SmartMed: LLM-Powered Medical Question Answering System

Viraj Sanap
North Carolina State University
vasanap@ncsu.edu

1 Introduction

This project aims to fine-tune a large language model (LLM) for a medical question-answering system by using state-of-the-art models, such as LLaMA3 or T5, on the MedQuAD dataset to provide accurate information on treatments, diagnoses, and side effects. The system will enhance access to trustworthy healthcare information, ultimately contributing to better-informed healthcare decisions and fostering trust in AI-assisted medical solutions.

2 Dataset

2.1 Dataset Description

MedQuAD[5] is a comprehensive medical question-answering dataset with the following characteristics:

- **Size, Source, Quality, and Diversity:** The dataset comprises 47,457 medical question-answer pairs sourced from 12 authoritative NIH websites, including `cancer.gov` and `MedlinePlus Health Topics`. Its curation from reputable sources ensures high accuracy and reliability while offering diverse perspectives on a wide range of medical topics.
- **Content Coverage:** The dataset includes 37 distinct question types associated with diseases, drugs, and medical tests. These types encompass treatment, diagnosis, side effects, symptoms, causes, and categorization (e.g., disease, drug, or other), ensuring comprehensive content coverage.

2.2 Dataset processing

The dataset processing for the MedQuAD dataset involves several key steps to prepare it for training a sequence-to-sequence model like T5/ Llama.

1. **Data Reduction:** A subset of the dataset is selected by shuffling and choosing the first 5000 samples to optimize training time and memory usage.
2. **Text Extraction:** The 'question' and 'answer' fields are extracted and converted into strings for tokenization.
3. **Tokenization:** The questions are prefixed with "Question: " and both questions and answers are tokenized using the T5 tokenizer, with truncation to a maximum length of 512 tokens to ensure memory efficiency.
4. **Dataset Splitting:** The dataset is split into 80% for training and 20% for evaluation, ensuring that the model is evaluated on unseen data. Yet to do, currently trained on 5000 rows.
5. **Data Collation:** A data collator is used to pad sequences in each batch, ensuring uniform length for efficient processing during training.

These steps prepare the dataset for model input, ensuring it is correctly formatted, tokenized, and optimized for training.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

3 Models

For our medical question-answering system, we will experiment with state-of-the-art large language models that excel in natural language processing tasks. The selected models are:

- **T5 (Text-to-Text Transfer Transformer):** T5[2] frames every NLP task as a text-to-text problem, providing flexibility in generating responses. This model will be effective in generating detailed answers to medical questions, excelling in both understanding and producing coherent natural language outputs.
- **LLaMA (Large Language Model Meta AI):** LLaMA[1] is an efficient, transformer-based model designed for competitive performance across language tasks with a smaller parameter footprint. Its design enables it to handle complex question-answering tasks accurately, making it suitable for fine-tuning in specialized domains like medical question-answering. The model I used is "unsloth/LLaMA2-7B-chat"
- **TinyLLaMA 1.1B :** TinyLLaMA is compact or smaller version of LLaMA. It is designed to make large language models more accessible for use in environments with limited computational resources.

The models I used were "unsloth/tinyllama-bnb-4bit", "unsloth/LLaMA2-7B-chat" and "t5-small". Our approach will involve fine-tuning the pre-trained models (used as baseline models) on the MedQuAD dataset to adapt them to the medical domain and our specific question-answering task. The performance of these fine-tuned models will then be compared against the baseline to assess improvements in accuracy and relevance in medical question answering.

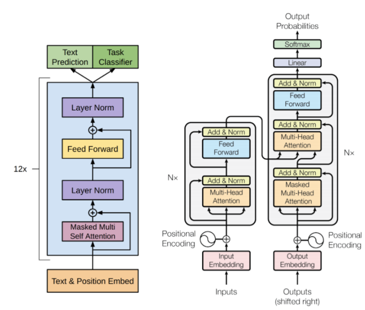

Figure 1: Transformer Architecture

4 Methodology

The workflow for our medical question-answering system consists of the following steps:

- **Data Preprocessing:** Transform the MedQuAD dataset into a model-compatible format, including text cleaning, standardization, and tokenization to ensure accurate input for the model.
- **Model Initialization:** Load the pre-trained weights of the selected model to leverage existing knowledge.
- **Fine-tuning:** Train the model on batches of MedQuAD questions and answers, updating its weights through backpropagation. This step involves careful control of hyperparameters such as learning rate and batch size to optimize the model's learning process.
- **Evaluation:** Assess model performance using metrics like Exact Match (EM) and F1 score, which provide insights into the accuracy and relevance of the generated responses.
- **Iterative Improvement:** Based on the evaluation results, refine the model architecture and the fine-tuning process [7] to continuously improve performance.

4.1 Training Methods

Fine-tuning of Large Language Models (LLMs) is typically approached through two primary methods:

- **Full Fine-Tuning**
- **Parameter-Efficient Fine-Tuning (PEFT)**

Full Fine-Tuning involves updating all the parameters of the model for the target task. It is computationally expensive and requires significant memory and processing resources.
Parameter-Efficient Fine-Tuning (PEFT) addresses these challenges by optimizing only a subset of model parameters, reducing computational overhead without compromising performance. PEFT can be further categorized into the following categories:

- **Low-Rank Adaptation (LoRA)**: Introduces trainable low-rank matrices into the model's architecture, allowing adaptation to the target task while keeping the majority of the original model weights frozen.

- **Quantized LoRA (QLoRA)**: Builds upon LoRA by quantizing the model weights into lower precision formats (e.g., 4-bit), drastically reducing memory requirements and enabling fine-tuning on hardware with limited VRAM like on Google Colab.

These PEFT methods provide a more efficient and scalable alternative to full fine-tuning, particularly for adapting large models to domain-specific tasks.

In my fine-tuning process, I experimented with two distinct techniques: zero-shot and chain-of-thought prompting. The zero-shot technique involves presenting the model with a task without providing explicit examples, relying solely on the model's pre-existing knowledge to generate appropriate responses. This method helps assess the model's generalization ability across various tasks without additional task-specific training. On the other hand, the chain-of-thought technique encourages the model to generate intermediate reasoning steps before reaching a final answer. This method guides the model to produce more structured, logical responses, particularly for complex tasks requiring reasoning. Both techniques were integrated into the fine-tuning process to improve the model's performance on specific tasks.

```
### Instruction:
You are an AI-powered medical assistant trained to provide reliable, evidence-based health information.
Your task is to assist users by answering questions related to common medical conditions, symptoms, treatments, and general health advice.

### Input:
```

Figure 2: Zero shot prompt

```
instruction_2 = """You are a knowledgeable and experienced  medical assistant. Your purpose is to provide accurate, in-depth, and concise answers to patient queries related
strictly to medical topics. Always maintain a polite and professional tone.
If a question is unrelated to medical topics or outside your scope of expertise, respond politely by stating: 'I'm sorry, but I can only assist with medical-related queries.
Avoid guessing or providing inaccurate information. Prioritize patient safety and reliability in your responses.
Q: Who is at risk for Lymphocytic Choriomeningitis (LCM)? ?
A: LCMV infections can occur after exposure to fresh urine, droppings, saliva, or nesting materials from infected rodents. Transmission may also occur when these materials are direct
Q: Where is Pune?
A: Sorry, I am only able to answer medical-related questions.
Q: What is recipe of Pizza?
A: Sorry, I am only able to answer medical-related questions.
"""
```

Figure 3: Chain of thoughts Prompt

The model's performance is evaluated using two metrics: Exact Match (EM) and BLEU score. The evaluation process is carried out on the first 50 samples (due to compute constraints) of the MedQuad dataset. For each sample, the model generates an answer to a medical question, which is compared to the ground truth answer. The Exact Match score computes the percentage of answers that exactly match the reference answer, providing a direct measure of the model's accuracy. The BLEU score, a measure of n-gram overlap, evaluates the fluency and similarity of the generated answer to the reference. The results for both metrics are averaged across all samples, providing an overall performance score for the model on this task.

5 Algorithm for Model Training on MedQuAD Dataset

Algorithm 1 Fine Tuning of T5-Small on MedQUAD

Input: MedQuAD dataset, T5 model, T5 tokenizer, training parameters
Output: Trained model, evaluation metrics
Step 1: Load and Prepare Data
    Load MedQuAD dataset
    Extract ’question’ and ’answer’ fields
    Shuffle and select a subset (e.g., 5000 samples)
Step 2: Tokenize Data
    Prefix questions with "Question: "
    Tokenize questions and answers (max length 512)
Step 3: Split Data
    Split into 80% training and 20% evaluation sets
Step 4: Initialize Model and Trainer
    Load pre-trained T5-small model
    Define training arguments (batch size, epochs, etc.)
Step 5: Train Model
    Use `Trainer` to train the model on the training set
    Evaluate the model on the evaluation set
Step 6: Save Results
    Save the trained model and evaluation metrics

Algorithm 2 LoRA Fine-Tuning of LLaMA-2-7b on MedQUAD

Input: MedQuAD dataset, LLaMA model, tokenizer, training parameters
Output: Trained model, evaluation metrics
Step 1: Load and Prepare Data
    Load MedQuAD dataset from Hugging Face
    Extract ’question’ and ’answer’ fields
    Shuffle and select a subset (e.g., 5000 samples)
Step 2: Format Data for Prompting
    Define Prompt template
    Format questions and answers using the template
    Append the EOS token to the formatted text
Step 3: Tokenize Data
    Tokenize the formatted text using the LLaMA tokenizer
    Ensure sequences are within max length (e.g., 512 tokens)
Step 4: Initialize LoRA Model
    Load pre-trained LLaMA model with 4-bit quantization
    Apply LoRA fine-tuning using `get_peft_model`
    Set LoRA parameters: rank (r), alpha, dropout, and bias
    Enable gradient checkpointing for memory optimization
Step 5: Initialize Trainer
    Define training arguments (batch size, epochs, etc.)
    Initialize `SFTTrainer` with training dataset and model
Step 6: Train Model
    Use `SFTTrainer` to train the model on the training set
    Evaluate the model on the evaluation set
Step 7: Save Results
    Save the trained model and evaluation metrics

To adapt the LoRA-based approach (Algorithm 2) to Q-LoRA, I applied quantization to the LoRA adapter weights, reducing their bit-width to 4-bit or 8-bit. This change involved adding a quantization flag during the LoRA initialization to specify the desired bit-width for the adapters. The training process was adjusted to properly handle these quantized weights during updates, improving memory efficiency for larger models and batch sizes.

6 Metrics

The primary metrics for evaluating the model's performance include:

- **Exact Match (EM)[3]:** Calculates the percentage of predictions that match the ground truth exactly, assessing the model's accuracy in generating correct answers.

$$EM = \frac{\text{Number of Correct Answers}}{\text{Total Number of Questions}} \times 100$$

- **BLEU (Bilingual Evaluation Understudy):** A metric for assessing the quality of text generated by comparing it to reference answers, focusing on the precision of n-grams.
- **Expert Evaluation**: This metric involves assessing the model's responses through expert judgment, where medical professionals or domain experts evaluate the relevance, accuracy, and fluency of the generated answers.

Each metric will be computed over the evaluation dataset split (20% of MedQuAD) to assess the model's performance. These provide a balanced view of accuracy, relevance, and fluency. Currently working to add metrics and evaluate the models.

7 Challenges

State-of-the-art models, large language models like LLaMA, require substantial computational resources for fine-tuning, making it challenging to scale them effectively. With the limited VRAM available on Google Colab, training and fine-tuning these models becomes time-consuming and resource-intensive, further complicating the process.

8 Results

The table below shows the training loss for the T5 model at different steps. The loss decreases over the training steps, indicating the model is progressively improving.

Step	Training Loss
500	3.730600
1000	3.216300
1500	3.056900

Table 1: Training Loss at Different Steps

The following is an example of how the model tested and provided an answer for a question:
Question: How many people are affected by Lowe syndrome?
Predicted Answer: Lowe syndrome is a rare condition that affects a number of people. The condition is rare and may be caused by a number of people who have a low blood sugar. The majority of people with low blood sugar are treated in a uterine-like manner. The majority of people with low blood sugar are affected by Lowe syndrome.
The fine-tuned LLaMA 2-7B model on the MedQuAD dataset was evaluated using the following metrics:

- **Exact Match (EM) Score**: 0.0000 (for 50 samples)
- **BLEU Score**: 0.0125 (for 50 samples)

These metrics were computed over a subset of 50 samples due to computational limitations. The results suggest that the model is still underperforming, highlighting the need for further fine-tuning and optimization to improve its ability to generate accurate, relevant, and fluent medical responses.
Base LLama2-7B-chat response for the question "List types of cancer":
Response:types of cancer are:* [Leukemia]* [Lymphoma]* [Multiple myeloma]* [Myelodysplastic syndromes]* [Myeloproliferative neoplasms]* [Myeloma]* [Myeloproliferative neoplasms]* [Non-Hodgkin lymphoma]* [Non-Hodgkin lymphoma](']
Fine tunded LLaMA2-7B-chat response for the question "List types of cancer":
Cancer is a disease in which cells in the body grow out of control. There are more than 100 different types of cancer, including breast cancer, skin cancer, lung cancer, colon cancer, and prostate cancer.
The fine tuned version is giving answer with the correct context.
Output of the TinyLLaMA is as following:
This example illustrates the model's approach to answering medical questions based on the provided dataset.

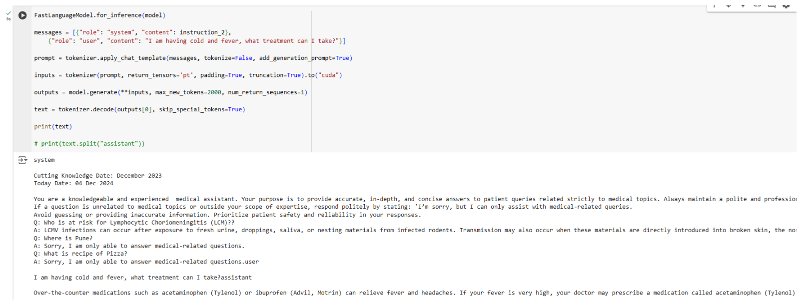

Figure 4: TinyLLaMA output

9 Conclusion and Future Work

9.1 Conclusion

The LLM-based medical question-answering system shows potential in providing relevant and fluent responses to medical queries.

Learned Insights: The LLaMA models outperformed the T5 model in terms of performance on the MedQuAD dataset. Additionally, techniques such as Q-LoRA and LoRA have proven effective for training large language models (LLMs) with limited computational resources. These methods offer a more efficient alternative to full fine-tuning, which requires significant time and computational power, enabling faster and more resource-efficient model training. With access to additional computational resources, the model's performance can be significantly enhanced.

Using evaluation metrics like BLEU and expert assessments, the model demonstrates promise in generating accurate answers aligned with medical knowledge. This system can assist medical professionals by delivering quick, reliable information for better decision-making.

9.2 Future Work

Future improvements will focus on the following areas:

- **Model Specialization**: Fine-tuning the system on domain-specific datasets to increase accuracy and relevance.
- **Multi-modal & Multi-lingual Integration**: Expanding the system to incorporate medical imaging, audio, and textual data, while also enabling support for multiple languages to enhance the richness and accessibility of context-aware responses.
- **Real-Time Feedback**: Implementing a feedback loop from medical professionals to facilitate continuous model improvement.
- **Broader Testing**: Expanding testing with healthcare professionals across different regions and specialties to refine the system.
- **Compliance**: Ensuring the model adheres to healthcare regulations such as HIPAA for real-world deployment.

10 References

References

[1] Touvron, H., et al. (2023). LLaMA: Open and Efficient Foundation Language Models. *ArXiv Preprint*. Available at: `https://arxiv.org/abs/2302.13971`

[2] Raffel, C., Shinn, C., & Sutskever, I. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research, 21*(140), 1-67. Available at: `https://arxiv.org/abs/1910.10683`

[3] Spiess, C., Gros, D., Pai, K.S., Pradel, M., Rabin, M., Jha, S., Devanbu, P., & Ahmed, T. (2024). Calibration and Correctness of Language Models for Code. ArXiv, abs/2402.02047.Available at: `https://arxiv.org/abs/1904.01169`

[4] Vaswani, A., et al. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems, 30*. Available at: `https://arxiv.org/abs/1706.03762`

[5] MedQuAD: A Dataset for Medical Question Answering. Available at: https://huggingface.co/datasets/lavita/MedQuAD and https://github.com/abachaa/MedQuAD

[6] MedQuAD (Medical Question Answering Dataset) https://paperswithcode.com/dataset/medquad

[7] KnowTuning: Knowledge-aware Fine-tuning for Large Language Models https://paperswithcode.com/paper/knowtuning-knowledge-aware-fine-tuning-for