MEDIGUIDE: Comparative Fine-Tuning for Medical Question Answering

---

Abstract

In this project, we present MEDIGUIDE, a conversational AI chatbot fine-tuned to provide accurate, medically grounded responses to patient queries. We experiment with three fine-tuning strategies on the Falcon-7B model: Prompt Tuning, LoRA, and QLoRA. Using a subset of the MedQuAD dataset, we measure each method's performance using ROUGE, Perplexity (PPL), latency, and model size. Our results highlight the trade-offs between quality, speed, and resource efficiency, leading to a recommended deployment strategy tailored for real-world medical chatbot scenarios.

Introduction

With the growing demand for quick medical advice, AI-powered chatbots have gained significant relevance. These systems offer immediate responses to general health queries and assist in triage, while clearly disclaiming that they are not substitutes for professional diagnosis.

Goals of the Project:

- Build a Falcon-7B-based chatbot that answers user medical queries.
- Ensure answers follow clinical tone, accuracy, and include safety disclaimers.
- Fine-tune using Prompt Tuning (both quantised and full precision), LoRA, and QLoRA.
- Compare each method based on output quality, efficiency, and deployability.

Dataset & Preprocessing

Dataset Used:

- Derived from MedQuAD - Medical Question Answering Dataset, is a collection of question-answer pairs meticulously curated from 12 trusted National Institutes of Health

(NIH) websites. These websites cover a wide range of health topics, from cancer.gov to GARD (Genetic and Rare Diseases Information Resource) , which makes sure that the model follows HIPAA-equivalent standards.

- Each entry includes:
    - `question`: Medical query
    - `answer`: Expert-written answer
    - `source`: source of the query
    - focus_area: label of the query

Preprocessing Steps:

- Removed short answers and low-context examples.
- Reformatted into:
    - `<human>`: "question"
    - `<assistant>`: "`answer`"
- Split into 200 training and 50 evaluation examples (shuffled)

Fine-Tuning Methods

We fine-tuned Falcon-7B and Falcon-1B (quantised models were trained with falcon-7B and full precision models were trained using falcon-1B due to lack of resources, but in turn were run for more epochs) using three parameter-efficient methods on identical training data:

1. Prompt Tuning

- Injected virtual tokens before the question.
- Only trained prompt embeddings (tiny size <1MB).
- Model weights frozen.
- Fast training, very low memory.

2. LoRA (Full-Precision)

- Injected adapter layers into attention blocks.
- Model remained in 16-bit (BF16) precision.
- Moderate training cost with better output fluency.

3. QLoRA (Quantized LoRA)

- Applied 4-bit quantization to Falcon-7B using `bitsandbytes`.
- Trained LoRA adapters using PEFT.
- Most memory-efficient, slight latency cost.

Common Libraries:

- Hugging Face Transformers, PEFT, TRL, BitsAndBytes, PyTorch, Accelerate

---

Evaluation Metrics

To evaluate and compare models, we used:

- **ROUGE (1, 2, L):** Token overlap vs. gold answer
- **Perplexity (PPL):** Model confidence in generating valid text
- **Latency:** Average inference time per question
- **Model Size:** Trainable parameters (adapter size)

Results

Method	ROUGE-1	ROUGE-2	ROUGE-L	Perplexity	Latency (s)	Adapter Size
Prompt Tuning (quantised)	0.21	0.04	0.12	5.85	8.81	~0.43 MB
Prompt Tuning (fp)	0.18	0.02	0.10	6.89	1.89	~0.20 MB
LoRA (FP)	0.21	0.04	0.12	5.31	3.53	~12 MB
QLoRA (4-bit)	0.25	0.07	0.15	3.45	10.94	~18 MB

Discussion

Prompt Tuning:

- Extremely lightweight and fast.
- Lower performance due to limited training capacity.
- Ideal for mobile or edge devices.

LoRA:

- Balances quality and latency.
- Easier to train than full finetuning.

QLoRA:

- Best perplexity (3.45)
- Slightly slower due to quantization.
- Highly memory-efficient (runs efficiently on 16GB GPU like Kaggle T4).

Trade-offs & Deployment Strategy

Comparison Summary:

- Prompt Tuning: Best for speed & memory, weakest results
- LoRA: Best latency–accuracy balance
- QLoRA: Best accuracy, acceptable latency

Recommended Deployment strategy:

1. Prompt Tuning

- **Size:** Extremely lightweight (~500KB for embeddings)
- **Speed:** Fastest inference (average ~3s per response)
- **Memory Usage:** Minimal; base model remains frozen
- **Ideal Environment:** Mobile/Edge devices, low-RAM cloud endpoints
- **Trade-offs:** Lower response accuracy and clinical fluency
- **Recommendation:** Use only when minimal memory and CPU constraints dominate

---

2. LoRA (Full Precision, 16-bit)

- **Size:** ~30MB adapter + 7B base model in FP16 (~13–16 GB VRAM)
- **Speed:** Moderate latency (~7.5s)
- **Memory Usage:** Requires T4 or higher GPU with ≥16GB VRAM
- **Ideal Environment:** Hosted GPU-backed servers (AWS, GCP, HF Inference Endpoints)
- **Trade-offs:** Balanced performance; suitable for real-time applications
- **Recommendation:** Preferred for high-volume production with reliable GPU access

---

3. QLoRA (4-bit Quantized + LoRA)

- **Size:** Same adapter (~30MB) + quantized 7B model (~5–6GB VRAM)
- **Speed:** Slightly slower (~10.9s avg latency)
- **Memory Usage:** Fits on 1× T4 or A10 GPU (16GB VRAM)
- **Ideal Environment:** Memory-constrained GPU servers, multi-instance cloud
- **Trade-offs:** Best perplexity and ROUGE, slower generation
- **Recommendation:** Best trade-off of performance and efficiency for cloud deployment (e.g., Hugging Face Spaces, TGI)

Compliance:

- No patient-identifiable information was used.
- Responses included disclaimers to meet HIPAA-like standards.

Conclusion & Future Work

We successfully developed a medically aware chatbot using Falcon-7B, trained with three PEFT methods. QLoRA emerged as the most effective trade-off between resource use and response quality.

Future Extensions:

- Train on 10k+ examples for generalizability and much better performance.
- Experiment with models with smaller size but with more epochs to capture the trade off.
- Try more PEFT techniques like prefix tuning and adapter training.
- Proper Hyperparameter tuning of the Fine tuning methods.
- Deploy these models using the strategy provided.

Hardware Used (KAGGLE)

- Kaggle T4 GPU, 15GB VRAM
- 29GB RAM
- 58GB DISK

Dataset Link :

https://www.kaggle.com/datasets/pythonafroz/medquad-medical-question-answer-for-ai-research

Model Links on Hugging Face Hub:

Qlora : https://huggingface.co/TestCase1/falcon-7b-qlora-chat-medical-bot
Lora : https://huggingface.co/TestCase1/falcon-7b-lora-chat-medical-bot
Prompt fp : https://huggingface.co/TestCase1/falcon-7b-prompt-fp-chat-medical-bot
Prompt quantised : https://huggingface.co/TestCase1/falcon-7b-prompt-chat-medical-bot