SWER – Paper Progress Report

Bowen Yang UNI: BY2365

February 2024

Introduction

Building on the concept I previously outlined, my aim is to develop a mathematics tutoring system that is both **swift** and **accurate**. In this report, I plan to introduce the research papers that feel inspirational.

Venn Diagram

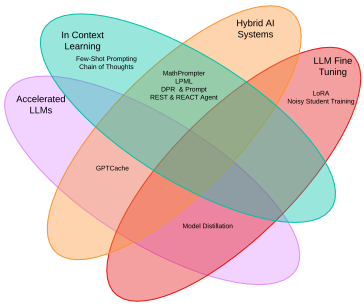

Figure 1: Current Progress Venn Diagram

Control LLMs' Behavior

The core of the product is a the concreteness of large language model's reasoning skill. We don't want large language model to make mistakes on questions before it tries to guide the student. To control the behavior of LLMs, three main strategies are identified:

1. Leveraging in-context learning

2. Fine-tuning LLMs

3. Creating hybrid AI systems.

In-context Learning

In-context learning refers to the ability of a LLMs to adapt to new tasks or understandings based on the context provided in the input, without the need for explicit retraining or fine-tuning. Within in-context learning, few-shot prompting[1] and chain-of-thought (COT)[2] prompting stand out as foundational techniques.

Fine Tuning LLMs

In the realm of refining Large Language Models (LLMs), the LoRA algorithm [5] stands out as a pivotal technique. Given the prohibitive costs associated with comprehensive fine-tuning of these expansive models, LoRA presents a viable alternative that economizes on resources. It achieves this by maintaining the core parameters of the model in a static state while focusing the training process exclusively on a minimal set of additions to each neuron. This targeted approach allows for the optimization of LLMs without the necessity of overhauling the entire model, offering a cost-effective solution for enhancing their performance.

In addition to financial considerations, the success of model training hinges on the availability of adequate data. Overcoming the hurdle of limited data can be achieved through the Self-training with Noisy Label technique[10], which enables the model to further enhance its learning by utilizing synthetically generated data.

Hybrid AI Systems

Hybrid AI systems refers to providing AI ability to interact with other tools such as python compiler. Usually, hybrid system are used with in-context learning because LLMs possess the capability to understand those artifacts generated by other tools in prompt. Yamauchi et al. [3] proposed an approach that integrates the Chain-of-Thought (CoT) methodology with a Python Read-Eval-Print Loop (REPL) to bolster the mathematical problem-solving capabilities of LLMs. A key insight from their work is the significant performance boost obtained when LLMs are tasked with generating Python scripts as an additional step to verify the outputs produced through CoT reasoning. Building on this concept, researchers at Microsoft have devised the MathPrompter framework [4]. This framework begins by converting a given question into an algebraic template, followed by the generation of Python code. It then employs a comprehensive multi-step verification and cross-checking process on the generated solutions to ensure higher accuracy and reliability in solving mathematical problems. Renat Aksitov et al.[11] proposed an another system combines the ReAct method, which integrates external knowledge, with the ReST self-training algorithm for continuous improvement through AI feedback and synthetic data. This method

enables the LLM to refine its reasoning and perform better in complex question-answering tasks, demonstrating the potential for LLMs to self-improve without direct human-labeled data, particularly in mathematical problem-solving contexts.

While these systems enhance LLMs' math reasoning skills, they share a common drawback: speed, due to the need for multiple client-LLM interactions. Inspired by Sewon Min's lecture on dense passage retrieval[7] and the uniform nature of primary school math problems, I see the value in using related problem contexts for improvement. My goal is to create a system that identifies and uses similar problems based on user queries, improving model outputs with contextual prompts.

Accelerated LLMs

My primary goal is to enhance the speed of real-time predictions, emphasizing the importance of promptly delivering feedback to users. This is crucial because interactions with LLMs via APIs often face delays due to network latency and the inherent processing times of LLMs. To address these challenges without the ability to directly accelerate LLM or network speeds, two innovative client-side solutions have been identified.

First, GPTCache, as introduced by Bang Fu and DiFeng, is a semantic caching mechanism for LLMs. It stores queries and their responses based on the semantic content, significantly reducing the need to send repetitive queries to LLM servers, especially when users pose similar questions.

Second, drawing inspiration from a guest lecture at Columbia University, model distillation presents another promising approach. This technique involves training a smaller, more specialized model under the guidance of a larger "teacher" model. Specifically, by leveraging a LLM like GPT as the teacher, it's feasible to develop a smaller model focused exclusively on replicating GPT's mathematical capabilities, thereby achieving more efficient operational performance.

Experiment

For highly specialized domains like primary school math, which often have repetitive structures, I propose that tailored prompts providing LLMs with analogous questions will outperform broader systems. My planned experiment will assess both a generalized approach and this specialized prompting strategy by testing them against a set of primary school math problems. Depending on difficulty, I might adjust the problem set to ensure a rigorous comparison, aiming to directly measure and compare their accuracy.

Reference

1. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language Models are Few-Shot Learners. Retrieved from https://ar5iv.org/abs/2005.14165

2. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Retrieved from https://ar5iv.org/abs/2201.11903

3. Yamauchi, R., Sonoda, S., Sannai, A., & Kumagai, W. (2023). LPML: LLM-Prompting Markup Language for Mathematical Reasoning. Retrieved from https://ar5iv.org/abs/2309.13078

4. Imani, S., Du, L., & Shrivastava, H. (2023). MathPrompter: Mathematical Reasoning using Large Language Models. Retrieved from https://ar5iv.org/abs/2303.05398

5. Hu, H., Peng, J., Wang, R., & Liang, P. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv. https://arxiv.org/abs/2106.09685

6. Zhuang, H., Qin, Z., Hui, K., Wu, J., Yan, L., Wang, X., & Berdersky, M. (2023). Beyond Yes and No: Improving Zero-Shot LLM Rankers via Scoring Fine-Grained Relevance Labels. Retrieved from https://ar5iv.org/abs/2310.14122

7. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-t. (2020). Dense Passage Retrieval for Open-Domain Question Answering. Retrieved from https://ar5iv.org/pdf/2004.04906v1

8. Bang, F., Tan, L., Milajevs, D., Chauhan, G., Gwinnup, J., & Rippeth, E. (2023). GPTCache: An Open-Source Semantic Cache for LLM Applications Enabling Faster Answers and Cost Savings. In Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023), December 2023. Association for Computational Linguistics.

9. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

10. Xie, Q., Luong, M.-T., Hovy, E., & Le, Q. V. (2020). Self-Training With Noisy Student Improves ImageNet Classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 10687-10698)

11. Aksitov, R., Miryoosefi, S., Li, Z., Li, D., Babayan, S., Kopparapu, K., Fisher, Z., Guo, R., Prakash, S., Srinivasan, P., Zaheer, M., Yu, F., & Kumar, S. (2023). ReST meets ReAct: Self-Improvement for Multi-Step Reasoning LLM Agent. arXiv