Chain-of-thought for complex reasoning using language models

Chandana Pamidi
cpamidi@umass.edu

Sreelekha Yaga
syaga@umass.edu

Sushrita Yerra
sushritayerr@umass.edu

Swetha Eppalapally
seppalapally@umass.edu

Yogeshwar Pullagurla
ypullagurla@umass.edu

1 Introduction

Recent advancements in large language models (LLMs), such as GPT-3, have significantly enhanced their reasoning abilities, making them invaluable for a variety of applications. The interest in improving these models by increasing their size and augmenting their reasoning capabilities with external tools is well-founded. However, LLMs often struggle with tasks requiring advanced, complex reasoning. Studies, such as (Wei et al., 2023) (Lightman et al., 2023), have documented these challenges, motivating further research into their limitations and potential enhancements.

This project is inspired by the PRM800k(Lightman et al., 2023) and MediMCQA(Pal et al., 2022) datasets, which consist of math and medical questions respectively, demanding intricate multi-step reasoning. However, due to our constraints, we have opted to focus on the Aqua(Ling et al., 2017) dataset, a collection of math reasoning problems. Our hypothesis posits that fine-tuning LLMs for both reasoning and answering will enhance their performance beyond what is achievable through standard or chain-of-thought prompting techniques. We aim to explore how such fine-tuning affects the model's reasoning abilities and its tendency towards generating incorrect information (hallucinations). This research will not only provide insights into fine-tuning's impact on LLM reasoning but can also serve as a foundation for future studies on more complex reasoning tasks.

By investigating these areas, our project seeks to contribute to the ongoing dialogue on enhancing the cognitive capabilities of LLMs, offering both theoretical and practical implications for their development.

2 Related work

Chain-Of-Thought(COT) While large language models (LLMs) have shown significant advancements in certain domains, there is still a need for improvement in tasks such as Complex Reasoning, Relation Extraction, and Data Summarization. The introduction of chain of thought prompting, as demonstrated in(Wei et al., 2023), allows reasoning abilities to naturally emerge in LLMs. Pipeline CoT, has been successfully applied to Relation Extraction (RE) without requiring explicit entity information in the prompt (Zhao et al., 2023), where RE involves automatically identifying semantic relationships between entities in unstructured or semi-structured natural language text. Despite their capabilities, LLMs currently lack formal reasoning in their processes. Even when results appear to have underlying reasons, LLMs primarily map input text or instructions (prompts) to probable continuations or responses without explicit reasoning. This limitation poses challenges for higher-level summarization. To address this, a more advanced form of data summarization, referred to as Hierarchical CoT, has been proposed to achieve higher-level summarization(Rukmono et al., 2023). Although the above hybrid models solve wide range of tasks, they still need to improve strategic decision-making by considering various reasoning paths. This can be achieved by using Tree-of-Thought(ToT) which is proposed by (Yao et al., 2023).

AQUA-RAT (Ling et al., 2017), short for Algebra Question Answering with Rationales, a dataset comprising approximately 100,000 algebraic word problems, each accompanied by detailed rationales. The dataset systematically explains the solution process for each problem in natural language, offering a valuable resource for training models in Chain of Thought (CoT) capabilities specifically tailored for mathematical contexts. (Zhang et al., 2022) conducted experiments by applying CoT to the AQuA(Ling et al., 2017) dataset, demonstrating the potential to streamline manual efforts by utilizing Large Language Models (LLMs). By employing the prompt ”Let's think step by step,” (Zhang et al., 2022) showed that LLMs can generate reasoning chains systematically, not only step by step but also one by one.

Large Language Model Meta AI (LLaMA) There are multiple variations of LLaMA. Variants of LLaMA include, but are not limited to 1) LoRALLaMA, as demonstrated in experiments by (Pathak et al., 2023), exhibited effective text summarization through finetuning, indicating a broad range of potential applications. 2) LLaMAReviewer introduces a framework for automating code review processes using large language models (LLMs) and implementing parameter-efficient fine-tuning (PEFT) methods. This approach achieves high performance with less than 1% of trainable parameters. (Lu et al., 2023) experiments on LLaMAReviewer demonstrated that aligning the input representation with the pretraining format enhances the utilization of LLM capabilities. 3) FactLLaMA combines instruction-following language models with external evidence retrieval to improve fact-checking performance (Cheung and Lam, 2023).

Mathematical Reasoning and Answering The process of generating solutions for math word problems, including both the final answer and step-by-step explanations, involves the machine autonomously providing the solution. Numerous researchers, such as (Gaur and Saunshi, 2022), (Kai, 2023), (He et al., 2023), (Zheng et al., 2021), (Gandhi et al., 2022) and others, have presented diverse hybrid models that demonstrate superior performance in solving mathematical problems when compared to the existing large language models.

3 Approach

We believe that training a large language model (LLM) to reason through mathematical problems, rather than simply focusing on answer generation, can significantly improve performance. Our approach combines contextual reasoning, answer generation, and multi-task fine tuning. We hypothesize that this will enhance the LLM's ability to generalize solutions and overcome common challenges in mathematical question-answering tasks.

The Baseline model used For our baseline, we'll use the open-source LLaMA 2 7B (Touvron et al., 2023) an open-source model, We have selected the LLaMA 2 7B model as our baseline as it was trained on 2 trillion tokens, giving it a strong grasp of natural language, including the nuances of mathematical explanations. The model's substantial context length of 4086 tokens is ideal for understanding complex problems and extended reasoning steps in a single inference pass. Furthermore, recent studies (Liu and Low, 2023) have highlighted that Llama 2 7B can be easily fine-tuned using LoRA on a 24GB VRAM GPU, which also gives us an additional scope to increase our training to larger datasets. For Baseline evaluation, we will use the model to predict answers for the AQUA-RAT dataset, by excluding the rationale. we will use the LlamaTokenizer accompanied by the Base model for compatibility with its architecture.

Fine-tuning methodologies:- We aim to explore different rationale-based fine-tuning methodologies to enhance the performance of language models on the math QA task. Firstly, we will investigate the Task-Specific Fine-Tuning approach, wherein we pre-process the dataset by merging rationale (contextual information or supporting evidence) with the answer to create an updated answer. During fine-tuning, the model is trained using both questions and these refined answers, with the objective of augmenting the model's ability to generate accurate responses by incorporating relevant contextual information. To mitigate concerns related to inference where the model outputs both rationale and answer, we will explore additional techniques to isolate and extract only the answer. This could involve designing post-processing steps to ensure that the final output includes only the relevant answer. Furthermore, we plan to leverage metrics such as “BertScore” and ”RougeScore” to evaluate the effectiveness of training.

Subsequently, we will explore Multi-Task Fine-Tuning with dual forward propagation. Our strategy involves creating two data points for each Question-Answer example during training: one for predicting the rationale underlying the answer and another for predicting the final answer itself.

Dual propagation is employed during the forward pass, entailing the separate computation of losses for rationale prediction and final answer prediction. The total loss, a weighted sum of these losses, encourages the model to simultaneously reason and predict accurate answers. During prediction, we propose appending supplementary information to the question to ensure that the model predicts solely the final answer. We anticipate utilizing ”BertScore” and ”RougeScore” metrics for rationale evaluation and ”Accuracy” and ”F1 Score” metrics for final answer prediction during training.

Lastly, if time permits, we aim to investigate Sequential Fine-Tuning, initially fine-tuning the model using question and rationale pairs to enhance its understanding of the reasoning process. Subsequently, the model will undergo further fine-tuning using question and rationale pairs as input and answers as output. This sequential fine-tuning process aims to refine the model's ability to generate accurate answers based on learned reasoning patterns, optimizing its performance for math QA tasks. During training, ”BertScore” and ”RougeScore” metrics will be employed for the first fine-tuning and ”Accuracy” and ”F1 Score” will be utilized for subsequent fine-tuning.

For each of these approaches, we plan to employ evaluation metrics such as ”Accuracy,” and ”F1 Score” during testing phases. Additionally, we aim to implement ”Parametric Efficient Fine Tuning” methodologies keeping computing constraints in consideration.

In conclusion, we will compare the performance of each fine-tuning method with the base model and with each other to glean insights into the efficacy of different approaches.

3.1 Schedule

Our team has agreed upon the following schedule for the subtasks.

1. **Literature review and Data setup** (March 17th): Investigate loss function combination techniques for reasoning and answer loss (e.g., weighted sum).

2. **Zero shot Predictions** (April 6th)

   a. Baseline Model setup (1 week)

   b. Initial evaluation and benchmarking (1 week)

3. **Fine Tuning** (April 26th):

   a. Basic run and hyper-parameter optimization (1 week)

   b. Evaluation(training and testing) (2 weeks)

4. **Analysis and Documentation** (May 5th)

Problem 1:
Question: Two trains running in opposite directions cross a man standing on the platform in 27 seconds and 17 seconds respectively and they cross each other in 23 seconds. The ratio of their speeds is:
Options: A) 3/7 B) 3/2 C) 3/88 D) 3/8 E) 2/2
Rationale: Let the speeds of the two trains be x m/sec and y m/sec respectively. Then, length of the first train = 27x meters, and length of the second train = 17 y meters. (27x + 17y) / (x + y) = 23 → 27x + 17y = 23x + 23y → 4x = 6y → x/y = 3/2.
Correct Option: B

Problem 2:
Question: From a pack of 52 cards, two cards are drawn together at random. What is the probability of both the cards being kings?
Options: A) 2/1223 B) 1/122 C) 1/221 D) 3/1253 E) 2/153
Rationale: Let s be the sample space.
Then n(s) = 52C2 = 1326
E = event of getting 2 kings out of 4
n(E) = 4C2 = 6
P(E) = 6/1326 = 1/221
Answer is C
Correct Option: C

Figure 1: Examples from AQUA-RAT dataset.

4 Data

The AQUA-RAT (Algebra Question Answering with Rationales) dataset, as introduced in the paper by (Ling et al., 2017), serves as the foundation for our project experiments. This dataset is tailored for automating quiz-solving tasks, providing math questions alongside five possible options (labeled A to E), detailed rationales explaining the reasoning behind selecting the correct option, and the correct answer itself. In total, the dataset comprises 100,000 examples, occupying 52MB of storage. To ensure a balanced distribution of examples, the dataset is split into three subsets: training (60%), validation (15%), and test (25%). This partitioning ensures a sufficient number of examples in each subset for robust evaluation.AQUA-RAT offers a rich training ground for the CoT model. The diverse set of algebraic word problems and their corresponding rationales allows the model to learn various reasoning pathways for different problem types. By comparing the model's generated reasoning steps with the provided rationale, we can assess how well the model captures the thought process behind solving the problem.

5 Tools

We will utilize PyTorch for fine-tuning and implementing deep learning techniques. To leverage computational power, we intend to utilize Google Colab, which offers access to GPUs for training our models efficiently. Additionally, Git will be employed to maintain version history, facilitating collaboration and tracking changes throughout the project development process.

6 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - Yes

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - i am writing the introduction for my school project where the introduction in the project proposal should contain: Tell us what problem you're going to work on. Provide some motivation for your idea: why is it interesting? Does it have any practical significance? Do you think this write-up answers this? And is formal enough for an academic paper?
  - Paraphrase the following with better wording
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - For rewriting my introduction for any language issues, the response was pretty good. It cited some things that are already known and some unknown. It also gave a response editing my writing with it's own suggestions which was almost really good. I only had to make a few tweaks based on my style preference.
  - Related work: I have used AI assistance for paraphrasing and to get detailed information/explanation about a topic which I felt difficult to understand while reading publications.
  - ChatGPT is used in Data and Tools section for better formatting and wording.
  - For the Approach section, when asked ChatGPT to improve the wording, has used vocabulary that was not cohesive with the rest of the document. I had to reword it myself going through every word.

References

Cheung, T.-H. and Lam, K.-M. (2023). Factllama: Optimizing instruction-following language models with external knowledge for automated fact-checking. In *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 846–853.

Gandhi, J., Gandhi, P., Gosar, A., and Hole, V. (2022). Natural language processing based math word problem solver and synoptic generator. In *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 12–16.

Gaur, V. and Saunshi, N. (2022). Symbolic math reasoning with language models. In *2022 IEEE MIT Undergraduate Research Technology Conference (URTC)*, pages 1–5.

He, X., Gao, H., He, J., and Sun, C. (2023). Evaluation of large scale language models on solving math word problems with difficulty grading. In *2023 International Conference on Intelligent Education and Intelligent Research (IEIR)*, pages 1–5.

Kai, G. (2023). Bidirectional training for generating math word problems using pre-trained model and prompt. In *2023 20th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 1–6.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. (2023). Let's verify step by step. *arXiv preprint arXiv:2305.20050.*

Ling, W., Yogatama, D., Dyer, C., and Blunsom, P. (2017). Program induction by rationale generation : Learning to solve and explain algebraic word problems.

Liu, T. and Low, B. K. H. (2023). Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks.

Lu, J., Yu, L., Li, X., Yang, L., and Zuo, C. (2023). Llamareviewer: Advancing code review automation with large language models through parameter-efficient fine-tuning. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*, pages 647–658.

Pal, A., Umapathi, L. K., and Sankarasubbu, M. (2022). Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Flores, G., Chen, G. H., Pollard, T., Ho, J. C., and Naumann, T., editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Pathak, A., Shree, O., Agarwal, M., Sarkar, S. D., and Tiwary, A. (2023). Performance analysis of lora finetuning llama-2. In *2023 7th International Conference on Electronics, Materials Engineering Nano-Technology (IEMENTech)*, pages 1–4.

Rukmono, S. A., Ochoa, L., and Chaudron, M. R. (2023). Achieving high-level software component summarization via hierarchical chain-of-thought prompting and static code analysis. In *2023 IEEE International Conference on Data and Software Engineering (ICoDSE)*, pages 7–12.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Zhang, Z., Zhang, A., Li, M., and Smola, A. (2022). Automatic chain of thought prompting in large language models.

Zhao, H., Yilahun, H., and Hamdulla, A. (2023). Pipeline chain-of-thought: A prompt method for large language model relation extraction. In *2023 International Conference on Asian Language Processing (IALP)*, pages 31–36.

Zheng, D., Hu, J., Xie, Y., and Li, J. (2021). Math word problem solving with ensemble learning. In *2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pages 202–206.