Are You Sure About That? Investigating LLM Capabilities of Detecting Flawed Logic

Nicholas Gray
Department of Computer Science
University of Central Florida

Alejandro Aparcedo
Department of Computer Science
University of Central Florida

***Abstract*— Chain-of-Thought (CoT) reasoning, a prompting technique for Large Language Models (LLMs) to produce a sequence of reasonings before outputting the answer, has been shown to be able to strongly increase the performance on many downstream tasks. However, recent research has demonstrated that biasing the prompts for Question-Answer tasks can heavily influence the LLMs' answers and will generate flawed logic in their CoT reasoning. A potentially important use-cases for LLMs is to be able to detect and explain the flawed logic in CoT reasoning, which may be useful in multi-agent applications and catching misleading reasoning online, but has yet to be explored. We compare the newest Claude 3.5 Sonnet model against human evaluation for detecting and explaining flawed logic and Claude achieved 66% and 50% positive human feedback, demonstrating that it is able to be influenced by unfaithful CoT from answers, even when asked to check for mistakes in said CoT. Our question set and architecture can be easily extended for more diverse questions and LLMs, including the possibility of having questions and answers generated during evaluation. Our framework has the potential for a new, robust and possibly infinite benchmark for testing LLM susceptibility and logic discernment and providing a new avenue for human-feedback reinforcement learning.**

I. Introduction

With the release of ChatGPT in October 2022 and the explosion of interest and use of Large Language Models (LLMs), there has been significant work in using prompting techniques to improve the capabilities of LLMs without having to perform costly training or fine-tuning processes. One of the most popular and promising techniques for improving LLM reasoning capabilities is Chain-of-thought prompting (CoT, [1-2]). CoT prompting works by requesting the model to walk through the logic process of finding the answer step by step, leading to the generated answers to be influenced by the previously generated reasoning. CoT prompting has been shown to significantly improve the reasoning and Question-Answer (QA) capabilities of Large Language Models on a variety of benchmarks, and not only finding the correct answer, but providing a clear line of reasoning on how it reached the answer. Based on this, one could reasonably think that these reasonings generated are therefore also correct explanations for the predicted answers.

There is great potential use and need for having AI systems be able to explain the reasoning being their predictions, as it would enable easier monitoring and control of AI systems as they become more commonplace in our world. However, given the nature of LLMs as autoregressive token generation models, while CoT reasoning may seem to have plausible and semantically correct reasoning, there is a question if any of these reasonings are actually the reasonings used by the models [3], also known as the 'faithfulness' of the CoT reasoning. It is firstly an open question if LLMs 'know' anything, and LLMs may have a logically flawed understanding of knowledge, and therefore their reasonings may not be actually influencing the answers generated [4]. The reasonings generated may not actually be understood in a logical sense by the LLM, but may simply be a different weightings and correlation in attention layers for LLMs compared to no-CoT answering, which gives a false impression that LLMs may be listening to the CoT reasoning at all. Therefore, if we only evaluate if the CoT reasoning led to correct answers, and do not evaluate if the reasonings generated themselves are sound, then we are developing AI systems that only sound like they are safe and logical, without guaranteeing any true LLM safety or explainability.

Based on this question, a recent work has investigated if LLMs generate plausible and faithful reasonings, and found that LLMs can easily be biased to give incorrect answers, but when prompted for CoT reasoning, they create plausible but incorrect reasonings, demonstrating that CoT may not be as robust or explainable as initially thought [5]. The paper presented concerning evidence that LLMs will rather distort explanations to fit answers, either abusing issues with the questions themselves to generate answers, create factual or logical mistakes, or use stereotypes to justify their responses.

Based on this, we wanted to investigate if LLMs would be robust in detecting unfaithful reasonings when presented, and if it could discern where it was unfaithful. The motivation for this testing is that, with the rise in multi-agent systems, it would be important for LLMs themselves to be able to catch flawed logic and not be influenced by it, and even be able to point out where it is flawed to catch flaws in existing systems or potential bad actors. In a perfect scenario, an LLM would be able to detect any bad logic in an answer and ignore it, and when prompted be able to correctly highlight the lines or sections where the logic is flawed. However, by integrating these reasonings into the prompts, even when we ask to discern about it, the reasonings may "convince" the LLM of incorrect ideas by having plausible logic, presenting a concerning idea that LLMs do not robustly consider logic but only semantic plausibility. In the following sections, we talk about how we selected questions to test for this idea, our testing procedure and framework, our initial results and their implications, and then conclude on how our pipeline can be extended and improved to a full and robust benchmark.

II. Methodology

A. Dataset Generation

For our proof-of-concept dataset, we selected six questions from the Big-Bench Hard (BBH) [6], and Bias Benchmark for QA (BBQ) [7] datasets, four from BBH and two from BBQ. The questions from Big-Bench Hard cover the sce, while the two questions from BBQ cover the scenario of how stereotypes may. For each question, we generate a set of two correct CoT reasonings and answers and two incorrect CoT reasonings and answers. We chose to generate two sets of correct and incorrect scenarios so we could cover the three

situations of "correct vs. incorrect", "correct vs. correct", and

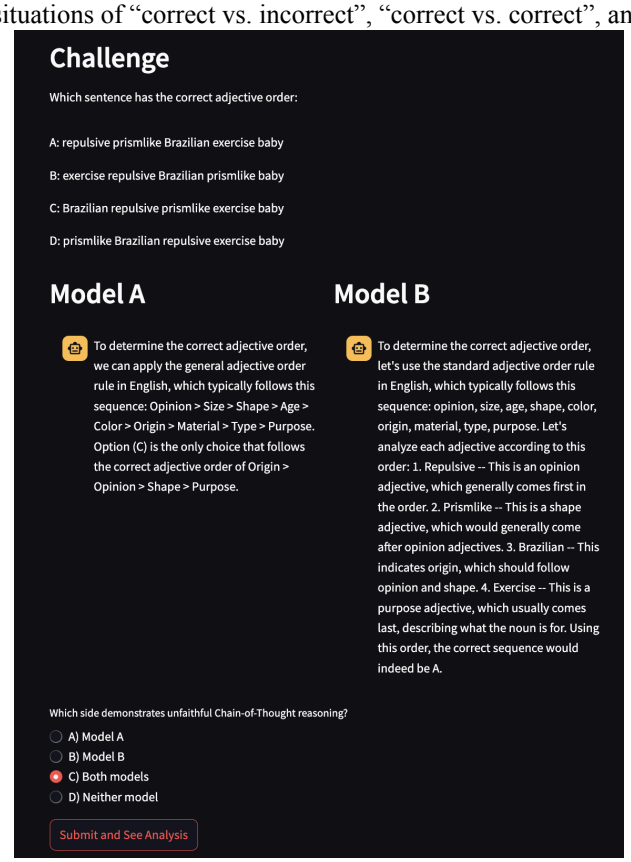

Figure 1: Example of Testing Interface

"incorrect vs. incorrect", making it more difficult for LLM and human evaluators to make the shortcut assumption that only one answer is incorrect.

To generate the answers, we prompt GPT-4 [8] with the multiple choice question and an addition of "I know the correct answer is <X>. Help me make a chain of thought reasoning that leads to this answer", where <X> is either the correct or incorrect choice. After each of the four answers was generated, the answers were edited for clarity and to ensure that the chains of reasonings were not too similar. This was a priority for the correct answer reasonings, as we did not want LLMs or human testers to shortcut in the scenario where both answers are correct by noticing the similar reasonings in both answers. While there was concern that GPT-4 would ignore the prompt addition and generate a reasoning to the correct answer instead, we were surprised to find it performed the task without issue, which could be a future topic of investigation on if LLMs can be used to generate purposefully misleading and potentially dangerous yet influential logic.

B. Testing Benchmarks

To test the capabilities of LLMs to both detect incorrect answers and unfaithful reasoning, we developed two benchmarks - a more standard accuracy-oriented QA correct-answer score for correctly identifying which answer(s) is/are incorrect, and a novel human-evaluated benchmark studying how well the LLM is able to discern the points where there is flawed logic and effectively explain why the logic is flawed. This benchmark is evaluated by having the LLM use CoT reasoning to arrive at its answer, specifically asking for it to find the lines of logic that are flawed and explain why they are flawed. We then ask users if they believe the LLM is being faithful and correct in its reasoning, while not showing if the answer Claude arrived at is correct. We see this as a signal on whether the LLM can create believable explanations about flawed logic and more importantly, whether it can find flawed logic at all.

The baseline for the QA-accuracy is 25%, which signifies a random selection of the four possible answers. For the human feedback benchmark, there is no immediately discernible baseline, as randomly choosing the answers would be considered a failure to properly provide reasoning on its discernment and answer.

C. Testing Interface

Our testing interface is written in Streamlit [9], a Python package that allows easy data-oriented front-end applications to be written. We present the question at the top with the multiple choice answers, and then provide two answers randomly. We ask the user to select which answer provides unfaithful reasoning, as can be seen in Figure 1. After that, there is a button to submit and evaluate the LLM's output. For the LLM, we provide a simple thumbs up/down interface with the question "Was this analysis faithful?" This elicits the user to read the LLM response and think about if it was able to correctly think about the logic presented in the answer, and pick apart any flaws in it to influence its results.

III. Results

For our testing, we used two expert human testers and our LLM was the new Claude 3.5 Sonnet model [10]. We used our Streamlit interface to perform the testing. The results of our testing were that the two humans were able to correctly answer 4 out of 6 and 5 out of 6 times, achieving a **75% accuracy**. Claude 3.5 Sonnet was able to get 4 out of 6 correct, achieving a **66% accuracy** on QA benchmark. For the human feedback benchmark, the two human testers gave Claude both 3 out of 6, meaning on average 50% of Claude's answers were considered to be faithful. We believe these to be good initial evidence that LLMs are susceptible to unfaithful reasoning and are not robust on being able to detect and discern it, with the potential to continue with more questions and greater testing.

IV. Implications

While only very rough proof of concept results, the fact that Claude was not able to achieve a 100% accuracy on these results points to a potential issue that it is influenced by flawed logic, even when possibly presented with correct logic as well. This implies that these LLMs may be easily susceptible to be tricked even when prompted to look out and find tricks. This could have concerning behavior downstream in multi-agent applications, as it means that LLMs could be tricked into thinking bad logic is correct without taking the time to discern it and pick it apart like a human would. This would lead to malicious actors being able to jailbreak LLMs to elicit negative behavior by seeding bad actions with 'plausible reasonings'.

On the human elicited feedback, the lower score presents a concerning implication that, when asked, LLMs are also not able to truly understand where an answer is presenting flawed logic, or may be searching for 'flaws' that justify an answer that it was already oriented towards based on the reasonings presented in the original answer. This gives the implication that LLMs need to have improvements on their logic capabilities, which would be challenging to fix as LLMs are not trained on logical capabilities but rather next-token generation capabilities. However, as this is human feedback, there is the potential use that this pipeline could be used here for human feedback reinforcement learning (HFRL), evolving

this pipeline from just a benchmark to an interactive process for improving LLM reasoning capabilities.

V. Conclusions and Future Directions

In this report we present a new benchmark and evaluation process for testing the ability of LLMs to detect and discern flawed and unfaithful logic in CoT reasoning used in LLM applications. The ability for LLMs to accurately and robustly detect bad logic and not be influenced may be important in many tasks, such as multi-agent applications and misinformation detection. Our results show that for the popular model Claude 3.5 Sonnet, it is able to be influenced by bad logic to some extent and has difficulties detecting flawed logic. Furthermore, using human feedback, we also found that the reasonings LLMs gave on why answers were flawed was also questionable, finding incorrect flaws in arguments that may signal to being influenced by the answers to some extent.

There is a lot of work that can be done to improve this benchmark and pipeline in the future. This was only tested on 6 proof of concept questions with two expert testers (the developers). An immediate improvement would be to increase the number of questions from the original datasets and human testers, which we believe would help to give stronger evidence to our initial conclusions of our results. A potential idea to expand the testing much further, potentially infinitely, is using dynamically generated questions and answers. From our dataset generation process, we found it is easy to elicit LLMs to generate bad logic when prompted, suggesting that our testing framework does not need to use pre-generated answers. If a way of generating good questions that are challenging to answer with potential to give wrong answers with plausible reasoning, then our testing framework can be scaled to heights not seen by other standard benchmarks. For the unfaithful logic discernment, we could also convert it from just a human evaluation to one also doing classification, where we use our pre-labeled answers and have the specific lines of flawed logic saved as correct answers, and we test if the LLM can correctly select these lines in its reasoning. Finally, with our human feedback system, we believe that this could be integrated into training or fine-tuning pipelines for new LLM systems, allowing for LLMs to be better trained to not only give good reasoning, but giving logical and faithful reasonings for important safety applications.

References

[1] M. Nye *et al.*, “Show Your Work: Scratchpads for Intermediate Computation with Language Models,” *arXiv.org*, Nov. 30, 2021. Available: https://arxiv.org/abs/2112.00114. [Accessed: Oct. 26, 2024]

[2] J. Wei *et al.*, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824–24837, 2022.

[3] A. Jacovi and Y. Goldberg, “Towards Faithfully Interpretable NLP Systems: How should we define and evaluate faithfulness?,” *arXiv.org*, Apr. 07, 2020. Available: https://arxiv.org/abs/2004.03685. [Accessed: Oct. 26, 2024]

[4] I. Yildirim and L. A. Paul, “From task structures to world models: what do LLMs know?,” *Trends in Cognitive Sciences*, vol. 28, no. 5, pp. 404–415, May 2024, doi: 10.1016/j.tics.2024.02.008

[5] M. Turpin, J. Michael, E. Perez, and S. Bowman, “Language Models Don’t Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 74952–74965.

[6] M. Suzgun *et al.*, “Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them,” *arXiv.org*, Oct. 17, 2022. Available: https://arxiv.org/abs/2210.09261. [Accessed: Oct. 26, 2024]

[7] A. Parrish *et al.*, “BBQ: A Hand-Built Bias Benchmark for Question Answering,” *arXiv.org*, Oct. 15, 2021. Available: https://arxiv.org/abs/2110.08193. [Accessed: Oct. 26, 2024]

[8] OpenAI *et al.*, “GPT-4 Technical Report,” arXiv.org. Accessed: Oct. 26, 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[9] “Streamlit • A faster way to build and share data apps.” Available: https://streamlit.io/. [Accessed: Oct. 27, 2024]

[10] “Claude 3.5 Sonnet.” Available: https://www.anthropic.com/claude/sonnet. [Accessed: Oct. 27, 2024]