Project Report: Training Mistral 7B for Research Problem Approach Generation

Course: CMPSC 497
Submission Date: April 25, 2025
Group Members: Yuv Boghani and Shreya Buddharaju

Introduction

Our project aimed to train an open-source large language model (LLM) to generate detailed approaches for given research problems. This process allowed us to gain hands-on experience in both dataset creation and model fine-tuning, while also critically evaluating the steps and challenges involved.

Dataset Construction

To build a dataset tailored to our task, we first used ChatGPT to generate pairs of research problems and their corresponding detailed approaches by asking it to read through real life research papers and some potential research problems of interest and then summarize them to their respective CSV entries. This enabled us to quickly gather a varied and relevant collection of data and examples. After generation, we exported the data to a CSV file, then extracted and cleaned the relevant columns to ensure consistency and remove duplicates. The final dataset consisted of entries where each "problem" was a short research description (1 line approx.) and each "approach" provided a longer, detailed solution strategy (3 lines approx.).

Data Preprocessing and Model Preparation

For preprocessing, we focused on standardizing the input to the model. We applied a tokenization process that converted each "problem" text into a fixed-length sequence, ensuring all inputs were either padded or truncated to the same size. This step was essential for efficient batch processing and compatibility with our model architecture.

To optimize for limited computational resources, we prepared the Mistral 7B model for parameter-efficient fine-tuning using LoRA (Low-Rank Adaptation). This involved configuring the model to update only a subset of its parameters, specifically targeting modules most relevant for adaptation. We also enabled k-bit training to further reduce memory usage, making it feasible to train a large model within our resource constraints.

Model Selection and Training Details

Model Selection and Training Details

We selected Mistral 7B as our base model due to its strong performance and open-source availability. To make fine-tuning feasible with limited resources, we used LoRA (Low-Rank Adaptation), a parameter-efficient technique that drastically reduces the number of trainable parameters by introducing small trainable matrices into the model's architecture while keeping the original weights frozen. This allows the model to adapt to new data with far less memory and compute, and the performance is often comparable to full fine-tuning.

During training, we configured LoRA to target the attention modules specific to Mistral 7B and used quantization to further reduce memory usage. We trained the model for three epochs, optimizing the causal language modeling loss. Only the LoRA adapter parameters were updated during this process, which made the training process much more efficient and practical given our hardware constraints.

We had multiple challenges when testing out various LLM models. Due to the constraints of running the models on Google Colab and its T4 GPU, we were unable to run Llama 2B, tinyllama, and even Mistral for a long period of time (both in the base model and with quantization techniques). Each of these models faced limitations with respect to the RAM available with the GPU. We tried a lightweight model with ‘BART’ but it struggled with the dataset provided and was unable to give a usable result, either repeating the question itself or repeating overused words. We assume it was unable to comprehend the question prompts as it was a far more lightweight model.

Evaluation Metrics and Experimental Results

To assess our model's performance, we used two primary metrics:

- **ROUGE Score:** Our final model achieved a ROUGE score of 0.40, indicating a moderate level of overlap between the generated approaches and the reference solutions.

- **BERTScore:** We also measured semantic similarity using BERTScore, with our model achieving a score of 0.72. This suggests reasonable alignment in meaning between the model outputs and the ground truth.

Analysis and Reflections

Through this project, we learned that constructing a high-quality, task-specific dataset is both challenging and critical for successful LLM fine-tuning. Parameter-efficient methods like LoRA and k-bit training proved essential for adapting large models with limited resources. However, our reliance on ChatGPT summarized data may have introduced biases, and training for only

three epochs most definitely limited the model's full potential and evaluation scores. Additionally, while metrics like ROUGE and BERTScore provide useful signals, they may not fully capture the quality or creativity of generated approaches for open-ended tasks.

Also having been through the generated results and the actual true labels for the approaches we feel that the main difference is the formatting and not the actual content and its quality, these differences in formatting may also have attributes to lower cores in our evaluation metrics. However, with more training time, we are confident the model would perform significantly better.

Conclusion

This project gave us practical experience with every step of fine-tuning a large language model, from building our own dataset to evaluating results. Even though we had limited computing resources, we were able to successfully adapt Mistral 7B for our specific academic task using efficient training methods. We learned what works well and what can be challenging when training large models with custom data and limited hardware.