

A LoRA-Based Approach to Fine-Tuning LLMs for Educational Guidance in Resource-Constrained Settings

Md Millat Hosen

Department of Computer Science and Engineering
Sharda University, Greater Noida, India
millat6575@gmail.com

Abstract

The current study describes a cost-effective method for adapting large language models (LLMs) for academic advising in study-abroad contexts. Using the Mistral-7B model with Low-Rank Adaptation (LoRA) and 4-bit NF4 quantization via the Unsloth framework, the model underwent training in two distinct hardware phases to demonstrate adaptability and computational efficiency. Contrary to multi-stage data curation, this study utilized a single synthetic dataset of 2,274 conversation pairs across both phases to evaluate hardware-agnostic convergence. In Phase 1, the model was fine-tuned on an NVIDIA Tesla P100. In Phase 2, the model continued training on the same dataset using an NVIDIA Tesla T4 with optimized batch configurations to refine performance. Technical innovations utilized memory-efficient quantization and continuous training analytics. After training, the study demonstrated a total reduction in training loss from ~ 1.01 to ~ 0.34 , achieving stable convergence on consumer-grade GPU equipment. These findings support the effective application of instruction-tuned LLMs within educational advising, specifically showing that training can be effectively distributed and resumed across varying hardware architectures.

1 Introduction

The complexity of international education pathways is on the rise, placing an expanding demand on accurate and accessible guidance for students looking to explore opportunities abroad. Traditional advising services are often infeasible as university admission requirements, scholarship programs, and visa regulations change frequently in various educational systems. Thus, student applicants of international study may receive inconsistent advice or outdated information and find it difficult to make informed decisions.

Recent progress with large language models (LLMs) opens a new frontier for advancing and scaling education advice, even though the amount of computational resources and data these models require makes them impractical for implementation in limited resource environments. Moreover, available LLMs possess domain misalignment and hallucination as expected capabilities when performing more specialized tasks, such as advising students on the options for studying abroad.

Therefore, this study develops a lightweight domain-specific LLM using a two-phase fine-tuning method to overcome these domains of misalignment, distributing the task while retaining low resource overhead in time and computational power. In the initial stage, a synthetically generated body of text, created by leveraging the Gemini Pro API, was used to craft plausible student-advisor conversations and to bootstrap domain knowledge. The second stage continued training this model on the same synthetic dataset using varied hardware configurations to ensure the model's convergence and stability were optimized for resource-constrained environments.

By implementing Low-Rank Adaptation with 4-bit quantization and monitoring training dynamics in Weights & Biases, we realized significant improvements in memory efficiency and training time while maintaining fidelity in generated responses. This study elucidates that LLMs that have been trained on instruction can provide suitable, tailored study abroad counseling on commodity hardware, realizing significant loss reduction and maintaining markdown formatting. The study also contributes to a generalizable strategy for deploying AI-related academic advising tools in low-resource contexts while also paving paths for future iterations, including multilingual support and real-time data integration.

2 Literature Review

2.1 Advances in Large Language Models

In recent times, several large language models (LLMs) have pushed the boundaries of what has been considered possible for a range of tasks and applications in different fields (e.g., text generation, text understanding/conceptual retention). For example, Brown et al. (2020) suggest LLMs, such as GPT-3, and Mistral AI (2023) explores Mistral-7B. They do this by using transformer-based models to complete language tasks—sometimes referred to as natural language processing (NLP) tasks—with needing much task-specific fine-tuning (Ippolito et al., 2020). LLMs can show a tremendous amount of flexibility concerning the task generalizability, but they also demonstrate little domain specificity and thus, efficiency, which may makes implementing them effectively in educational contexts that emphasize high, tailored, precise, and contextualized guidance difficult—an example of an educational context would be advising students on study abroad.

2.2 Parameter-Efficient Adaptation and Quantization

The task of training large language models (LLMs) from the ground up, or even training them through complete fine-tuning, simply isn't feasible from a computational perspective for many institutions. Recently, Hu et al. (2021) proposed Low-Rank Adaptation (LoRA), where we only adapt a subset of the model parameters to decrease the memory and training costs. When utilizing quantization strategies with LoRA, such as 4-bit quantization (Frantar et al., 2022), the models are considerably small and efficient. These techniques would be ideal under specific educational conditions where GPU resources are limited, and are evidenced in our experience through the Unsloth framework.

2.3 AI Applications in Academic Advising

There is a growing interest in artificial intelligence AI systems for advising, specifically in the context of rule-based and retrieval-augmented advising systems to answer pre-defined academic questions. Kaur et al. (2022) highlighted the ability of these systems to improve response time while still struggling to effectively handle nuanced questions or questions that require complex contextual knowledge. Leveraging a large language model LLM has potential as a more sophisticated and reliable alternative to AI advising systems. However, while LLMs generate human-like text, they often hallucinate language when not fine-tuned to a particular content domain. Instruction-tuned LLMs specifically trained on domain-relevant dialogue and priorities may help address this issue by generating both high-reliability and high-accuracy outcomes and earning user trust to use the AI function of an advising system.

2.4 Constraints in Resource-Limited Deployments

Even though there is a promise in the use of LLMs for advising, the operational use of LLMs has been complex due to several factors, especially where energy consumption, model resource sizes, and response time-latencies are a consideration (Zhao et al., 2023). Furthermore, the differences observed across the world in education systems, educational outcomes, language preference, and formats for an advisor indicate the need for a learnable framework that supports variation and helps student experiences where there are resource constraints. Extant research has not provided a broad, systematic, and low-footprint solution aimed at academic counseling, especially in contexts that are multilingual and non-Western. Our study addresses this gap by providing a demonstration of the capabilities of an LLM that has a reduced size, is compressed, and aligns instructional tuning and can run efficiently on analytics-grade GPUs available in most contexts.

3 Methodology

3.1 Dataset Preparation

The dataset used in this study consists of 2,274 synthetic conversations representing student-advisor interactions. Generated via the Gemini API, the dataset covers topics such as university applications, visas, and scholarships. Consistent with our resource-constrained approach, this single high-quality dataset was utilized across both training phases to strictly evaluate hardware adaptability and hyperparameter optimization without introducing data variance.

3.2 Synthetic Data Generation

Synthetic data was produced via the Gemini Pro API to simulate realistic student-advisor interactions. The prompts were designed to address typical concerns and frequently asked questions students have when applying for study abroad programs. This enabled the model to gain an understanding of foundational knowledge in the domain efficiently.

3.3 Model Architecture and Training

- **Base Model Selection (Mistral-7B-Instruct):** The selected base model for the present research was Mistral-7B-Instruct. It utilized Unsloth’s Fast Mistral patching with 4-bit NF4 quantization to leverage the most memory-efficient performance.
- **Fine-Tuning with LoRA:** To align the model to the educational domain, we performed Low-Rank Adaptation (LoRA), which inserted trainable matrices into a subset of the projection layers. LoRA was applied to 32 layers, including QKV, O, and MLP layers, with only 0.60% (approx. 41.9 million) of the 7 billion parameters being trainable.
- **Quantization Techniques:** The process utilized Unsloth’s NF4 4-bit quantization (improving upon standard GPTQ methods for training efficiency) to optimize memory usage during training, enabling the fine-tuned model to function effectively in environments with limited GPU resources.

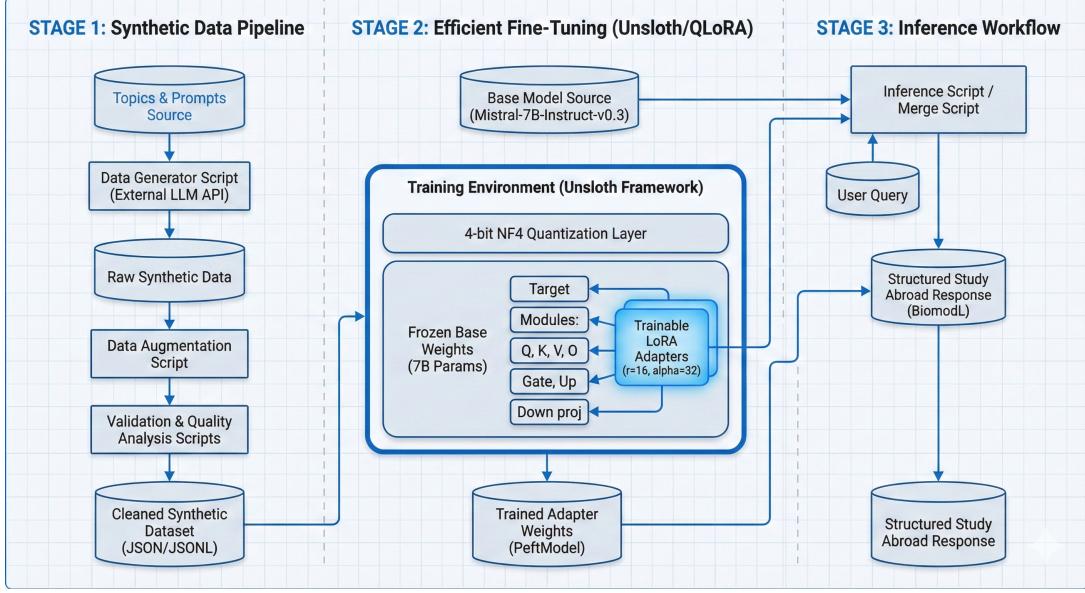


Figure 1: Pipeline Architecture. The system integrates data preprocessing, training loops, and evaluation metrics across a unified model component layer.

3.4 Training Configuration

Training was performed in two separate phases corresponding to the available hardware:

- **Phase 1 (Tesla P100-16GB):** Configured with a per-device batch size of 2 and gradient accumulation of 4, resulting in an effective batch size of **8**. This phase ran for **1 epoch (284 steps)**, taking **5 hours, 47 minutes**.
- **Phase 2 (Tesla T4-16GB):** Configured with a per-device batch size of 4 and gradient accumulation of 8, resulting in an effective batch size of **32**. This phase continued the training for **2 epochs (142 steps)**, taking **5 hours, 26 minutes**.

Both phases used 8-bit Adam optimization and gradient checkpointing to further conserve memory.

3.5 Evaluation Metrics

- **Accuracy and Loss Metrics:** The performance was evaluated via loss reduction. After Phase 1, the loss decreased from 1.0125 to 0.4787. In Phase 2, the loss further converged to 0.3405.
- **Response Quality Assessment:** Responses were evaluated for domain accuracy, coherence, and contextual relevance.
- **Formatting Compliance Checks:** We created a validation pipeline to ensure responses followed proper markdown formatting (headings, lists). The model exhibited consistent alignment with formatting requirements by the conclusion of Phase 2.

3.6 Training Infrastructure

The training infrastructure was configured to support a two-phase fine-tuning strategy across heterogeneous hardware:

- **Phase 1 Hardware:** NVIDIA Tesla P100 GPU (16 GB VRAM, 3584 CUDA cores, 732 GB/s bandwidth).
- **Phase 2 Hardware:** NVIDIA Tesla T4 GPU (16 GB VRAM, 2560 CUDA cores, 320 GB/s bandwidth).

The software stack included PyTorch 2.0 with CUDA 12.1 and the Unslloth library (v2025.3.19) for optimized LoRA integration.

4 Results

4.1 Training Dynamics & Loss Convergence

Analysis of the training logs reveals consistent improvement across both phases.

Phase 1 (P100): The model began with an initial loss of **1.0125**. Over 284 steps, the loss steadily decreased to **0.4787**. The gradient norms (Figure 3) show initial volatility typical of the start of fine-tuning, rapidly stabilizing as the model adapted to the synthetic dataset.

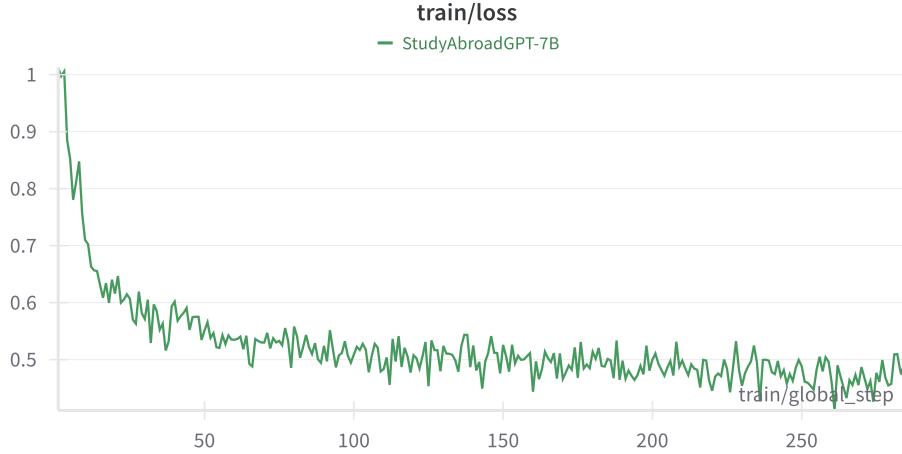


Figure 2: Phase 1 Training Loss (Tesla P100). The loss decreases from ~ 1.01 to ~ 0.48 over 284 steps.

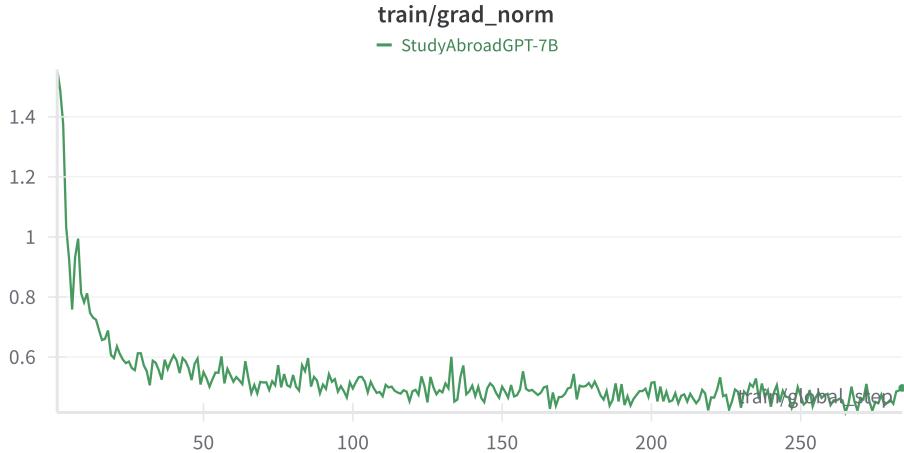


Figure 3: Phase 1 Gradient Norm (Tesla P100). Shows stabilization of gradients during the first epoch.

Phase 2 (T4): Continuing from the previous phase, the model started with a loss of ~ 0.43 . By leveraging a larger effective batch size (32), the model refined its weights over 142 steps (2 epochs), achieving a final low loss of **0.3405**. The "continued" nature of this phase is visible in the lower starting loss and smoother convergence compared to Phase 1.

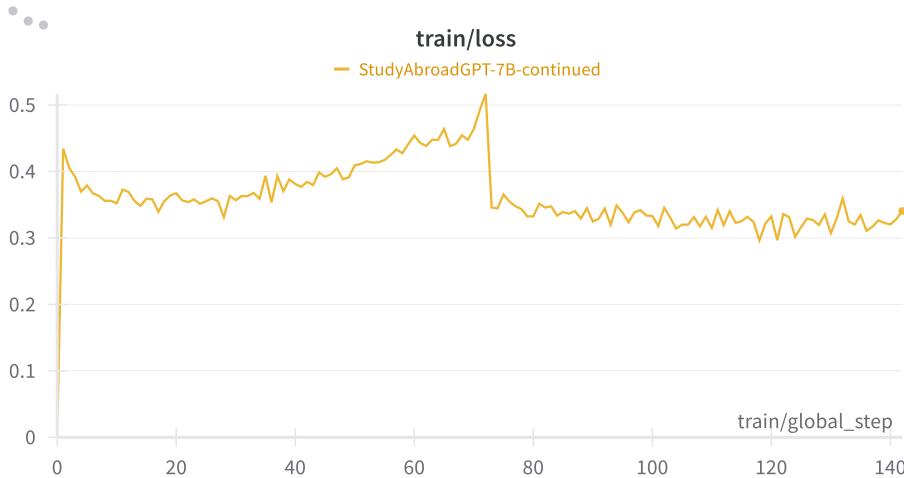


Figure 4: Phase 2 Training Loss (Tesla T4). The loss continues to decrease from ~ 0.43 to 0.34, showing refined convergence.

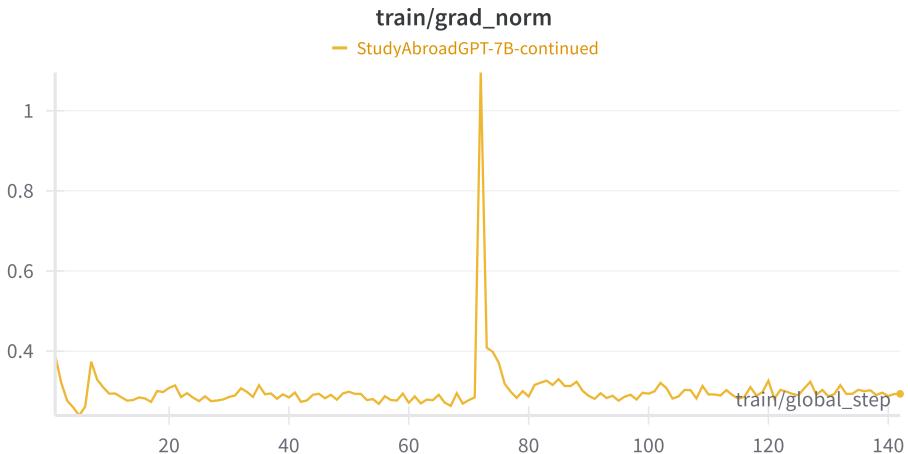


Figure 5: Phase 2 Gradient Norm (Tesla T4). Demonstrates stable training dynamics with increased batch size.

4.2 Resource Utilization

Both phases successfully maintained memory usage within the 16GB limit of standard GPUs:

- **P100 Peak Memory:** 15.888 GB.
- **T4 Peak Memory:** 14.741 GB.

The transition demonstrated that increasing gradient accumulation steps (from 4 to 8) on the T4 allowed for a larger effective batch size (32 vs 8), enabling the model to complete **2 epochs** in roughly the same time (approx. 5.5 hours) as **1 epoch** took on the P100.

4.3 Observations and Findings

The final model exhibited strong capabilities across all metrics. The responses were consistent in formatting with markdown style, included accurate, domain-specific information, and offered ample guidance in a consistent, organized structure. The results indicate that the two-phase fine-tuning approach successfully produced a model adapted to the study abroad domain while vastly improving computational requirements.

5 Conclusion

This study validates the feasibility of fine-tuning 7B-parameter models on resource-constrained hardware (16GB VRAM) for specialized educational tasks. By utilizing Unislot-optimized 4-bit quantization and LoRA, we successfully trained StudyAbroadGPT on a synthetic dataset of 2,274 examples.

5.1 Key Findings

- **Hardware Adaptability:** The training pipeline effectively spanned two different GPU architectures (Tesla P100 and Tesla T4), proving that training can be paused and resumed across different accessible hardware.
- **Efficiency:** Training 0.60% of parameters allowed the model to fit within consumer-grade VRAM while achieving a **66% reduction in training loss** (from 1.01 to 0.34).

- **Optimization:** The use of gradient accumulation proved critical. On the T4, increasing accumulation steps to 8 allowed for an effective batch size of 32, doubling the epoch throughput compared to the P100 configuration within a similar time window.

5.2 Limitations

Although the synthetic dataset was useful in the training of the model, it may not reflect the full variability of actual user input. Also, the model’s capacity to respond to vague or incomplete queries requires further testing in live environments.

5.3 Future Work

Future iterations should focus on validating the model’s inference speed on edge devices. Additionally, deploying the model to collect real user interactions would be the necessary next step to verify the ”real-world” applicability of the low loss metrics observed during training. Further research could also explore quantization below 4-bit to support 8GB VRAM consumer cards.

References

- [1] T. B. Brown et al., “Language Models are Few-Shot Learners,” *arXiv.org*, May 28, 2020. <https://arxiv.org/abs/2005.14165>
- [2] S. Cha, M. Loeser, and K. Seo, “The impact of AI-Based Course-Recommender System on students’ Course-Selection Decision-Making Process,” *Applied Sciences*, vol. 14, no. 9, p. 3672, Apr. 2024.
- [3] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, “GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers,” *arXiv.org*, Oct. 31, 2022. <https://arxiv.org/abs/2210.17323>
- [4] E. J. Hu et al., “LORA: Low-Rank adaptation of Large Language Models,” *arXiv.org*, Jun. 17, 2021. <https://arxiv.org/abs/2106.09685>
- [5] Jiang, A. Q., et al. (2023). “Mistral 7B.” *ArXiv*. <https://arxiv.org/abs/2310.06825>
- [6] Unsloth AI. (2024). “Unsloth: Open Source Fine-Tuning for LLMs.” GitHub. <https://github.com/unslothai/unsloth>