CSE584: Final Project

Analyzing LLM Behavior on Faulty or Invalid Science Questions

Wahid Uz Zaman
wzz5219@psu.edu

Project Description:
Collect or create a set of faulty science questions that can fool a top-performing LLM (e.g., ChatGPT, GPT4, Gemini-1.5-Pro, or Claude-3-Opus, etc.)Then, design some research questions based on your dataset and conduct experiments to explore them.

Dataset Creation:
I have created 160 Questions from 22 different branches of Science. I prompted those questions to 4 Top LLMs. ChatGPT, Gemini-1.5-pro, Llama 3.1 70b-versatile, MistralChat. Each of these questions fools one or more of these LLMs. The dataset is already uploaded to GitHub.

Research Questions:

At first, I used original questions as prompts for ChatGPT and Gemini Pro. These LLMs can understand nearly 40% of the questions as invalid/faulty.

Then, I considered whether prompt engineering could be used to modify an LLM's response. However, I wanted to avoid crafting individual prompts for each faulty question. So, I explored whether adding a general suffix to the original faulty question could help the LLM recognize that the question itself is invalid or faulty.

This led to my research question:

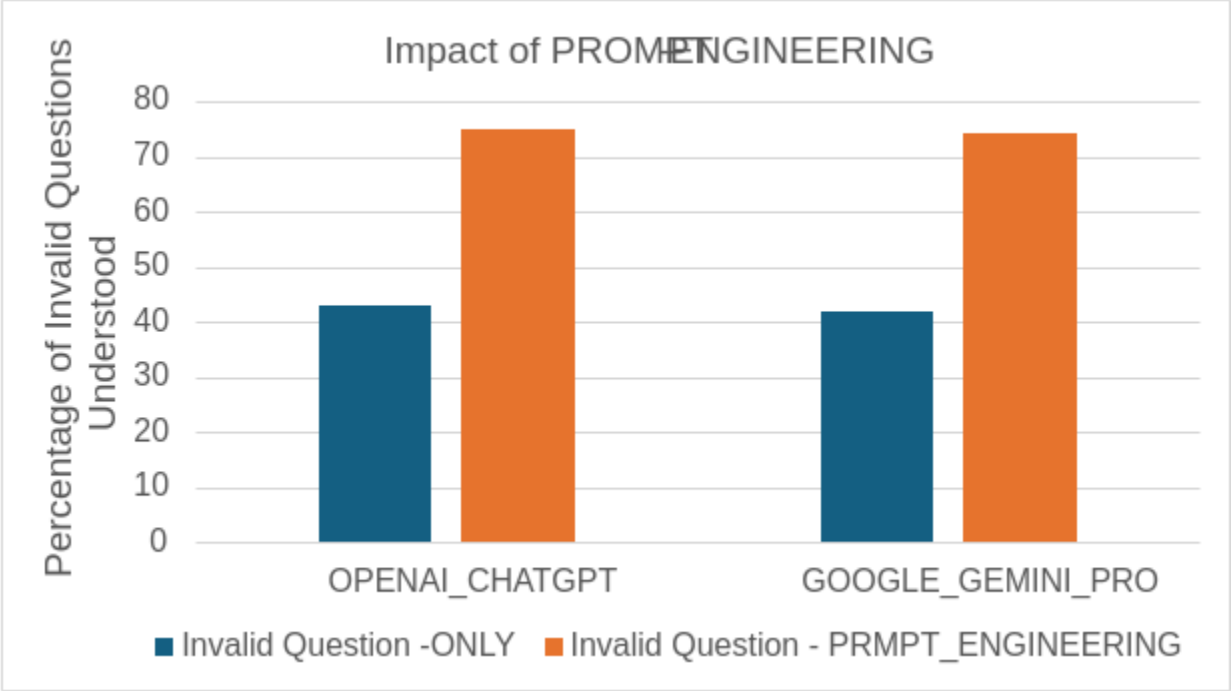
“Can general and simple prompt engineering enable LLMs to recognize invalid or faulty questions?”

So, for those 160 questions, I made modified prompts in this format:

Original question + “. Please make sure the question is valid beforehand.”

For both LLMs, their responses nearly doubled in accuracy. With general prompt engineering, 75% of the responses demonstrated that these LLMs can identify the problem as faulty, misleading, or impractical.

The result is shown in the below graph:



Although these LLMs do not exhibit consistent behavior across questions (e.g., ChatGPT may understand certain questions with prompt engineering while Gemini does not, and vice versa), their responses, on average, improve when using a general suffix prompt that explicitly instructs the models to check for the validity of the questions.

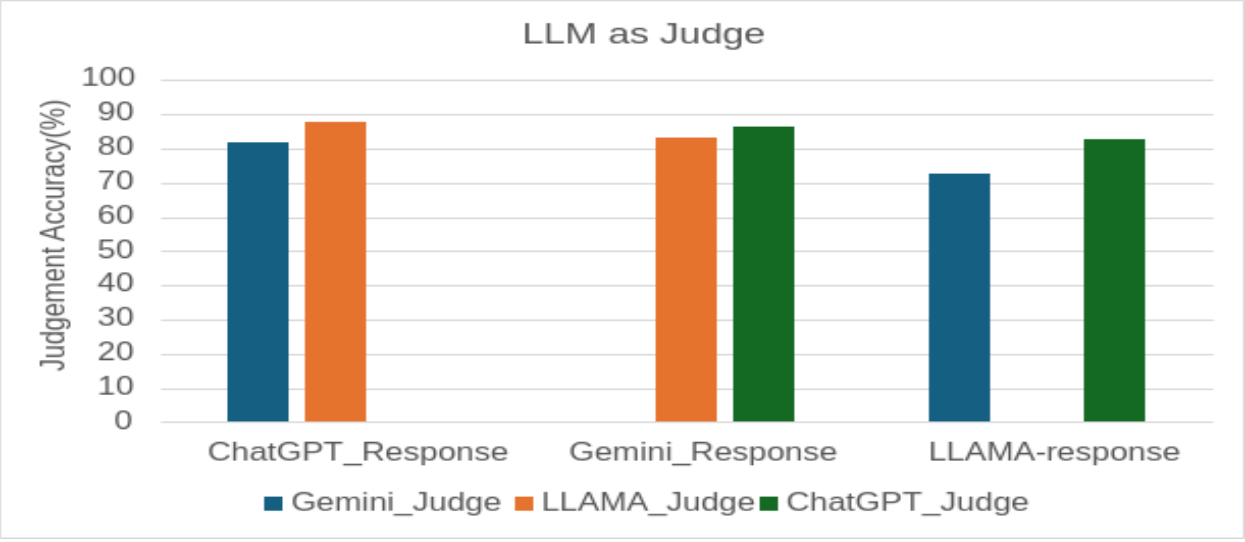
In the first experiment, I manually evaluated all the responses, effectively acting as a judge. This process was tedious and time-consuming, leading to my second research question:

“Can the responses of one LLM be evaluated by another LLM to determine if the first LLM recognized the original question as faulty?”

To do this, let’s say, ChatGPT’s responses will be evaluated by Gemini. The process involves the following steps:

1. **Obtain ChatGPT's Response**:
   Prompt the original question to ChatGPT and retrieve its response.
2. **Construct Evaluation Prompt**:
   Create a prefix and suffix as follows:
   a. **Prefix**: “I asked an invalid question to an LLM, meaning the question was attempting to fool the LLM. The question and the LLM's response are: “
   b. **Suffix**: “Now, can you tell me, from the LLM's response, whether that LLM recognized that the question was invalid? Just answer 'yes' or 'no'.”
3. **Ask the Evaluator (Gemini)**:
   Combine the prefix, ChatGPT's response, and the suffix to form the evaluation prompt:
   `prefix + response + suffix`
   This prompt is then sent to Gemini for evaluation.
4. **Assess Gemini's Judgement**:
   Compare Gemini's response against the intended judgment:
   a. **Correct Judgement**: Gemini says "yes" if ChatGPT's response demonstrates an understanding of the invalid question.
   b. **Correct Judgement**: Gemini says "no" if ChatGPT's response does not reflect an understanding of the invalid question.

Experimental results are given below:



I utilized the Gemini and LLAMA-3.1 70B versatile models to assess ChatGPT's responses to original faulty questions. Similarly, to evaluate Gemini's responses, I used ChatGPT and LLAMA-3.1 70B, and for LLAMA's response evaluation, I employed both ChatGPT and Gemini. The figure above shows that in all cases, accuracy exceeded 80%, except for Gemini's evaluation of LLAMA's response, which was around 72%. For improved results, I believe larger datasets are needed.

I also conducted experiments with specific types of questions, such as:

'If wave speed is 5.8 × 10^8 m/s and frequency is 6 × 10^14 Hz, what is the wavelength?'

In this case, all the LLMs perform the math correctly but fail to recognize the issue that the given wave speed exceeds the speed of light. I even tried prompt engineering, but the LLM still couldn't identify the problem in the question. Since I didn't perform any numerical evaluations, I couldn't include graphs here.

All the experiment data files and code are uploaded on GitHub.