Improving LLM Test Summarization Performance Through Fine-tuning

LI Xiaoting
the Hong Kong University of Science and Technology
Department of Big Data Technology
xliiz@connect.ust.hk

***Abstract*—Large language models (LLMs) have shown promising capabilities in natural language processing tasks, including text summarization. This study explores the effectiveness of fine-tuning the open source Llama-2-7b LLM for the specific task of text summarization. The research approach involves fine-tuning the Llama-2-7b model using the LoRA method and analyzing the impact of varying the LoRA parameters on the model's performance. The fine-tuned models are then evaluated on both in-domain and out-of-domain datasets, using a range of evaluation metrics, including ROUGE, BERTScore, and GPT-4 score. This comparative analysis aims to provide insights into the generalization capabilities of the fine-tuned models and the trade-offs between model performance, computational efficiency, and task-specific adaptation. The findings of this study are expected to contribute to the understanding of effective fine-tuning strategies for LLMs and evaluation metrics of GPT-4 score in the text summarization task. The results may inform the selection and application of appropriate fine-tuning strategy for LLM in other natural language processing tasks. Code upload in github repo: https://github.com/Lxt115/Fine-tunning-Llama2-on-text-summariaztion**

I. Introduction

A. Background

The landscape of generative AI has witnessed a dynamic evolution, driven by recent breakthroughs exemplified by models like ChatGPT,Llama and etc. With transformative potential spanning natural language generation, translation, and imaginative content creation, generative AI has become a focal point of global discussions. A new era in artificial intelligence is underway, marked by the prominence of LLMs in applications such as customer service, virtual assistants, content creation, and programming assistance.

Abstractive summarization models aim to extract essential information from long documents and to generate short, concise and readable text. Recently, neural abstractive summarization models have achieved remarkable performance [4] [5]), and large-scale generative pre-training [6] has shown itself to be surprisingly effective at generation tasks, including abstractive summarization. Large Language Models (LLMs) have made remarkable progress on a diverse array of natural language processing tasks. One such task is abstractive text summarization, which involves generating a concise version that captures the most salient information from a given document [1]. Recent works study the domain adaptation abilities of LLMs on the summarization task. However, the research is still limited to a single domain [2] [3]. There is a lack of research across domains to better understand the abilities of these models to adapt to different targets.

B. Objectives

- Examine the Impact of Fine-Tuning on LLM Performance - Explore the effects of fine-tuning the LLM parameters on the abstractive summarization task. Analyze how different fine-tuning strategies, such as varying the model parameters, impact the overall performance of the LLM on in-domain and out-of-domain datasets.
- Assess the Generalization Capabilities of LLMs - Evaluate the LLMs' ability to adapt to and perform well on a diverse range of input domains, beyond the initial training data. Investigate the models' robustness and performance on both in-domain and out-of-domain summarization datasets.
- Comprehensive Evaluation and Comparison - Utilize multiple evaluation metrics, including ROUGE, BERTScore, and GPT-4 score, to assess the quality and coherence of the generated summaries. Compare the performance of the fine-tuned LLMs across different datasets and evaluation metrics to gain a deeper understanding of their strengths and limitations.
- Provide Insights and Recommendations - Analyze the research findings to draw conclusions about the capabilities and limitations of LLMs for abstractive text summarization. Provide insights and recommendations for future research and practical applications of LLMs in the field of natural language processing and text summarization.

II. Related Work

A. Fine-tuning methods

Many applications in natural language processing rely on adapting one large-scale, pre-trained language model to multiple downstream applications. Such adaptation is usually done via fine-tuning, which updates all the parameters of the pre-trained model. The major downside of fine-tuning is that the new model contains as many parameters as in the original model.

Fine-tuning LLMs with tens or hundreds of billions of parameters is computationally intensive and time-consuming. To avoid complete model fine-tuning, numerous parameter efficient fine-tuning (PEFT) techniques try to achieve acceptable model fine-tuning performance at reduced costs. As compared

to full fine-tuning, PEFT performs better in low-resource setups, achieves comparable performance on medium-resource scenarios, and performs worse than full fine-tuning under high-resource availability.

Adapter Tuning adds a few trainable parameters within the transformer block. The adapter layer is a sequence of feature downscaling, non-linearity, and upscaling. Low-Rank Adaptation (LoRA) [12] is an adapter-based method for parameter-efficient finetuning that adds trainable low-rank decomposition matrices to different layers of a neural network, then freezes the network's remaining parameters(Figure 1. LoRA is most commonly applied to transformer models, in which case it is common to add the low-rank matrices to some of the linear projections in each transformer layer's self-attention. The learned weights are fused with the original weights for inference, avoiding latency.

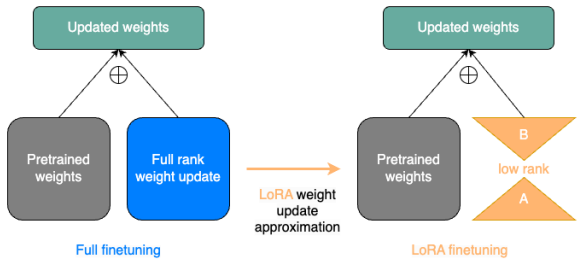

Fig. 1: Comparison of Full Finetuning vs. LoRA Weight Update

III. Experimental Setup

A. Datasets

To explore how different domain data can affect the performance of original Llama-2-7b model and fine-tuning models, the study utilizes three distinct datasets.

- **CNN/Daily Mail(DM)** [9] [10]: is a dataset for text summarization. Human generated abstractive summary bullets were generated from news stories in CNN and Daily Mail websites as questions (with one of the entities hidden), and stories as the corresponding passages from which the system is expected to answer the fill-in the-blank question. The authors released the scripts that crawl, extract and generate pairs of passages and questions from these websites. In all, the corpus has 286,817 training pairs, 13,368 validation pairs and 11,487 test pairs, as defined by their scripts. The source documents in the training set have 766 words spanning 29.74 sentences on an average while the summaries consist of 53 words and 3.72 sentences.
- **XSum** [8] : The Extreme Summarization (XSum) dataset is a dataset for evaluation of abstractive single-document summarization systems. The goal is to create a short, one-sentence new summary answering the question "What is the article about?". The dataset consists of 226,711 news articles accompanied with a one-sentence summary. The articles are collected from BBC articles (2010 to 2017) and cover a wide variety of domains (e.g., News, Politics, Sports, Weather, Business, Technology, Science, Health, Family, Education, Entertainment and Arts). The official random split contains 204,045 (90%), 11,332 (5%) and 11,334 (5) documents in training, validation and test sets, respectively.
- **DialogSum** [11] : DialogSum is a large-scale dialogue summarization dataset, consisting of 13,460 dialogues with corresponding manually labeled summaries and topics.

This study used the first dataset (CNN/Daily Mail) as in-domain dataset and the other two as out-of-domain datasets to test the performance of models.

B. Large Language Model

This project use Llama-2 [7] as the foundational Large Language Model (LLM).

Llama-2 encompasses a series of generative text models spanning parameters from 7 billion to 70 billion. Notably, the meticulously fine-tuned Llama Chat model emerges as a standout, undergoing robust training on diverse publicly available instruction datasets and benefiting from insights derived from over 1 million human annotations.

As a versatile family of generative text models, Llama-2 is tailored for a broad spectrum of natural language understanding and generation tasks. While it excels in assistant-like chat scenarios, its adaptability extends across various applications, including content generation for blog posts, articles, stories, poems, social interactions, summarization, and question answering. This flexibility positions Llama-2 as a valuable resource for text-related tasks across diverse fields, encompassing AI and NLP.

Meta, in collaboration with Microsoft, unveiled Llama-2 as the successor to LLaMA. Llama-2 was developed and launched in three configurations: 7, 13, and 70 billion parameters(Figure 2). Utilizing an optimized transformer architecture, Llama-2 functions as an auto-regressive language model. The tuned versions employ supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) to align with human preferences for helpfulness and safety.

Llama 2

MODEL SIZE (PARAMETERS)	PRETRAINED	FINE-TUNED FOR CHAT USE CASES
7B	Model architecture:	Data collection for helpfulness and safety:
13B	Pretraining Tokens: 2 Trillion	Supervised fine-tuning: Over 100,000
70B	Context Length: 4096	Human Preferences: Over 1,000,000

Fig. 2: Different configurations of Llama-2

This project chose to deploy the Llama-2-7b-hf. The decision to opt for Llama-2-7b is rooted in practical considerations, as its smaller parameter size makes it conducive for efficient operation on personal laptops. If further performance enhancement is desired, alternative models with different parameter sizes can be considered.

C. Fine-tuning Settings

Zhou et al. [13] argue that 1k samples are enough to fine-tune LLMs. This study experimented with 1k, 2k, and 5k training samples and find that the training loss show little difference with only difference on training time. This study selected many different composition of LoRA parameters on rank, lora-alpha and learning-rate with dropout of 0.1 and the paged AdamW optimizer (Table I).

	dataset	lora_alpha	rank	lr
1	CNN/DM	64	32	1e-4
2	CNN/DM	32	64	1e-4
3	CNN/DM	16	64	1e-4
4	CNN/DM	64	16	8e-4
5	CNN/DM	16	64	4e-3

TABLE I: Parameters Settings

For test part, I use Azure OpenAI which is the collaboration between Microsoft Azure and OpenAI. Azure OpenAI Service provides REST API access to OpenAI's powerful language models including GPT-4o, GPT-4 Turbo with Vision, GPT-4, GPT-3.5-Turbo, and Embeddings model series. The response genrator's temperature used the default 1 and the max new tokens used 128.

I tried the training configuration with these 6 cases and get the training loss picture(Fig 3. Find that case1/2/3 almost get the same train loss tendency and case5's loss is too big(Fig 4). Finally use Case3 and case4 as the final two fine-tuning

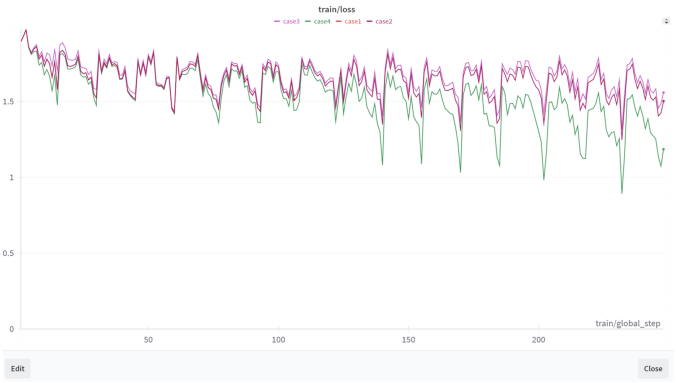

Fig. 3: Train Loss-case1/2/3/4

models in evaluation step. Then, obtain the generate response summary from the original pre-trained model(Llama-2-7b-hf) and two Prompt-based Efficient Fine-Tuning (PEFT) models on 3 different datasets, considering 50 rows from each dataset.

D. Evaluation Metrics

1) Rouge(Recall-Oriented Understudy for Gisting Evaluation):: a set of criteria for evaluating automatic abstracts and

Fig. 4: Train Loss-all cases

machine translation. It calculates a score by comparing an automatically generated summary or translation with a set of reference summaries (usually manually generated) to measure the "similarity" between the automatically generated summary or translation and the reference summary.

$$R_{lcs} = \frac{LCS(X,Y)}{m}$$

$$p_{lcs} = \frac{LCS(X,Y)}{n}$$

$$F_{lcs} = \frac{\left(1+\beta^2\right) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}}$$

2) BERTScore:: ROUGE relies on the exact presence of words in both the predicted and reference texts, failing to interpret the underlying semantics. This is where BERTScore comes in and leverages the contextual embeddings from the BERT model, aiming to evaluate the similarity between a predicted and a reference sentence in the context of machine-generated text. By comparing embeddings from both sentences, BERTScore captures semantic similarities that might be missed by traditional n-gram based metrics. This study used F1 score and computed the mean.

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j$$

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j$$

$$F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}$$

3) GPT4-score: : BERTScore may not fully grasp subtleties and high-level concepts that a human evaluator might understand, reliance solely on this metric could lead to misinterpreting the actual quality and nuances of the summary. In addition to these traditional metrics, this study showcase a method (G-Eval [14]) that leverages Large Language Models (LLMs) as a novel, reference-free metric for assessing abstractive summaries. In this case, I use gpt-4o to score candidate outputs. gpt-4 has effectively learned an internal model of language quality that allows it to differentiate between fluent,

coherent text and low-quality text. Harnessing this internal scoring mechanism allows auto-evaluation of new candidate outputs generated by an LLM. Here defined four distinct criteria [15]:

- **Relevance(1-5)**: Evaluates if the summary includes only important information and excludes redundancies.
- **Coherence(1-5)**: Assesses the logical flow and organization of the summary.
- **Consistency(1-5)**: Checks if the summary aligns with the facts in the source document.
- **Fluency(1-3)**: Rates the grammar and readability of the summary.

For each evaluation criteria, I craft custom prompts that take the original document and the summary as inputs. Chain-of-thought generation techniques are leveraged to guide the GPT-4 model to output a numeric score between 1 and 5 for each criteria.

The defined prompts are then used to generate scores from GPT-4, which are compared across the different summaries. In this demonstration, a direct scoring function is employed where GPT-4 produces a discrete score (1-5) for each evaluation metric. Alternatively, normalizing the scores and taking a weighted sum could result in more robust, continuous scores that better reflect the quality and diversity of the summarization outputs.

IV. Conclusion and Future Work

A. Conclusion

I get summarization from three models-orignal Llama-2-7b and two peft models. Then compute the rouge score and BERTScore between the generated summary and the original manual abstractive summary(Table II). In the result table, Llama 7b refers to the original model while Llama 7b[1] and Llama 7b[2] refers to models that trained from the case3 and case4.

	ROUGE			**BERTScore**
	ROUGE-1	ROUGE-2	ROUGE-L	
in-domain dataset				
Llama2 7b	0.31	0.10	0.28	0.855
Llama2 7b[1]	**0.37**	**0.14**	**0.34**	0.868
Llama2 7b[2]	0.36	0.13	0.33	**0.874**
out-of-domain dataset[1]				
Llama2 7b	0.15	0.01	0.13	0.846
Llama2 7b[1]	**0.20**	**0.04**	**0.17**	**0.848**
Llama2 7b[2]	0.19	0.04	0.16	0.847
out-of-domain dataset[2]				
Llama2 7b	0.14	0.03	0.13	**0.843**
Llama2 7b[1]	**0.18**	**0.04**	**0.17**	0.838
Llama2 7b[2]	0.15	0.02	0.14	0.823

TABLE II: Rouge score and BERTScore of three summariztion

To clearly demonstrate the effect of fine-tuning on the GPT-4 metric across different datasets(Table III), I calculated the ratio between the fine-tuned model's score and the original model's score.The results show that the ratio for the in-domain dataset is significantly higher than the ratios for the out-of-domain datasets(Table IV). This indicates that the fine-tuning process had a more pronounced positive impact on the model's performance on the in-domain dataset compared to the out-of-domain datasets.

- Comparison among Models: The fine-tuning model demonstrated superior performance, achieving higher ROUGE and BERTScore metrics compared to the original model. This suggests that the fine-tuning process was effective in adapting the model's capabilities to the specific requirements of the summarization task, allowing it to generate more accurate and coherent summaries.
- Comparison among Datasets (In-domain vs. Out-of-domain): The model's performance on the in-domain dataset was significantly better than its performance on the out-of-domain datasets. This can be attributed to the fact that the in-domain dataset provided the model with training data that closely matched the characteristics and distribution of the target task, enabling it to learn the relevant patterns and features more effectively. In contrast, the out-of-domain datasets introduced additional challenges, as the model had to generalize its understanding to data that was not as well-aligned with the summarization task.
- Comparison among Out-of-domain Datasets: The PEFT (Parameter-Efficient Fine-Tuning) model 1 achieved better scores on the out-of-domain datasets compared to the other models. This observation suggests that the format and characteristics of the out-of-domain dataset 1 were more similar to the in-domain dataset, allowing the PEFT model 1 to leverage its learning and perform better on this particular out-of-domain task. The similarity in data format and distribution between the in-domain and out-of-domain dataset 1 appears to have played a significant role in the PEFT model 1's superior performance.
- GPT-4 Score as a Reliable Metric: The GPT-4 score has a positive correlation with the ROUGE and BERTScore metrics. This indicates that the GPT-4 score can effectively capture the differences between the original model and the fine-tuning models, and can be considered a reliable metric for evaluating the quality of the generated summaries. The GPT-4 score provides a holistic assessment that aligns with the more established ROUGE and BERTScore metrics, reinforcing its usefulness as a complementary evaluation tool.

B. Future Work

The experiments conducted in this study have revealed some intriguing findings regarding the performance of different fine-tuning models on in-domain and out-of-domain datasets. The most notable observation is the discrepancy between the training loss and the actual evaluation metrics. Specifically, the PEFT model2 exhibited a lower training loss compared to the PEFT model1, yet the latter achieved better scores across almost all the evaluation metrics. This suggests that

	Relevance	Coherence	Consistency	Fluency
in-domain dataset				
base	3.78	3.66	4.60	2.84
Llama2 7b	3.26	3.04	4.48	2.20
Llama2 7b[1]	**3.70**	**3.26**	**4.82**	**2.46**
Llama2 7b[2]	3.60	3.0	4.50	2.42
out-of-domain dataset[1]				
base	2.23	2.70	3.47	2.96
Llama2 7b	3.48	2.96	**4.90**	2.24
Llama2 7b[1]	**3.52**	**3.14**	4.24	**2.48**
Llama2 7b[2]	3.12	2.78	4.36	2.22
out-of-domain dataset[2]				
base	3.83	4.38	4.48	2.78
Llama2 7b	2.31	**2.80**	**4.71**	2.22
Llama2 7b[1]	**2.58**	2.30	3.64	**2.46**
Llama2 7b[2]	1.64	1.62	2.36	1.80

TABLE III: Average GPT4-score of baseline(manual summary) and LLMs on three datasets

	Relevance	Coherence	Consistency	Fluency
in-domain dataset				
Llama2 7b[1]	1.13	1.07	1.07	1.12
Llama2 7b[2]	1.10	0.99	1.0	1.1
out-of-domain dataset[1]				
Llama2 7b[1]	1.01	1.06	0.87	1.11
Llama2 7b[2]	0.90	0.94	0.89	0.99
out-of-domain dataset[2]				
Llama2 7b[1]	1.12	0.82	0.77	1.11
Llama2 7b[2]	0.71	0.58	0.50	0.81

TABLE IV: The ratio between average GPT4-score of fine-tuning models and original model on three datasets

the training loss may not be a reliable proxy for the true performance of the fine-tuned models, and that the evaluation metrics provide a more accurate assessment of the models' capabilities.

Furthermore, the study has highlighted the challenge of transferring the model's capabilities from the in-domain dataset to the out-of-domain datasets. The performance on the out-of-domain datasets was generally not as strong as the results obtained on the in-domain dataset, indicating that the fine-tuning process may have been overly specialized to the training data and struggled to generalize to more diverse or dissimilar input distributions.

To further investigate these findings and address the observed challenges, the following future work is proposed:

- Test a wider range of fine-tuning models: Expanding the exploration to include more diverse fine-tuning approaches, such as different architectures, training strategies, or hyperparameter configurations, could uncover more regularities and shed light on the underlying reasons for the observed performance patterns.
- Improve the prompt design for the evaluation metrics: The current prompts used for the GPT-4 and other evaluation metrics may not be optimally capturing the nuances of the task at hand. Exploring different prompt formulations and investigating their impact on the metric scores could lead to more reliable and informative assessments.
- Investigate techniques for leveraging both domain-specific and out-of-domain knowledge: Developing methods that can effectively combine the strengths of the in-domain and out-of-domain data sources could potentially lead to models with improved generalization capabilities, better able to handle both familiar and unfamiliar input distributions.
- Experiment with more advanced language models, such as LLaMA-3: Exploring the performance of newer and potentially more capable language models, like LLaMA-3, could provide valuable insights and potentially lead to further advancements in the fine-tuning approaches.

REFERENCES

[1] Basyal L, Sanghvi M. Text summarization using large language models: a comparative study of MPT-7B-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT models[J]. arXiv preprint arXiv:2310.10449, 2023.

[2] Goyal T, Li J J, Durrett G. News summarization and evaluation in the era of gpt-3[J]. arXiv preprint arXiv:2209.12356, 2022.

[3] Zhang T, Ladhak F, Durmus E, et al. Benchmarking large language models for news summarization[J]. Transactions of the Association for Computational Linguistics, 2024, 12: 39-57.

[4] Gehrmann S, Deng Y, Rush A M. Bottom-up abstractive summarization[J]. arXiv preprint arXiv:1808.10792, 2018.

[5] Paulus R, Xiong C, Socher R. A deep reinforced model for abstractive summarization[J]. arXiv preprint arXiv:1705.04304, 2017.

[6] Lewis M, Liu Y, Goyal N, et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension[J]. arXiv preprint arXiv:1910.13461, 2019.

[7] Touvron H, Martin L, Stone K, et al. Llama 2: Open foundation and fine-tuned chat models[J]. arXiv preprint arXiv:2307.09288, 2023.

[8] Narayan S, Cohen S B, Lapata M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization[J]. arXiv preprint arXiv:1808.08745, 2018.

[9] Hermann K M, Kocisky T, Grefenstette E, et al. Teaching machines to read and comprehend[J]. Advances in neural information processing systems, 2015, 28.

[10] Nallapati R, Zhou B, Gulcehre C, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond[J]. arXiv preprint arXiv:1602.06023, 2016.

[11] Chen Y, Liu Y, Chen L, et al. DialogSum: A real-life scenario dialogue summarization dataset[J]. arXiv preprint

[12] LoRA: Low-Rank Adaptation of Large Language Models-sarXiv:2105.06762, 2021.

[13] Zhou C, Liu P, Xu P, et al. Lima: Less is more for alignment[J]. Advances in Neural Information Processing Systems, 2024, 36.

[14] G-EVAL: NLG Evaluation using GPT-4 with Better Human Alignment

[15] https://cookbook.openai.com/examples/evaluation/how_to_eval_abstractive_summarization

APPENDIX : MEETING MINUTES

C. Minutes of 1st Project Meeting

Date: June 10, 2024
Time: 8:00 PM
Place: Zoom
Present: Prof. Zhang, LI Xiaoting

1) Minutes of Meetings: Introduction of the project demands

2) Discussion Items: Share the details of the project and assign different ideas to choose.

3) Goals for the next meeting: Confirmation on the specific implementation process of the project.

4) Next meeting: Date: June 28, 2024

D. Minutes of 2nd Project Meeting

Date: June 28, 2024
Time: 6:10 PM
Place: Zoom
Present: Prof. Zhang, LI Xiaoting

1) Minutes of Meetings: Project Plan decided.

2) Discussion Items: Clarification of the project objective & scope and the project initiation.

3) Goals for the next meeting: Solving problems encountered in the project

4) Next meeting: July 23, 2024

E. Minutes of 3rd Project Meeting

Date: July 23, 2024
Time: 3:20 PM
Place: Zoom
Present: Prof. Zhang, LI Xiaoting

1) Minutes of Meetings: Problems sharing and solving.

2) Discussion Items: Discussion on problems and give some advice.

3) Goals for the next meeting: Final project presentation.

4) Next meeting: Date: August 8, 2024

F. Minutes of 4th Project Meeting

Date: August 8, 2024
Time: 7:30 PM
Place: Zoom
Present: Prof. Zhang, LI Xiaoting

1) Minutes of Meetings: Project report to be submitted by 24th January, 2024.

2) Discussion Items: Final project presentation.

3) Goals for the next meeting: No goals for next meeting.

4) Next meeting: Date: No more meetings.