Fine-Tuning Transformers for Text Completion using Parameter-Efficient Techniques

Aliza Momysheva, Dulat Rakhymkul
School of Engineering and Digital Sciences
Nazarbayev University
Email: aliza.momysheva@nu.edu.kz, dulat.rakhymkul@nu.edu.kz

***Abstract*—The following report is intended to present our approach to fine-tuning large language models (LLMs) for the task of sentence completion in specific literary contexts from a set of books derived from a dataset of Gutenberg corpus. We leveraged parameter-efficient fine-tuning (PEFT) methods for efficient training of large models with limited computational powers, specifically Low-Rank Adaptation (LoRA) technique was applied. Our work involves systematic preprocessing of unstructured literary text from a random set of books, implementation of causal language modeling objectives using large language models, and comparative analysis across multiple transformer-based architectures with different sizes of total parameters. This report uses both quantitative metrics and qualitative assessments to evaluate the performance and discuss the implications of the findings for creative AI-assisted writing tools.**

I. Introduction

Nowadays, LLMs get extensive attention and usage by people throughout the world for a variety of purposes. Modern LLMs have demonstrated proficiency in general-purpose language understanding and text generation. However, most pretrained models are optimized on various resources from the internet without a proper structure, which typically lacks the nuances style, vocabulary, and tone characteristic of literary texts. This poses a lot of limitations for tasks requiring stylistic consistency or creative text generation for completion of sentences for specific purposes of writing with literary context.

The primary objective of this project was to adapt LLMs for literary sentence completion by fine-tuning them on a rigorously preprocessed subset of books from the popular open-access Gutenberg corpus. We aimed to generate coherent and contextually appropriate continuations given a partial sentence, with an emphasis on literary tone and fluency. The final product allows us to generate several additional words for not complete sentences in a specific style from the books. The models fine-tuned within the project are GPT-Neo, GPT-2, DistilGPT-2 and Distilled Deepseek-R1.

A. Motivation

Our motivations were both practical and research-driven:

- To explore the feasibility of using LoRA-based PEFT for training on stylistically rich data within limited computational environments mostly related to open-access online notebooks with free GPU usage.
- To create a sentence completion system that could support writers by suggesting stylistically consistent continuations depending on the set of training books.
- To evaluate transformer model performance not just in generic generation tasks, but in the stylistic adaptation domain.

II. Background and Related Work

While the most known large models like GPT-3 and GPT-4 became synonymous with generative NLP tasks, they remain very expensive in terms of computational power for fine-tuning. Therefore, in our prior research that was our initial inspiration was studies that attempted to adapt LLMs for creative domains using standard fine-tuning or prompt engineering, but often lacked rigorous evaluation or scalability. One of them was an open-source GitHub repository [1] that attempted to train the GPT-2 model on Shakespeare and Plato but did not report any essential metrics other than perplexity, which led to unstructured and unjustified evaluation of outputs.

Our work expands upon such efforts by incorporating rigorous training-evaluation pipelines, multiple baseline comparisons, and controlled use of PEFT.

In addition, while selecting training methods we discovered a recent analysis by Anyscale [2] investigated the performance trade-offs between LoRA and full-parameter fine-tuning using LLaMA models on various tasks. Figure 1 from their blog post illustrates how LoRA can match or approach full fine-tuning performance in accuracy on SQL test sets across model sizes (7B and 13B), with significantly fewer trainable parameters. This was the main reason to go fully into LoRA-based fine-tuning in this project.

III. Dataset and Preprocessing

A. Data Collection and Loading

Our initial source of data was a Gutenberg Corpus as it was an open-source web page with a variety of books [3]. For consistency of the training, we used the English portion of the Gutenberg dataset from the HuggingFace hub using the `manu/project_gutenberg` dataset [4]. Out of 70,000 total books, only 200 books were processed and stored as raw text, due to limitations in computational resources for training models. Intermediate text files were written in chunks to prevent memory overflow during the processing stage.

B. Cleaning and Preprocessing Pipeline

We designed a multi-step pipeline in Python that:

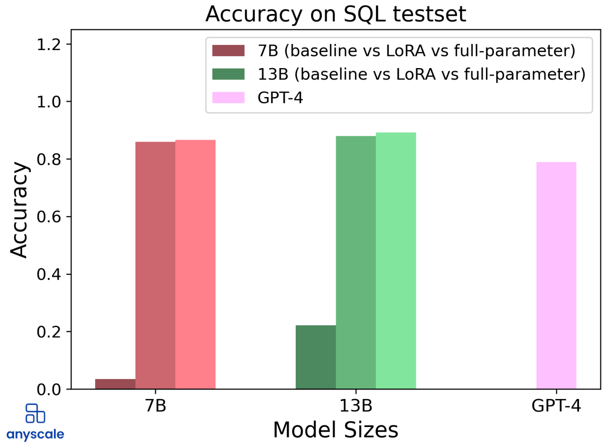

Fig. 1. Comparison of LoRA and full-parameter fine-tuning accuracy across different model sizes on SQL test sets. Source: Anyscale [2].

- Removed Project Gutenberg headers/footers, license texts, and metadata (title, author, etc.).
- Filtered out URLs, page numbers, and uppercase headings.
- Used NLTK to tokenize the dataset into individual sentences.
- Discarded sentences that were too short (¡5 words), too long (¿100 words), or lacked syntactic coherence.
- Removed special characters but kept basic punctuation and letters.

We also applied additional rules to remove incorrectly joined lines and preserve sentences split across quotations. The cleaned sentences were saved to a final file for model input with lines consisting of individual sentences from the dataset.

C. *Dataset Structure*

After filtering and cleaning:

- Total sentences after preprocessing: 140 257
- Saved as newline-separated entries in ‘.txt‘ format.
- Train/test splits were saved and imported using Hugging-Face’s ‘text‘ loader.
- Training Sentences: 136 750
- Testing Sentences: 3 507
- Final datasets were filtered again for minimum text length (10 words).

IV. Model Architecture and Training

A. *Models Used*

We fine-tuned the following transformer models from Hugging-Face [5]:

- GPT-2 (137M parameters)
- GPT-Neo (125M parameters)
- DistilGPT-2 (88.2M parameters)
- DeepSeek-R1-Distill (1.7B parameters)

B. *Training Objective*

We adopted a causal language modeling (CLM) setup, where the model predicts the next token given a sequence of previous tokens. Each sentence was tokenized using Hugging Face’s AutoTokenizer and formatted for CLM training.

C. *Fine-Tuning Strategy*

Given the computational constraints of using Google Colab and Kaggle notebooks, full fine-tuning was often infeasible. We, therefore, employed LoRA via the PEFT library [6], allowing us to:

- Freeze base model weights
- Insert trainable low-rank adapters into attention layers
- Significantly reduce memory and time costs

However, some of the models were fully finetuned as we had some limited availability to powerful GPUs by the end of the project. We used this opportunity for better comparisons and deep discussion of results.

LoRA hyperparameters were chosen rigorously on the Hugging Face models library, specifically, we accumulated already fine-tuned models with LoRA and derived average parameters for the most efficient training.

LoRA hyperparameters:

- Rank (r): 8
- LoRA Alpha: 32
- Dropout: 0.1
- Target modules: Model-specific (e.g., attention layer and feed forward layer)

Training hyperparameters:

- Number of training epochs: 5
- Batch size: 16 (due to GPU memory constraints)
- Model specific learning rate: 5e-4 and 2e-4 for Distilled Deepseek

V. Evaluation Metrics

A. *Quantitative Metrics*

We utilized the following metrics:

- **Perplexity (PPL)**: Measures model uncertainty. Lower is better.

$$\text{Perplexity} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i \mid w_1, w_2, \ldots, w_{i-1})\right) \tag{1}$$

Where:

- N is the total number of tokens in the sequence,
- w_i is the i-th token,
- $P(w_i \mid w_1, \ldots, w_{i-1})$ is the probability of token w_i given the previous tokens.

- **BERTScore**: Measures semantic similarity between generated and reference text. We report Precision, Recall, and F1.

$$\text{Precision} = \frac{1}{|T|} \sum_{t \in T} \max_{r \in R} \cos(\mathbf{e}_t, \mathbf{e}_r) \tag{2}$$

$$\text{Recall} = \frac{1}{|R|} \sum_{r \in R} \max_{t \in T} \cos(\mathbf{e}_r, \mathbf{e}_t) \tag{3}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Where:

- T is the set of tokens in the generated (candidate) sentence,
- R is the set of tokens in the reference sentence,
- $\mathbf{e}_t$ and $\mathbf{e}_r$ are the contextual embeddings of token $t \in T$ and $r \in R$, respectively,
- $\cos(\cdot, \cdot)$ is the cosine similarity between two embeddings.

B. Qualitative Evaluation

Generated continuations were manually analyzed for:

- Stylistic coherence
- Grammatical correctness
- Semantic relevance

VI. Results and Discussion

A. Quantitative Results

TABLE I
GPT-Neo Fine-Tuning results

Model	PPL	BERT Precision, Recall, F1
GPT-Neo Pretrained	101.84	0.39 0.50 0.43
GPT-Neo LoRA	42,28	0.41 0.54 0.46
GPT-Neo Full-Finetuning	13.70	0.52 0.70 0.59

Table I shows that fine-tuning GPT-Neo with LoRA significantly reduces perplexity from 101.84 to 42.28 and also it improves all BERTScore components. Full fine-tuning provides the best results overall, with the lowest perplexity (13.70) and highest semantic similarity (BERT F1 = 0.59) which demonstrates the effectiveness of full parameter updates in capturing literary style. However, the example text generated by a fully finetuned model is a bit difficult to read which is an indicator of overfitting of the model.

TABLE II
GPT-2 Fine-Tuning results

Model	PPL	BERT Precision, Recall, F1
GPT-2 Pretrained	92.27	0.46 0.55 0.49
GPT-2 LoRA	41.94	0.46 0.57 0.50
GPT-2 Full-Finetuning	13.44	0.48 0.61 0.53

Table II demonstrates that LoRA fine-tuning nearly halves the perplexity of the pretrained GPT-2 model, from 92.27 to 41.94, with slight gains in BERTScore metrics. Full fine-tuning again shows the highest performance for this model so it achieves the lowest perplexity (13.44) and the best semantic match (BERT F1 = 0.53). It highlights the consistent benefits of full adaptation over LoRA in literary domains. However, again as with GPT-Neo, the generated text gives hints on overfitting.

The following models were fine-tuned only using LoRA.

TABLE III
DistilGPT-2 Fine-Tuning results

Model	PPL	BERT Precision, Recall, F1
DistilGPT-2 Pretrained	155.60	0.46 0.54 0.49
DistilGPT-2 LoRA	49.21	0.46 0.56 0.50

Table III shows a pattern consistent with the GPT-2 and GPT-Neo results. So, LoRA fine-tuning significantly reduces perplexity, in this case from 155.60 to 49.21. Also, it shows improvements in BERTScore metrics (F1 increases from 0.49 to 0.50). Similar to earlier models, full fine-tuning would likely outperform LoRA, but even lightweight adaptation demonstrates substantial gains, reinforcing the effectiveness of LoRA across architectures.

TABLE IV
DeepSeek-R1-Distill Fine-Tuning results

Model	PPL	BERT Precision, Recall, F1
DeepSeek-R1-Distill Pretrained	516.89	0.40 0.51 0.45
DeepSeek-R1-Distill LoRA	55.23	0.47 0.57 0.51

Table IV reveals that LoRA fine-tuning significantly enhances the performance of the large DeepSeek-R1-Distill model, reducing perplexity from 516.89 to 55.23 and improving BERT F1 from 0.45 to 0.51. This model shows the most evident improvements from fine-tuning as it initially has a large perplexity score. These improvements align with those seen in GPT-2 and DistilGPT-2, further confirming that LoRA remains effective even when applied to substantially larger models. It maintains consistency in both perplexity reduction and semantic coherence gains.

B. Qualitative Examples

Table V shows qualitative comparisons of text generations from pre-trained and fine-tuned models. It clearly shows that fine-tuning leads to improvements in the stylistic choices of models, also sentences become more consistent and coherent. Specifically, the completions from pre-trained models are often erratic or off-topic, whereas LoRA and full fine-tuning produce outputs that are more fluent, stylistically aligned, and semantically consistent with the original intent. Across all models, full fine-tuning yielded the most coherent results, but LoRA was able to offer strong improvements with significantly fewer parameters. However, we observed that fully fine-tuned models sometimes may generate strange unknown words which are most possibly learned from the training dataset of books as we did not remove rare words from there. So, it indicates some overfitting during the training.

C. Interpretation

While full fine-tuning yielded the best results across models, LoRA-based fine-tuning demonstrated impressive performance

TABLE V
EXAMPLE COMPLETIONS FROM PRETRAINED AND FINE-TUNED MODELS

Prompt	Reference	Pretrained Output	Fine-Tuned Output (LoRA / Full)
When inquisitors punish heretics it is not with the desire to	*destroy them, but that they shall be converted and live.*	**GPT-Neo:** please, but with the desire to do evil. **GPT-2:** have their minds blown over... not allowed to tell the truth. **DistilGPT-2:** be a Christian, but with the desire to be **DeepSeek:** kill her, but with the desire to ensure that	**GPT-Neo LoRA**: see them punished. **Full**: destroy them, but that... **GPT-2 LoRA**: be punished by the Inquisition **Full**: remove them. **DistilGPT-2 LoRA**: see, but by a fear of persecution **DeepSeek LoRA**: find a solution, but for their own interest
The most personally courageous become bullies and	*the terror of the community.*	**GPT-Neo:** bullies again. That's what the NZ Police are about. **GPT-2:** this is how we are able to combat them. **DistilGPT-2:** bullies and bullies." **DeepSeek:** ...and the most... become bullies. The most...	**GPT-Neo LoRA**: bullies and bullies. The more one... **Full**: rule is to maintain **GPT-2 LoRA**: and, when the time is... **Full**: make victims of it **DistilGPT-2 LoRA**: and thieves. They would never forget... **DeepSeek LoRA**: and the most dangerous are the soldiers...

given its resource efficiency. GPT-Neo's performance highlights its suitability for creative tasks when fully trained.

Across both GPT-Neo and GPT-2, LoRA fine-tuning reduced perplexity by over 50% compared to the pretrained baseline (e.g., from 101.84 to 42.28 in GPT-Neo), with moderate improvements in BERTScore metrics. These results confirm that LoRA can significantly enhance a model's contextual understanding and fluency without incurring the full cost of updating all parameters. This makes it a practical choice for lightweight deployment and experimentation in low-resource environments.

While full fine-tuning achieved the best overall performance (lowest perplexity and highest semantic similarity), qualitative inspection revealed occasional issues such as verbosity, repetition, and unnatural phrasing. These artifacts suggest mild overfitting, likely due to limited dataset size or domain-specific over-adaptation. In contrast, LoRA outputs remained more restrained, even if slightly less expressive.

GPT-Neo benefited more from full fine-tuning than GPT-2 in both perplexity and BERTScore gains (e.g., +0.16 F1 vs. +0.04), suggesting that GPT-Neo may have a better capacity for capturing the richer structure and longer-range dependencies typical in literary prose. However, this also came with a greater risk of overfitting, as noted in qualitative analysis.

VII. CHALLENGES AND LIMITATIONS

- **Computational Bottlenecks**: Reliance on cloud notebooks limited training duration and model size.
- **Metric Suitability**: BLEU score could not be applied to creative text generation as it is too strict and dedicated to machine translation tasks; BERTScore inflated due to partial prompt-references.
- **Data Noise**: Even after preprocessing, artifacts in literary texts occasionally misled the models.

VIII. CONCLUSION

This project demonstrates that parameter-efficient fine-tuning, specifically using LoRA, can significantly improve sentence completion in literary contexts even with limited computational resources. Across all models, LoRA fine-tuning reduced perplexity and improved semantic coherence, often approaching the quality of full fine-tuning.

Our results show that LoRA offers a practical way to adapt language models for stylistically-aware applications such as writing assistants or literary tools. This work opens up further opportunities to expand into multilingual datasets, more complex prompts, or interactive user-guided generation.

REFERENCES

[1] P. Reddy, "WordSense-NLP-Auto-Suggestion-Prediction," GitHub repository, 2023. [Online]. Available: https://github.com/pareekshitreddy/WordSense-NLP-Auto-Suggestion-Prediction

[2] Anyscale, ”Fine-Tuning LLMs: LoRA or Full-Parameter? An In-Depth Analysis with LLaMA 2,” 2023. [Online]. Available: https://www.anyscale.com/blog/fine-tuning-llms-lora-or-full-parameter-an-in-depth-analysis-with-llama-2

[3] Project Gutenberg, https://www.gutenberg.org/

[4] M. Faysse, "Project Gutenberg Dataset," Hugging Face, 2023. [Online]. Available: https://huggingface.co/datasets/manu/project_gutenberg

[5] T. Wolf et al., ”Transformers: State-of-the-Art Natural Language Processing,” *EMNLP 2020.*

[6] E. Hu et al., ”LoRA: Low-Rank Adaptation of Large Language Models,” *arXiv:2106.09685*, 2021.