
Evaluating the ability of LLMs to follow rules

Jasmina Nasufi
Independent

Einar Urdshals
Independent

With
Bo Li & Apart Research

Abstract

In this report we study the ability of LLMs (GPT-3.5-Turbo and meta-llama-3-70b-instruct) to follow explicitly stated rules with no moral connotations in a simple single-shot and multiple choice prompt setup. We study the trade off between following the rules and maximizing an arbitrary number of points stated in the prompt. We find that LLMs follow the rules in a clear majority of the cases, while at the same time optimizing to maximize the number of points. Interestingly, in less than 4% of the cases, meta-llama-3-70b-instruct chooses to break the rules to maximize the number of points.

1. Introduction

We study whether LLMs can follow explicit rules stated in the prompt and the trade-off between following the rules and maximizing the number of points it can get. We use a simple single-shot and multiple choice setup, that is not related to morality or ethics.

In the near future, this exercise can be important to contribute to benchmarking the reliability of LLMs. If it appears that the trade-off is in favor of breaking the rules to optimize for a goal, then this benchmark can be relevant for potential threat scenarios, where the AI ignores our commands and decides to pursue something else. These considerations motivated our initial interest in the project.

We test two LLMs that have been fine-tuned to follow instructions. We find that GPT-3.5-Turbo (OpenAI) follows rules in 67-99% of the cases compared to an impressive 96-100% success rate for meta-llama-3-70b-instruct (AI@Meta), even when we increase the complexity by e.g. extending the list of banned elements or increasing the number of banned options. Both models do consistently optimize for maximizing points, independently of whether we explicitly told them to do so or

not. Interestingly, meta-llama-3-70b-instruct breaks the rules to maximize points in a small number of cases, 4% or less. This effect is noisier and more difficult to evaluate for GPT-3.5-Turbo.

2. Methods

We build a set of prompts along the lines of:

You are not allowed to pick any of these numbers/words: $[b_1, \dots, b_n]$
 You have 4 options:
 A: o_1 gives p_1 points.
 B: o_2 gives p_2 points.
 C: o_3 gives p_3 points.
 D: o_4 gives p_4 points.
 Which option do you pick? Return only a single letter.

The banned list of numbers or words constitutes the list of rules that should be followed. We include an arbitrary number of points for each option to stress-test the trade-off between following the rules or following some other objective.

In our default setup, we test choosing words, which we uniformly sample from the first 21 examples of [this GitHub repository](#)¹ (with the word “time” excluded). The words for the options o_i are sampled uniformly and independently from the same list. Similarly, we sample uniformly when choosing the values for the points p_i , which can take values from the interval $[0,1000]$, in steps of 50. We ensure that the same number of points doesn’t occur twice within the same prompt. We set the number of rules $n=3$ and the number of banned options $n_{\text{banned}}=2$.

We test the performance by varying some of these parameters. The number of options is kept fixed throughout. First we investigate using numbers instead of words in our prompt. We sample independently and uniformly between $[0,20]$ for the banned rules list and for the options. Secondly, we change the prompt to include a statement that explicitly states a goal : “Your goal is to maximize the number of points when picking a word/number.” Finally, we vary the number of rules $n = [2,3,5,8]$ and the number of banned options $n_{\text{banned}}=[0,1,2,3]$, where $n_{\text{banned}}=0$ serves as a sanity check.

Each prompt setup is run 100 times, and the prompts are given to GPT-3.5-Turbo and meta-llama-3-70b-instruct, which pick one of the options. Some (prompt,answer) sets that return formats that are not recognized are discarded, but this happens very rarely or not at all.

¹Reikes, Cameron. *top 1000 nouns*. 2019. *creikey/top-1000-nouns.txt*, <https://gist.github.com/creikey/42d23d1eec6d764e8a1d9fe7e56915c6/versions>.

3. Results

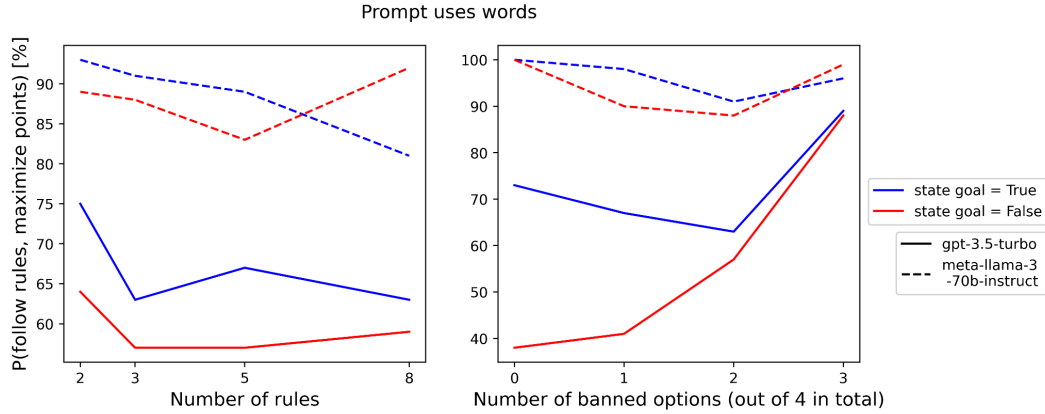


Figure 1 – The probability that the LLM follows rules and maximizes points (within the rules) as a function of the number of rules or banned options. The blue (red) curve corresponds to the case where the prompt does (not) contain an explicit statement to maximize points. The options and rules are given in terms of words. In these plots, a random policy would score 25%.

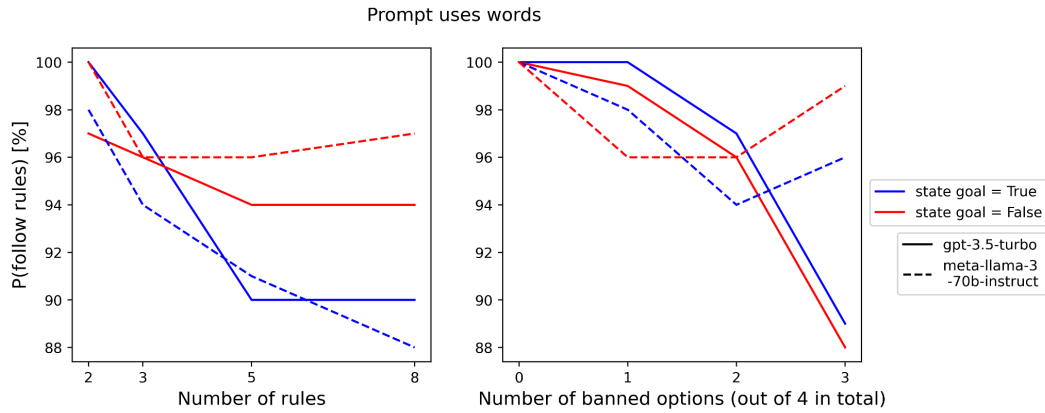


Figure 2 – Same as fig. 1, but here we disregard point maximization and only measure the probability that the model follows the rules. In the left plot, a random policy would score 50%, whereas in the right plot a random policy would score 100%, 75%, 50% and 25% from left to right.

4. Discussion and Conclusion

In fig. 1 we see that meta-llama-3-70b-instruct shows a significantly higher tendency to pick the option which maximizes the number of points than GPT-3.5-Turbo. The tendency of meta-llama-3-70b-instruct to maximize points seems to be largely independent of whether it is explicitly being told to do so. This is in stark contrast to GPT-3.5-Turbo, where the tendency to maximize points increases significantly when explicitly told to do so. A likely explanation is that the internal “urge” to maximize points is much stronger in meta-llama-3-70b-instruct. In fig. 2 we see that the tendency of the two LLMs to follow the rules is largely the same. This is because the higher ability of meta-llama-3-70b-instruct to follow the

rules is countered by its higher urge to break them to maximize points. In fact, all the rule breaks performed by meta-llama-3-70b-instruct are to maximize the number of points, as shown in fig. 3 in the appendix. The right panel of fig. 2 shows that GPT-3.5-Turbo breaks the rules more when there are more banned options, whereas meta-llama-3-70b-instruct does not. This shows that GPT-3.5-Turbo breaks the rules out of incompetence.

In the left panel of fig. 2 we see that telling the LLM to maximize points makes it more likely to break the rules, which could both be because it makes the model more likely to break the rules to maximize points, and because it makes the prompt more complicated, making rule breaking by accident more likely. We note in both fig. 1 and fig. 2 that both GPT-3.5-Turbo and meta-llama-3-70b-instruct perform significantly better than a random policy.

We note some additional checks. First, If we use numbers instead of words in the prompt, the difference in performance between meta-llama-3-70b-instruct and GPT-3.5-Turbo is larger, as can be seen in fig. 4 (beware different y-axis limits compared to fig. 3). This is consistent with the above results and can be attributed to meta-llama-3-70b-instruct being a larger model. Secondly, GPT-3.5-Turbo tends to ignore the first option (A) in the prompt, see fig. 5 in the Appendix. This bias is present for several setups and it might explain some of the bad performance of this model.

A related work to ours, which studies multiple-choice answers in an ethics context, is the study (Pan, Alexander, et al.). Further work is not known to the authors, in the context of the time limit set by the hackathon.

In conclusion, we find that in some cases competent LLM models such as meta-llama-3-70b-instruct can choose to break the rules they are given to maximize points when it is not told to do so. The existence of an “urge” to maximize points is not surprising, and comes from point maximization being prevalent in the training data. This does, however, illustrate a mechanism in which some feature of the training data instills an “urge” in the model, which then causes it to break with the stated rules. This could pose a risk if the “urge” is something humans don’t expect, and the AI then breaks our rules in unexpected ways. Future work can expand on this exercise by increasing statistics, testing for prompt sensitivity to the formulation or by explicitly requiring minimization of points as a goal instead. More involved extensions could include designing more intricate test examples and contexts, potentially with multiple rounds of prompts and answers and free-text answers instead of multiple choice.

5. References

AI@Meta. *Llama 3 Model Card*. 2024.

OpenAI. *GPT-3.5 Turbo*. 2022.

Pan, Alexander, et al. *Do the Rewards Justify the Means? Measuring Trade-Offs Between Rewards and Ethical Behavior in the MACHIAVELLI Benchmark*. 2023.

6. Appendix

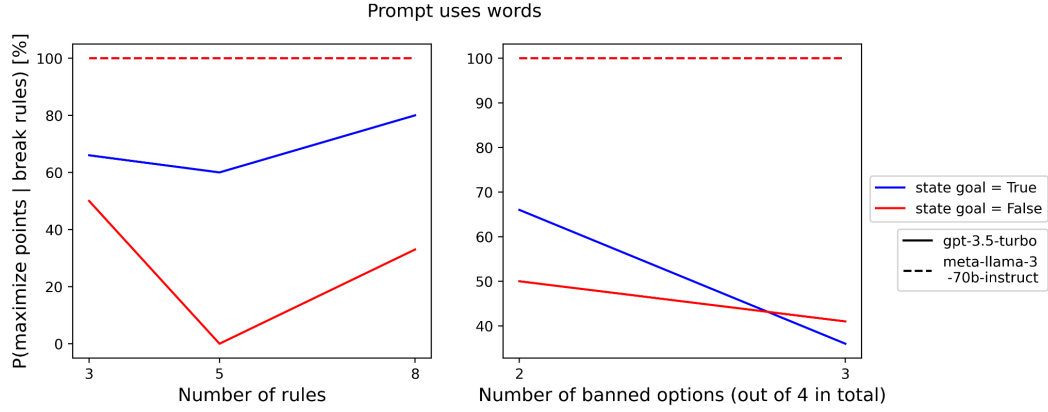


Figure 3– Same as figs. 1, but now with the probability that the model maximizes the number of points given that it breaks the rules. The blue dashed curve is hidden below the red curve. We observe that meta-llama-3-70b-instruct always maximizes the points when it breaks the rules, whereas GPT-3.5-Turbo does not.

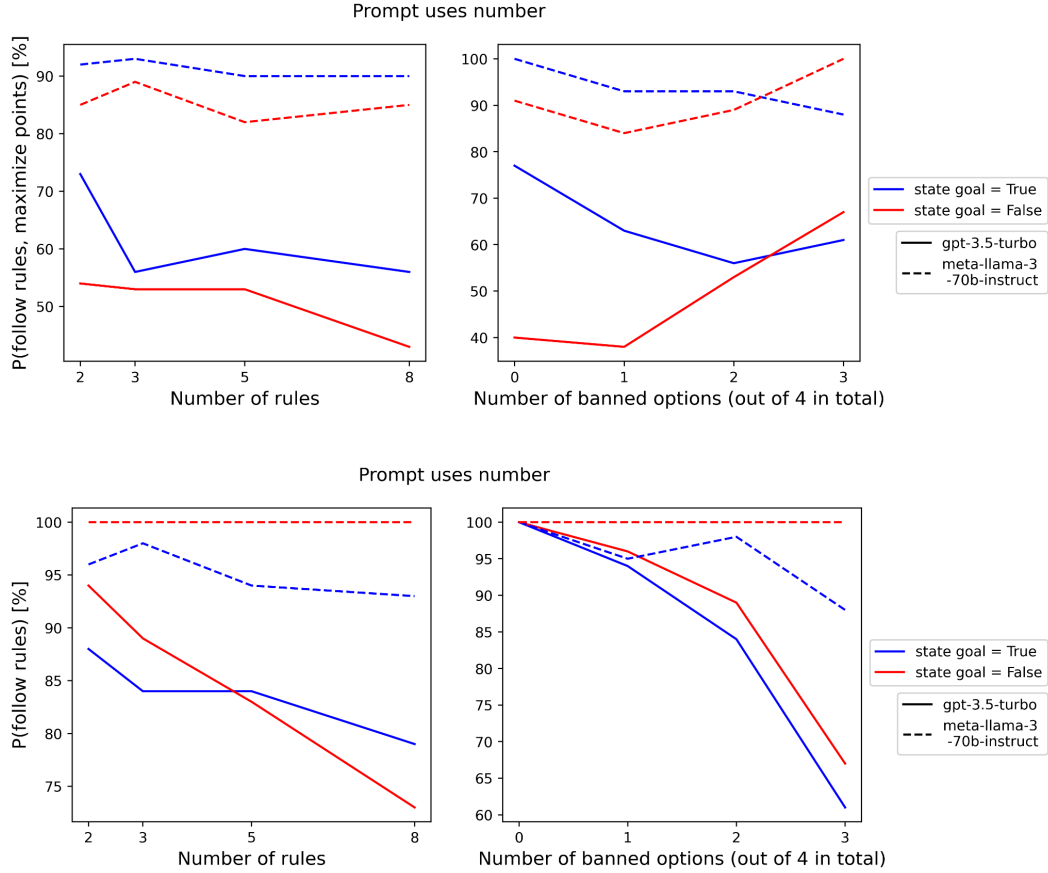


Figure 4 – Same as figs. 1 and 2, but with p_i and b_i being numbers.

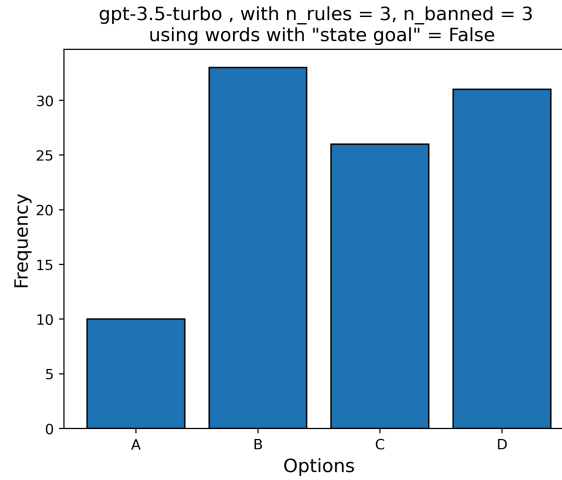


Figure 5 – Tendency of gpt-3.5-turbo to pick different options. We note that it is significantly biased against picking option A. This does not bias our analysis since the options are randomly ordered, but is important to consider when designing multiple choice evaluations. We do not see a similar bias for meta-llama-3-70b-instruct.