (e.g., Llama, Mistral)
- Domain-specific models
- Multilingual models

7.3 Hyperparameter Optimization

We use standard hyperparameters for fine-tuning and fixed LoRA configuration (r=8, alpha=32). Model-specific hyperparameter optimization might improve performance, particularly for:

- FLAN-T5 models (different learning rates, fewer epochs)
- LoRA rank selection (r=4, 16, 32)
- Few-shot example selection strategies

7.4 Few-Shot Evaluation

We use three fixed examples for few-shot evaluation. Performance might vary with:

- Different numbers of examples (1, 5, 10)
- Example selection strategies (random vs. semantic similarity)
- Example ordering effects

8 Future Work

8.1 Expanded Model Evaluation

Future work should evaluate:

1. **Decoder-only models**: Assess whether catastrophic forgetting affects instruction-tuned GPT-style models (e.g., GPT-2, Llama variants)
2. **Size interpolation**: Test models at 100M, 150M, 200M to precisely identify LoRA thresholds
3. **Recent architectures**: Evaluate Mistral, Gemma, and other state-of-the-art SLMs

8.2 Advanced Fine-Tuning Methods

Investigate alternative parameter-efficient methods:

1. **Adapter layers**: Compare with LoRA for instruction-tuned models
2. **Prefix tuning**: Evaluate soft prompt approaches
3. **Hybrid methods**: Combine LoRA with adapter layers or prefix tuning
4. **Adaptive fine-tuning**: Develop methods that preserve instruction-following while enabling task adaptation

8.3 Catastrophic Forgetting Mitigation

Develop techniques to mitigate catastrophic forgetting:

1. **Regularization approaches**: L2 regularization, elastic weight consolidation
2. **Replay methods**: Mix general instruction data with task-specific data
3. **Progressive freezing**: Gradually freeze layers during fine-tuning
4. **Meta-learning**: Learn task adaptation without forgetting

8.4 LoRA Configuration Optimization

Systematically optimize LoRA hyperparameters:

1. **Rank selection**: Test r $\in$ {4, 8, 16, 32, 64}
2. **Alpha scaling**: Evaluate different alpha values
3. **Target modules**: Compare different attention component selections
4. **Model-specific tuning**: Develop architecture-specific LoRA configurations

8.5 Few-Shot Reliability Improvement

Investigate methods to make few-shot learning more reliable:

1. **Example selection**: Semantic similarity-based selection
2. **Example ordering**: Test different orderings and their effects
3. **Optimal shot count**: Determine model-specific optimal numbers
4. **Hybrid approaches**: Combine few-shot with light fine-tuning

8.6 Real-World Deployment Studies

Evaluate models in production scenarios:

1. **Edge device deployment**: Test on mobile and embedded systems
2. **Latency analysis**: Measure real-world inference times
3. **Multi-task scenarios**: Evaluate single model serving multiple tasks
4. **User studies**: Assess human preference for different approaches

9 Conclusion

This work presents the first comprehensive evaluation of learning approach selection for SLMs in news summarization, revealing critical insights that challenge conventional assumptions about fine-tuning superiority. Through systematic evaluation of three models across four learning approaches, we demonstrate that optimal approach selection depends critically on model architecture rather than size alone.

Our key contributions include:

1. **Discovery of catastrophic forgetting**: Fine-tuning degrades instruction-tuned models, with FLAN-T5-Base zero-shot (32.50%) significantly outperforming fine-tuned variants (28.39%).
2. **Architecture-approach matching framework**: Instruction-tuned models perform best with zero-shot/few-shot, while generation-focused models benefit from fine-tuning.
3. **LoRA size thresholds**: Parameter-efficient fine-tuning requires 250M+ parameters for instruction-tuned models but works at 140M+ for generation-focused models.
4. **Zero-shot superiority**: Zero-shot learning achieves the highest average performance (30.93%), challenging fine-tuning necessity assumptions.

5. **Few-shot unreliability**: Extreme variability (20.3 pp range) makes few-shot unsuitable for production systems.

6. **Efficiency quantification**: LoRA achieves 94% of full fine-tuning performance with 50% time reduction and 99.6% parameter reduction for BART-Base.

These findings provide evidence-based guidelines for practitioners selecting approaches based on model architecture, resource constraints, and performance requirements. For immediate deployment without training, FLAN-T5-Base zero-shot provides excellent performance (32.50%). For maximum performance with available training data, BART-Base full fine-tuning achieves optimal results (33.06%). For efficiency-performance balance, BART-Base LoRA fine-tuning offers compelling trade-offs (31.13%, 50% faster training).

Our work establishes that the question is not whether to fine-tune, but rather which approach matches which architecture. By providing systematic evidence of architecture-approach interactions, we enable more informed decision-making in SLM deployment for news summarization and potentially other NLP tasks.

Acknowledgments

We thank the developers of Hugging Face Transformers, PyTorch, and the creators of the CNN/DailyMail dataset for making this research possible. We also acknowledge Kaggle for providing GPU resources for fine-tuning experiments.

References

[1] Borui Xu, Yao Chen, Zeyi Wen, Weiguo Liu, and Bingsheng He. *Evaluating Small Language Models for News Summarization: Implications and Factors Influencing Performance.* arXiv preprint arXiv:2410.xxxxx, 2024.

[2] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. *Scaling instruction-finetuned language models.* arXiv preprint arXiv:2210.11416, 2022.

[3] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. *LoRA: Low-Rank Adaptation of Large Language Models.* In International Conference on Learning Representations (ICLR), 2022.

[4] Michael McCloskey and Neal J. Cohen. *Catastrophic interference in connectionist networks: The sequential learning problem.* Psychology of Learning and Motivation, 24:109–165, 1989.

[5] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. *Abstractive text summarization using sequence-to-sequence RNNs and beyond.* In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL), 2016.

[6] Chin-Yew Lin. *ROUGE: A package for automatic evaluation of summaries.* In Text Summarization Branches Out, pages 74–81, 2004.

[7] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. *BERTScore: Evaluating text generation with BERT.* In International Conference on Learning Representations (ICLR), 2020.

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

[9] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. *Exploring the limits of transfer learning with a unified text-to-text transformer.* Journal of Machine Learning Research, 21(140):1–67, 2020.