Do LLMs answer faulty questions? Unfortunately, yes!

Introduction

In this project, we collect science questions that are not valid or faulty to answer that a human with enough knowledge would refuse to answer. These questions need to fool large language models (LLMs). One Math example for this is “Lily received 3 cookies from her best friend yesterday and ate 5 for breakfast. Today, her friend gave her 3 more cookies. How many cookies does Lily have now?” This question does not make sense because Lilly initially had 3 cookies but ate 5 afterward that is not possible. So a human or an LLM is supposed to refuse to answer this question, state that the question is wrong, or state an answer that is not listed in the options. We collect faulty multiple choice questions about eleven science domains such as medicine, material science, or biology that GPT4o (Achiam et al., 2023) answers. The reason for being faulty is not having the correct option in the answers so the expected behavior is saying none of the above or stating the question is faulty etc. After we have the data, we answer the following research questions: (1) how do different state-of-the-art LLMs respond to these faulty questions, (2) how can we encourage models to state the answer is faulty or none of the options are correct? Experimental results show that LLMs do not recognize the invalidity/incorrectness of questions even if they are explicitly instructed or a new option as none is added. Although we do not propose sophisticated approaches to encourage models, it would be a better idea to solve this issue during RLHF or finetuning process.

Dataset

We sample data from MMLU (Hendrycks et al., 2020), MMLU-Pro (Wang et al., 2024), and Sciassess (Cai et al., 2024). We mostly sample data from MMLU-Pro for ten categories: biology, psychology, mathematics, electrical engineering, computer science, physics, chemistry, econometrics, medicine, and professional law. We sample data for material science from Sciassess. All of these questions are multiple-choice with a different number of options. We delete the correct option so the question becomes faulty. After this automated process, we manually check the questions to see if there are some errors, for example; some questions ask what best describes…. In those cases, there might still be a correct option because essentially the question asks the best among the options. Therefore, we discard those cases so there is no subjectivity in the question. Afterward, we prompt GPT4o and discard questions that do not fool it (i.e. the model says the questions are faulty etc.). Then, we sample more data from MMLU if needed to make the number of questions per category balanced. We present the statistics about the dataset in Table 1. The average number of options for each question is 7.7 with a standard deviation of 2.2. Material Science category is the only one that has only 3 options after the correct option is excluded. Others have at least 7 options on average. The differences between categories and within a category bring diversity into the dataset to challenge LLMs more.

Category	**Number of Examples**	**Avg (std) number of options**
Biology	45	7.3 (2.4)

Psychology	45	7.7 (1.7)
Mathematics	45	8.8 (0.8)
Electrical Engineering	45	8.4 (1.5)
Computer Science	45	8.2 (1.7)
Physics	45	8.9 (0.4)
Chemistry	45	8.6 (1.3)
Econometrics	45	7.9 (2.1)
Medicine	45	8.5 (1.2)
Material Science	45	3.0 (0)
Professional Law	45	6.9 (2.4)
Total	495	7.7 (2.2)

Table 1: Statistics about the dataset. The number of options does not count the correct answer.

Experiments and Results

We experiment with GPT4o (Achiam et al., 2023), Claude3.5 Sonnet (Anthropic, 2024), and Llama3.1-70b (Dubet et al., 2024) models. We first prompt the models naively by just giving the options and asking them to answer in a zero-shot setting (*naive*). In that case, GPT4o always answers the questions because of our data collection process. To find some answers to our second research question, we propose two different techniques: (1) instructing model to encourage to say the question is not answerable (*instructed*) (we add *"If the question does not make sense or is not answerable due to a logical error etc., please respond with 'This question is not answerable'"* into our prompt.), (2) we add a new option as 'none of the above' (*with none option*). All prompts can be found in Appendix. We report accuracy which means the fraction of examples where the model states the question is faulty, none of the options are correct, or, states an answer that is not in the given options. Intuitively, with the "with none option", the question becomes valid and models should be able to choose that option. However, Xu et al., 2024 show that this is usually not the case.

We report the accuracy for each model and prompting in Table 2. With a naive approach Llama3.1-70b model understands the invalidity of 5% of questions, the others being worse. On the other hand, when we explicitly tell models that the question might be faulty, GPT4o and Sonnet detect 10% of the questions as faulty, whereas Llama70b finds only 6%, a 1% increase from the naive version. This might indicate that GPT4o and Sonnet might have better instruction following capabilities. When we add the none option, GPT4o finds 23% of the examples as faulty which is a 13% increase from the instructed version. The other two models gain little even with this version. This indicates some problems regarding the evaluation of LLMs because the models may not understand the questions well. These findings support the findings of Xu et al., 2024.

Model	Naive Acc	Instructed Acc	With None Option Acc
GPT4o	0	**0.1**	**0.23**
Sonnet	0.03	**0.1**	0.13
Llama70b	**0.05**	0.06	0.07

Table 2: Results with the original prompt (naive, without any instruction to encourage none), instructed prompt (encouraging model to state none, the question is wrong etc.), with none of the above as an option.

In addition to overall results, it is important to see the differences between domains. We report accuracy by domain in Table 3. The effect of different prompting varies from domain to domain significantly. For Biology and Medicine, results are correlated with the overall results, adding none option outperforms the other two and instructing model outperforms the naive version. Furthermore, for CS and Math, results are similar but the effects are less strong. However, there is no single pattern for the other domains, they vary from model to model. The bigges improvement is observed ofr GPT4o by adding none option on Material Science domain with 44% increase. Another interesting observation is that by adding an instruction to encourage model that there might be faulty questions sometimes hurt the performance for Llama70b model. Moreover, adding none option also sometimes hurts performance for both Sonnet and Llama70b model.

Model-Config	Mat. Sci.	Bio.	Chem.	CS	Math.	Med.	Phys.	Econ.	EE	Law	Psy.
gpt4o-naive	0	0	0	0	0	0	0	0	0	0	0
gpt4o-instr.	0.31	0.13	0.11	0.04	0.04	0.15	0.02	0.07	0.09	0.0	0.11
gpt4o-none	0.44	0.40	0.20	0.13	0.13	0.35	0.18	0.18	0.22	0.09	0.20
sonnet-naive	0.02	0.04	0.04	0.04	0	0.04	0.02	0	0.07	0	0
sonnet-instr.	0.15	0.18	0.11	0.09	0	0.15	0.02	0	0.18	0.13	0.04
sonnet-none	0.29	0.31	0	0.09	0.04	0.20	0.07	0.07	0.13	0.09	0.13
llama-naive	0.09	0.04	0.07	0.02	0.02	0.04	0.11	0.02	0.07	0	0.04

llama-instr.	0.15	0.13	0.04	0.02	0.02	0.09	0.04	0.02	0.02	0	0.04
llama-none	0.11	0.18	0.09	0.04	0.04	0.07	0.07	0	0.09	0.02	0.02

Table 3: Accuracy results per category and prompt method.

Conclusion and Future Work

In this work, we show that different top-performing LLMs behave differently to faulty questions. In addition to this, we show that we can push models to refuse to answer faulty questions, or state another option or none by explicitly prompting them or adding none as an option. However, these two approaches are not enough to make models more reliable because our best result is getting only 23% of examples as faulty. The results underscore the overconfidence of models and the difference between the instruction following capability of models.

Future work might focus on more sophisticated prompting techniques, improving finetuning and RLHF process. We have also tried chain-of-thought prompting but the model needed to produce very long contents until they state the answer so we could not finish those experiments because of cost issues. That might help too. Another approach might be a two-step approach where in the first stage the model is asked if the question is faulty or not and the answer can be asked only if it is found valid.

Lastly, our work is limited in terms of the approaches we try to improve the issue found in this paper (LLMs cannot understand the invalidity of questions) due to time constraints. Furthermore, we only focus on one type of faulty question where the correct option is not given in a multiple-choice setting.

References

1. Cai, H., Cai, X., Chang, J., Li, S., Yao, L., Wang, C., ... & Ke, G. (2024). Sciassess: Benchmarking llm proficiency in scientific literature analysis. *arXiv preprint arXiv:2403.01976*.
2. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2020). Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
3. Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., ... & Chen, W. (2024). Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.
4. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
5. Anthropic, A. I. (2024). Claude 3.5 sonnet model card addendum. *Claude-3.5 Model Card*, *3*.
6. Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., ... & Ganapathy, R. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
7. Xu, H., Lou, R., Du, J., Mahzoon, V., Talebianaraki, E., Zhou, Z., ... & Yin, W. (2024). LLMs' Classification Performance is Overclaimed. *arXiv preprint arXiv:2406.16203*.

Appendix

Our system prompt is always “You are an expert scientist.”

User Prompt for the Naive Approach:

```
"""Your task is to answer the given question. Do not give any reasoning for your
answer

      Respond in the format:
      Answer: [answer]

      \n\Question: {question}\nOptions: {option_text}\n\n
      """
```

User Prompt for the Instructed Approach:

```
"""Your task is to answer the given question. Do not give any reasoning for your
answer.
      If the question does not make sense or is not answerable due to a logical
error etc., please respond with 'This question is not answerable'.

      Respond in the format:
      Answer: [answer]

      \n\Question: {question}\nOptions: {option_text}\n\n
      """
```

User Prompt for the None of the options Approach:

```
"""Your task is to answer the given question. Do not give any reasoning for your
answer

      Respond in the format:
      Answer: [answer]

      \n\Question: {question}\nOptions: {option_text}\n\n
      """
```

‘Question’ is the question itself and ‘option_text’ is each option in the question separate by lines.