LLM Fine-tuning Challenge: Enhancing Qwen 2.5 3B for AI Research QA

In this hackathon challenge, you will fine-tune the Qwen 2.5 3B model to answer questions based on recent AI research papers, blogs, and documents related to the research paper. Your goal is to create a specialized model that can accurately retrieve, interpret, and generate responses from technical AI literature.

Requirements

Core Requirements

- Generate synthetic dataset for training the LLM, split dataset appropriately.
- Fine-tune the Qwen 2.5 3B model (base or instruct version)

  https://huggingface.co/Qwen/Qwen2.5-3B-Instruct

  https://huggingface.co/Qwen/Qwen2.5-3B

- Produce a 4-bit quantized version of your model in .gguf format - Create an inference script for the quantized model.
- Thoroughly document your approach and reasoning for all decisions.
- Evaluate the final system using any framework of choice using the split test set.

Model Selection

- Use either Qwen/Qwen2.5-3B (base) or Qwen/Qwen2.5-3B-Instruct as the backbone of your system.
- For multi-agent or workflow systems: Any additional LLMs must have fewer parameters than Qwen 2.5 3B.
- **Optional**: For embedding and retrieval components: You may use any model of your choice, regardless of parameter count. You may fine-tune them if necessary, however, you cannot use embedding through API calls.
- **Optional**: Document your reasoning for each model selection decision.

Training Process

- Document all hyperparameters with justifications.
- Explain your fine-tuning methodology and any pre-training steps.
- If applicable, detail your approach to reinforcement learning.
- If applicable, describe techniques used to enhance reasoning capabilities.

Data & Dataset Creation

You should use:

- Provided research papers/documents

Additionally, you may use:

- Open-source datasets
- Custom datasets you create
- Document your data generation approach. (Using RAGAS, DeepEval, no code tools, manual creation, etc.)
- Clearly explain data preprocessing and augmentation techniques.

Optional Components (Bonus Points)

- Implement a RAG (Retrieval-Augmented Generation) system
- Choose and justify an in-memory database
- Select and document embedding model choice (can be fine-tuned)
- Optionally implement a retrieval model (e.g., ColBERT)
- Apply reinforcement learning or human alignment techniques. (eg.- DPO, ORPO, KTO ...)
- Train for reasoning ability. (ex.- GRPO ..)

Evaluation Criteria

- Final Model performance will be evaluated on a hidden evaluation dataset (created from provided documents) by the judges.
- Quality of documentation and justifications for technical choices.
- Innovativeness of approach.
- Efficiency and practical usability of the solution.
- Performance improvements over the base model.

Submission Requirements

1. The 4-bit quantized model in .gguf format

2. Complete inference script (including RAG components if used)

3. Documented training process with code

4. Technical report explaining all decisions and methodologies

5. Evaluation results if you implemented your own evaluation framework

Environment Recommendations

- Google Colab or Kaggle notebooks for training
- Recommended libraries:
  - ➢ finetuning: unsloth, axolotl, PyTorch, transformers, or keras
  - ➢ Dataset generation: RAGAS, DeepEval, Kiln-AI

Tips for Success

- Focus on efficient fine-tuning techniques suitable for the 3B parameter model.
- Consider the trade-offs between different approaches. (LoRA, QLoRA, full fine-tuning)
- Pay attention to quantization quality to preserve model capabilities.
- Carefully document your reasoning to demonstrate understanding.
- Balance complexity with practical implementation.

The final evaluation will be conducted on an unseen dataset created from the same source documents, testing your model's ability to generalize and accurately answer questions about AI research.