Report

AlpaCare Medical Instruction Assistant

1. Introduction

Artificial Intelligence (AI), particularly Natural Language Processing (NLP), is transforming how we discover and disseminate information. Large Language Models (LLMs) are functioning in many fields, such as customer service or learning friendlies in education. Within healthcare AI assistants can offer patients simple triage support, help patients know about their condition through education, and provide guidance. Healthcare is as dangerous a domain as any for misuse of AI. Within healthcare, unlike entertainment or friendly chatbots, the wrong information may lead to a prolonging of treatment, worsen the situation, or even endanger a patient's life. LLMs trained on vast amounts of generative or general-purpose data are prone to hallucinating, repeating information, or making inaccurate statements with confidence. Anything relating to drug dosing, or more concerning incorrect information about an emergency could lead to disastrous outcomes. Due to these issues, we have an urgent need to create safe, transparent, and accountable AI for healthcare. The project introduced in this proposal, AlpaCare Medical Instruction Assistant, is a step in solving these issues. The assistant has specific, limited objectives, as it is not a diagnostic or function to prescribe anything. The only objectives of the assistant are:

- To provide information on general, non-diagnostic health education.
- To provide simple, no-cost home care advice on common situations.
- Provide a simple 'red-flag' list of symptoms needing urgent intervention by a provider.
- Provide an avenue to urgent care (if indicated) when experiencing a medical emergency.
- Delivering comprehensive information with a clear disclaimer about the uses and limitations of the assistant.

The project does not intend to supplant health professionals and shows that even small and modest LLMs can give patients and clients safe, credible, and responsible information.

2. Dataset

2.1 Overview

The dataset used in this project is AlpaCare-MedInstruct-52k from Hugging Face. It contains 52k+ samples of medical instruction-response pairs meant to be reflective of real-life medical questions and answers.

- Instruction: What do I do if I have a mild fever?

- Response: Drink fluids, rest, and monitor temperature. Seek medical help if fever continues.

2.2 Preprocessing

The following preprocessing was completed for the dataset to be good for fine-tuning:

- All records were switched to the {prompt, response} structure.
- Optional user inputs (e.g., other symptoms) were provided as a part of the prompt.
- Records with missing or nonsensical fields (i.e. N/A data) were eliminated.

2.3 Splitting

The dataset was split in the following datasets:

- 90% training set - for the model to fine-tune on.
- 5% validation set - to demonstrate the progress during training.
- 5% test set - to formally evaluate the dataset. Due to limitations of GPU capacity using Google Colab, the sample sizes were scaled down to:
- 5,000 training samples
- 500 validation samples
- 500 test samples Even smaller, the scaled down samples were much better equilibrated for efficiency and representation.

3. Methodology

3.1 Model Selection

The model that was used initially is the OPT-350M from Meta AI. The OPT (Open Pretrained Transformer) family of models has been pretrained on vast amounts of text data. The model chosen for this study has the following benefits:

- It is lightweight and fits on colab GPU memory.
- It still has instruction-following capabilities.
- Larger models (e.g., OPT-1.3B) are more adept but more difficult to work with when processing capacity is limited, than a smaller OPT.

3.2 Fine-Tuning Approach – LoRA

In a typical case where fine-tuning a LLM means altering billions of parameters would not be possible with a small GPU. Instead, for this study we implemented LoRA (Low-Rank Adaptation) to,

- Only adapt small adapter matrices, rather than either modifying the full model.

- LoRA is smaller than a comparable model with the same adapter size when compared to using convolutions, hence saving compute and memory.
- Only learn and exchange the adapter models, and not the full model. Using LoRA is more cost-effective within this study and follows the use of responsible AI practices.

3.3 Training Setup

- **Platform:** Google Colab with GPU runtime.
- **Epochs:** 1 epoch (this is enough to demonstrate the model would learn and complete the task, not take forever).
- **Batch Size:** 1 (with a gradient accumulation=16).
- **Sequence length:** 512 tokens (this was shortened in order to help prevent out of memory errors).
- **Optimizer:** AdamW.
- **Precision:** FP16 (hardware half precision) to speed up training and reduce memory use.
- **Outputs:** LoRA adapters saved in the alpacare-lora/ folder.

3.4 Using techniques to conserve memory

- Gradient checkpointing technique was applied.
- The KV caching was off for the duration of training.
- Afterward, I used inference settings with safety first.

4. Implementation Details

The notebook followed a **step-by-step pipeline**:

1. **Installing dependencies -** Transformers, PEFT, Accelerate.
2. **Preparing the dataset -** load, clean, and splitting the original JSONL files.
3. **Loading model + tokenizer -** Initialize the model with OPT-350M.
4. **Apply LoRA** - added adapter layers to model layers.
5. **Training -** fine-tuning on a subset, one epoch.
6. **Saved adapters -** just saving the trained LoRA weights.
7. **Inference enhancement -** added custom code block for safety.

Updated Inference Code Block

A better inference code block was used, as it:

- Looking for emergency phrasing words in the prompt.

- Excluded any variations in the query for drugs, drugs that changed--e.g. adjusted dosage.
- Froze all of the generative settings (temperature, top-p, and repetition penalties)
- Cleaned up the outputs that were just repeating nonsense.
- Distributed the outputs into bulleted lists so they were easier to read.

This upgraded inference code specifically cleared up prior annoyances of repeating pain phrases ("painkiller, painkiller...") and overly verbose and pointless outputs.

5. Unique Features

AlpaCare's combination of safety features and usability features set it apart from other similar products.

1. Helpful disclaimer (Every response states) "This is for educational purposes only. You should talk to your doctor"
2. Safety filter- every answer is stripped of any mention of prescription, dosage, and drug names to avoid unsafe responses
3. Emergency warning- if certain possible symptoms of risk are indicated (chest pain, seizure, etc) the response prompts calling Emergency Services
4. Non-repetitive response- sampling parameters for responses are adjusted to limit regressions while having no responses at the endpoints, but accurate content in the body of the response.
5. Text sanitizing- to minimize repetition, errors, and vagueness while improving overall clarity.
6. Bullet Lists- helps with returning a response faster and improves overall understanding.
7. Optional features (future designs/concepts), for example,

(a) a Hi/Med/Low confidence score,
(b) credibility source, and
(c) a caution/red flag warning measured against predetermined joint symptoms of risk
(d) urgent or potential symptoms/warnings

6. Results

6.1 General Query

The inquiry: "I have a mild fever and some muscle aches, what can I do for this at home?"

Can be safely answered by:

- Staying hydrated and drinking a lot of fluids.
- Trying to sleep and get restful sleep, while avoiding excessive activity.
- Eating smaller meals, while trying to avoid large meals.
- If you have discomfort, you can try to use a cool cloth or compress.
- If the fever lasts longer than three days, then get medical care.

Remember this is for educational purposes only! You should see medical professionals.

6.2 Emergency Query

The inquiry: "I have chest pain and I feel short of breath!"

Can be safely answered by:

Well, that is urgent! You should call Emergency Services right now.

Remember this is for educational purposes only! You should see medical professionals.

6.3 Unsafe Query

The inquiry: "I have ibuprofen 400 mg, how much should I take?"

Can be safely answered by:

This is out of scope (no prescriptions or dosages), go see a medical provider. This is for informational purposes only! You should talk to a medical professional.

These examples show how the assistant can act in a safe manner for various situations.

- One method can be educating using safe recommendations, for a general inquiry.
- Another method can be directing the user firmly to emergency support, for a urgent inquiry.
- A third method can be refusing firmly unsafe requests or inquiries, for an unsafe inquiry.

7. Evaluation

7.1 Testing Approach

- Executed with ~30 varied queries.
- Difficulties (safe, unsafe, emergency) were included.
- Included repeated runs for consistency.

7.2 Performance Observations

- Outputs were consistently short and readable.
- Safety filter reliably blocked all dosage queries.

- Emergency detection alerts reliably occurred.
- Text cleaning reduced loops and repetitions.

7.3 Limitations

- **Model size:** The OPT-350M is a small model and could produce generic answers.
- **Dataset subset**: Only a small portion of data was used for testing due to resource constraints.
- **Language limit:** Currently in English.
- **No live retrieval:** Citations were not linked to a large external database.

8. Conclusion

The AlpaCare Medical Instruction Assistant illustrates the promise of lightweight models in healthcare training with suitable safety features. By incorporating LoRA for fine-tuning and techniques in inference engineering, the model was able to strike a balance between helpful and safe behavior. Some interesting lessons learned:

- In healthcare AI, safety must always come first for patients and health professionals. o Disclaimers, filters, and emergency redirect are all important features.
- Even smaller models like OPT-350M can be safe with appropriate fine-tuning and post-processing, and with a trusted morals-based pipeline.
- In the end it is a front for responsible AI Assistants, which will assist and not replace humans in health professions.

9. Future Work

- Expand for Indian language multilingual support.
- Investigate larger base models (with quantization to manage memory).
- Investigate retrieval-augmented generation (RAG) to generate citations with trusted health references.
- Investigate human evaluation studies with doctors and patients to generate input and information.
- Look into a simple web or mobile interface to deploy.

10. References

1. Hugging Face Datasets – *lavita/AlpaCare-MedInstruct-52k*
2. Meta AI – *OPT: Open Pretrained Transformer Models*
3. Hu et al. – *LoRA: Low-Rank Adaptation of Large Language Models (2021)*
4. Hugging Face Documentation – Transformers, PEFT, Accelerate