Instruction-Following Fine-Tuning of Large Language Models for Enhanced Daily Life Tasks and Mathematical Reasoning

Raiyan Ahmed RAIYAN.AHMED05@NORTHSOUTH.EDU
ID: 2013130042
Department of ECE
North South University

Mehedi Hasan MEHEDI.HASAN41@NORTHSOUTH.EDU
ID: 2011425042
Department of ECE
North South University

Mahir Shahriar Abir MAHIR.ABIR@NORTHSOUTH.EDU
ID: 2013640042
Department of ECE
North South University

Editor: Raiyan Ahmed, Mehedi Hasan ,Mahir Shahriar Abir

Abstract

Instruction Fine-tuning improves pretrained language models from basic next-word prediction to complex instruction following. In this work, we focus on developing an instruction-based system that enhances daily life through prompt engineering. This system is created to handle three main tasks, which can work with general instructions, simulating roleplay scenarios, and solving simple math problems. That is why we used datasets like Alpaca, which contains various instructions, and the Camel-AI Math Dataset for simpler mathematical tasks. At first, we fine-tune Llama-3.2-1B-Instruct-bnb-4bit and Qwen 2.5 1.5B bnb 4bit. We used a custom prompt format to teach them how to follow instructions better. After training, the models were evaluated using metrics like perplexity, which measures how well the model predicted the next token. The results showed excellent performance with low perplexity scores, specifically in math tasks. Llama-3.2 achieved a perplexity of 1.68 on the Math dataset, 2.81 on the Alpaca dataset, and 2.68 on the combined Math + Alpaca dataset. On the other hand, Qwen 2.5 achieved a perplexity of 1.36 on the Math dataset, 1.61 on the Alpaca+math dataset, and 2.67 on the combined Math + Alpaca dataset.

Keywords: prompt engineering, Qwen, Llama, and perplexity scores

1 Introduction

In recent years, prompt and instruction-based tuning (1) have appeared as popular techniques for natural language processing. It helps us with our various tasks, from answering questions. AI interacts with users through instructions and prompts. The main idea is to use prompt engineering, which means teaching the model to understand instructions and react accurately based on the context. Here, LLM-based models play a vital role. We can use those LLM-based models and fine-tune them through prompts for our specific task. This project focuses on developing an instruction-based system that uses LLM-based models to enhance daily life tasks. LLM is a large language model that is used to predict the next word (2) and generate human-like text. These models are trained on vast amounts of text data. We used six models for fine-tuning including Qwen and LlaMa. LLaMA is a type of model created by Meta (3). LLaMA makes it a great choice for projects where accuracy and efficiency are important. It can understand and generate text with less training compared to other models of similar size. At the same time, Qwen works well with multiple languages and tasks (4). Both are optimized for fine-tuning for specific tasks. We used popular datasets like the Alpaca dataset, which provides instructions and the Camel-AI Math Dataset for math-related problems. We used perplexity on these six models, which showed better results on mathematical tasks. This project shows how instruction-based fine-tuning can create AI systems that are easy to use and benefit people in their daily activities.

2 Methodology

2.1 Dataset

2.1.1 Alpaca Dataset

The Alpaca Dataset includes a diverse set of instruction-response pairs intended for fine-tuning pre-trained large language models on general instruction-following tasks. Alpaca was developed as an open-source alternative to proprietary instruction datasets and thus provides a wide array of both real-world and synthetic scenarios to enhance the comprehension and adaptability of large language models (LLMs).

Key Features:

- **Directions:** The directions include a wide variety of tasks ranging from answering factual questions, summarization, reasoning, and generating explanations to creative tasks.
- **Domains:** Broad domains of sciences, technologies, humanities, business, arts, and even day-to-day scenarios.
- **Prompt Structure:** Designed to simulate natural human-like interactions, often containing substantial context and tailored specific requirements.
- **Training on Utility:** This training engages the model in handling day-to-day instructions, creating meaningful completions, or producing diverse and relevant outputs.

Example Entries:

- Instruction: Name a reason why someone might switch to a paperless office environment.
  Response: To reduce waste and promote environmental sustainability.
- Instruction: Explain the principle of a light bulb producing light.
  Response: When electrical current flows through the filament, it heats up and emits light due to incandescence.

Purpose: The dataset is particularly useful for teaching models to follow structured instructions, adapt to different contexts, and improve general usability.

2.1.2 Camel AI Dataset

The Camel-AI Math Dataset is focused on domain-specific tasks in mathematics. It aims to develop models that excel in logical reasoning, problem-solving, and structured mathematical thinking. This dataset incorporates various kinds of mathematical challenges and is structured to train and benchmark models for precision and coherence in mathematical reasoning.

Key Features:

- **Topic Coverage:** Coverage of various mathematics disciplines, including:
  - **Algebra:** Solve inequalities, graph equations.
  - **Geometry:** Solve for area and properties of shapes.
  - **Calculus:** Problems on integrals, derivatives, work-energy.
  - **Statistics:** Survival analysis, probability.
  - **Combinatorics:** Pascal's triangle, Polya's enumeration theorem.
  - **Linear Algebra:** Solution of systems of equations.
  - **Graph Theory:** Chromatic numbers, properties of graphs.
- **Problem Types:** Word problems, proof of theorems, and computations using formulas.
- **Style:** Includes problem statements, contextual hints, and interactive challenges.

Example Entries:

- Instruction: Solve the equation: $3x + 5 = 20$.
  Response: Subtract 5 from both sides: $3x = 15$. Then divide by 3: $x = 5$.
- Instruction: Find the area of a rectangle with length 10 units and width 5 units.
  Response: The area is length × width $= 10 \times 5 = 50$ square units.

Purpose: This dataset is designed to enhance the ability of models to solve mathematical exercises with improved accuracy and fluency, providing well-justified step-by-step responses.

2.2 Data Pre-processing

We didn't need to clean the alpaca dataset, which takes instruction, as it was already cleaned from the beginning. Then, We modified the topic column of the Camel-AI math dataset and attached it with the sub-topic, making a new column named instruction and the input and output columns. To sum up, we cleaned the Camel-AI dataset by using prompting. We used unsloth, which is a framework used to clean, format, and optimize datasets for training (5). We also applied 4-bit Quantization, A technique that compresses the model's parameters into 4-bit representations (6). It is a very useful method for fine-tuning large language models, though it reduces smaller accuracy. Lastly, we added Parameter-Efficient Fine-Tuning. It's a method for fine-tuning only selected modules of a model instead of the entire model, which can save training time. Here, QLoRA fine-tunes only targeted modules of the model while keeping the rest untouched (7).

3 Result

Tasks	**Version**	**Filter**	**n-shot**	**Metric**	**Value ± StdErr**
hellaswag (Instruction)	1	none	0	acc	0.5022 ± 0.0050
mathqa_gen	3	none	0	acc_norm	0.6679 ± 0.0047
				bleu_acc	0.3684 ± 0.0169
				bleu_diff	-2.4674 ± 0.6554
				bleu_max	21.8842 ± 0.7716
				rouge1_acc	0.4027 ± 0.0172
				rouge1_diff	-3.1973 ± 0.7711
				rouge1_max	45.1347 ± 10.8531
				rouge2_acc	0.3048 ± 0.0161
				rouge2_diff	-4.6801 ± 0.9059
				rouge2_max	30.9575 ± 0.9659
				rougeL_acc	0.4100 ± 0.0172
				rougeL_diff	-3.3127 ± 0.7746
				rougeL_max	42.3403 ± 0.8627
mathqa_mc1	2	none	0	acc	0.3023 ± 0.0161
mathqa_mc2	2	none	0	acc	0.4780 ± 0.0148

Table 1: Evaluation Metrics for Different Tasks and Versions

Model	**Dataset**	**Perplexity**
Llama-3.2-1B-Instruct(Math)	Math	1.682
Llama-3.2-1B-Instruct(Alpaca)	Alpaca	2.813
Llama-3.2-1B-Instruct(Math+Alpaca)	Math + Alpaca	2.685
Qwen 2.5 1.5B(Math)	Math	1.365
Qwen 2.5 1.5B(Alpaca+math)	Math	1.609
Qwen 2.5 1.5B(Alpaca+math)	Alpaca	2.660

Table 2: Perplexity Scores of Different Models on Various Datasets

Here, the table 1 shows the result of the Qwen 2.5 1.5B model on the alpaca plus math dataset. We chose this model because it was better than the rest of the models and could perform Mathematical tasks perfectly.

Again, the table 2 shows the results with low perplexity scores, specifically in math tasks. Llama-3.2 achieved a perplexity of 1.68 on the Math dataset, 2.81 on the Alpaca dataset, and 2.68 on the combined Math + Alpaca dataset. On the other hand, Qwen 2.5

achieved a perplexity of 1.36 on the Math dataset, 1.61 on the Alpaca+math model with math data, and 2.67 on the combined Math + Alpaca model with alpaca data.

4 Conclusion

Fine-tuning of large language models showed significant improvements for many kinds of instruction-execution tasks, roleplay scenarios, and even mathematical problem-solving. Fine-tuning models Llama-3.2-1B-Instruct-bnb-4bit and Qwen 2.5-1.5B-bnb-4bit was carefully implemented upon selected datasets: Alpaca for general instruction tasks, Camel-AI Math for mathematical reasoning. Both models did very well for all these tested datasets, as reflected by the evaluation upon perplexity-a generally adopted metric of assessing the accuracy of prediction. Llama-3.2 scored 1.68 on the Math dataset, 2.81 on the Alpaca dataset, and finally 2.68 on the combined Math + Alpaca dataset. These are serious improvements in the case of math reasoning and instruction-following tasks. For one, Qwen 2.5 outperformed Llama-3.2, scoring 1.36 on the Math dataset, 1.61 on the Alpaca + Math dataset, and 2.67 on the combined dataset. This is indicative of Qwen 2.5's better understanding of integrated and mathematical tasks. The overall performance was quite decent to produce results that were accurate, contextually relevant, and logically coherent, especially when working with more complicated combinatorial, graph-theoretic, and calculus-heavy domains aside from general knowledge and algorithmic problems. The above benchmarks not only report the reduction in perplexity but also how well the different models respond to varied instructions, with Qwen 2.5 marginally edging out in mathematical reasoning. These results really pinpoint the potentials of prompt engineering and dataset curation as a catalyst in improving instruction-following LLMs. Further work is encouraged to be done by incorporating more domain-specific datasets, trying other parameter-efficient tuning methods such as LoRA, and adding human-in-the-loop evaluations for a more fine-tuned quality and application in reality.

References

[1] Ouyang et al., "Training language models to follow instructions with human feedback," arXiv (Cornell University), Jan. 2022, In NeurIPS.

[2] S. Longpre et al., "The Flan Collection: Designing Data and Methods for Effective instruction tuning," arXiv (Cornell University), Jan. 2023.

[3] H. Hu, P. Chen, and E. M. Ponti, "Fine-tuning Large Language Models with Sequential Instructions," arXiv (Cornell University), Mar. 2024.

[4] I. Jindal, C. Badrinath, P. Bharti, L. Vinay, and S. D. Sharma, "Balancing Continuous Pre-Training and instruction Fine-Tuning: Optimizing Instruction-Following in LLMs," arXiv (Cornell University), Oct. 2024.

[5] S. Kumar, "Overriding Safety protections of Open-source Models," arXiv (Cornell University), Sep. 2024.

[6] S. Ashkboos et al., "QUIK: Towards End-to-End 4-BiT Inference on Generative Large Language models," arXiv (Cornell University), Jan. 2023.

[7] L. Wei et al., “An adapted large language model facilitates multiple medical tasks indiabetes care,” arXiv (Cornell University), Sep. 2024.