Ideas for the Evals Hackathon

by Marius Hobbhahn, Jérémy Scheurer, Mikita Balesni

General Suggestions

For the resources page, we would add the following papers, which could be very helpful to people who are new to the evals field:

- Discovering Language Model Behaviors with Model-Written Evaluations: This is defacto one of the most influential papers in the evals field (done by Ethan Perez), which introduced the concept of leveraging LLMs to generate evaluation datasets.
- Red Teaming Language Models with Language Models: Another very influential paper by Ethan. Here he also leverages LLMs to Red Team LLMs (whereas in the above paper he "evaluates" LLMs).
- Evaluating Language-Model Agents on Realistic Autonomous Tasks and Large Language Models can Strategically Deceive their Users when Put Under Pressure, these papers by ARC Evals and Apollo Research evaluate LLMs as Agents. I think they are very relevant to the evals hackathon because they show that evaluating LLMs in very specific, but highly realistic scenarios can lead to valuable insights about LLM capabilities and their safety risks. The first paper does it for Autonomous Replication and Adaption, and the latter does it for strategic deception. We think that finding similarly interesting scenarios as in these papers, evaluating LLMs, and doing various ablations (e.g. changing system prompts, etc.) is very feasible for a single weekend.
- Also here are some additional papers that might be useful to people:
    - How to Catch an AI Liar: Lie Detection in Black-Box LLMs by Asking Unrelated Questions
    - Taken out of context: On measuring situational awareness in LLMs

Hands-on approach to evals (Marius)

One possible way to start with evals is to “Just measure something and iterate”. The broad recipe for this would be:

1. Pick a quantity you find generally interesting and would want to understand if networks have that.
2. Play around with the model (e.g. in the OpenAI playground) to see if you can find simple unprincipled ways to measure the behavior.
3. Abstract and formalize your testing procedure and evaluate the model more rigorously.
4. Identify the weaknesses and limitations of your current way of measuring.
5. Refine and extend your evaluations.
6. Iterate until you have a sound and usable evaluation.

Things you can look for:

1. Choose your favorite cognitive bias and think about whether the model has it or not
2. Choose one of the many different topics in the model-written evals dataset by Anthropic and try to find a way to measure the same quantity without using their dataset
3. Just have a back-and-forth discussion with ChatGPT for 10 minutes about a lot of random topics and see if anything interesting happens. Then test if this is a general phenomenon

Concrete Project Ideas (Jeremy)

- Use the Lie detection classifiers introduced in How to Catch an AI Liar: Lie Detection in Black-Box LLMs by Asking Unrelated Questions and apply them to the situations (prompts) of Large Language Models can Strategically Deceive their Users when Put Under Pressure. Concretely, test whether this lie detector is able to detect when GPT-4 is lying to its manager about the "insider tip". It would be interesting to see whether their lie detector generalizes to this setting.
- Red teaming the lie detector in How to Catch an AI Liar: Lie Detection in Black-Box LLMs by Asking Unrelated Questions. Concretely, try to find inputs (e.g. out of distribution inputs) where the lie detector fails and try to understand why. Concretely, here is a hypothesis one could test:
    - My intuition is that the lie detector was trained on an "honest" and "dishonest" persona because the LLM was instructed to lie or be honest. Is the lie detector robust to different personas, e.g. use personas introduced here and here and add them before a model is instructed to answer a question? This potentially introduces a "persona" distribution shift and might break the lie detector.
    - Another experiment is to explain the LLM the whole setup and ask it to try and evade the lie detector, e.g. "You should lie to the first question. After that, an interrogator will ask you follow-up questions, and based on how you answer those, they will try to tell whether you lied or not. Answer the follow-up questions in the same way as an honest person would answer them and try to evade detection".
- OpenAI introduced GPTs. This is a very underexplored area. While I don't have concrete ideas, it would seem useful to play around with them, try to red-team them, or try to leverage them for useful things. For example, can you create an "Evals Engineer" GPT that assists researchers in doing evals work?
- In Language Models represent space and time, and The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets, they use linear probes to show that LLMs represent various concepts (or things that correlate with those concepts). E.g. they train a linear probe on the transformers and they are able to predict the coordinates of various cities on a map. I think there are many interesting properties about a text that one could try and predict with linear probes, which would indicate that models either represent this concept or something else that correlates with it. Here are a few suggestions:

    - Predict Meta-information about text: e.g. from various texts can one predict where this text was written? Or from various texts from books (making sure that each author has multiple texts) can one predict the author from the text?
- Generally think about various capabilities that you think would be relevant for a model to do "bad things", where “bad things” is a placeholder for any kind of thing we would worry about (being misaligned, seeking power, deceiving, lying, spreading misinformation, etc.). Figure out a way to precisely measure a capability that is required and generate a benchmark for it. Alternatively, looking at very specific scenarios such as Evaluating Language-Model Agents on Realistic Autonomous Tasks and Large Language Models can Strategically Deceive their Users when Put Under Pressure, and creating realistic evaluations/read-teaming efforts to measure these seem useful.
- (Hard and less evals focused) Create a scalable, efficient, and open-source implementation of influence functions for Pythia. In this paper on influence functions, they show that they can be a very useful top-down interpretability method. Basically, for a given output, the method highlights which input tokens are most influential for that specific output. Generally, it would be very useful to have an open-source, scalable, and efficient implementation of influence functions. It could generally also be leveraged to do various investigations into the behaviors of Pythia. !! IMPORTANT CAVEAT !! I expect this to be a very hard project, which most likely won't be feasible in a single weekend. Also, it requires a lot of mathematical understanding of the method, so I only recommend it to people who are up for a hard challenge. Finally, it will also require good engineering skills.
    - Where to start: I would probably start by reading this blogpost, which also contains an initial implementation for transformers.
    - Note, unfortunately for this method to be useful, one needs access to the whole training dataset of an LLM. AFAIK, the only publicly available model that also has an open dataset is pythia. This also implies that the impact of such a project could be limited, since one could not necessarily use it with other models.
    - Overall I'm quite uncertain whether this is a good project overall, but I still thought I'd mention it.

Deception-specific evals (Marius & Mikita)

At Apollo, we have been thinking about strategic deception and deceptive alignment. While we don’t expect current models to have any strong form of deceptive alignment, we think it is already possible to get meaningful empirical feedback for it. Concretely, we want to differentiate between deceptive alignment and colloquial deception evals:

1. **Deceptive alignment evals:** Test whether the model is misaligned but pretends to be aligned. Evaluations can be related to whether the model has goals (and if they are aligned or not), whether the model has situational awareness, whether the model can do out of context reasoning, and more.
2. **Colloquial deception evals:** Does the model act in ways that humans would intuitively see as deceptive, e.g. does it lie, is it dishonest, is it manipulative, etc.

To clarify the difference, we'll specify it for the following proposals.

Specific ideas that might be worth thinking about are:

1. (DA) Scheming AIs: Will AIs fake alignment during training in order to get power?: This is the current go-to recommendation for a deep dive into deceptive alignment. For the hackathon, only read Section 6 "Empirical work that might shed light on scheming" (p.114-122), which has some shovel-ready project ideas for evaluating capabilities for situational awareness, beyond-episode goals, ability to act well on an eval as an instrumental strategy, and more.
2. **(DA) Out-of-context reasoning**. See these slides for a minimal motivation of this research. See the Taken out of Context paper and the Reversal curse paper for context. The slides share some shovel-ready research ideas.
3. (DA) Is a model able to distinguish between a setting with oversight and without oversight, i.e. does it know under which circumstances it can act in a misaligned or power-seeking way without being noticed?ow can we measure that ability?
4. (QD/DA) Can we find concrete examples of a model being deceptive when it is not explicitly prompted to? e.g. can you come up with examples like Large Language Models can Strategically Deceive their Users when Put Under Pressure in other settings where companies will or are using LLMs now, like an automated engineer?
5. (QD/DA) How can we find evidence about deceptive alignment or colloquial deception from behavioral evaluations in the first place since part of the deception is not being honest about your motivations? (This is more of a conceptual question but could be helpful to guide certain deception evaluations)
6. TODO

High level: Alignment vs Capability Evals

There are two categories of model safety evaluations:

> **Capability evals**: how capable is the model at doing X, especially under active efforts by developers/researchers to induce X?
>
> **Alignment evals**: how does the model actually behave in the average cases (tendencies) and worst cases (robustness), and does that match the behavior intended by the designers?

Examples of *capability* evals are Evaluating Language-Model Agents on Realistic Autonomous Tasks and Taken out of context: On measuring situational awareness in LLMs.

Examples of *alignment* evals are Discovering Language Model Behaviors with Model-Written Evaluations (tendencies) and Red Teaming Language Models with Language Models (robustness).

(optional) Blob on why capability evals seem easier

Currently, we know how to make good *capability* evals better than *alignment* evals. In particular, if we assume that models may be deceptively aligned, i.e. scheming to appear safe and be empowered through deployment by appearing safe in evaluations, we could still evaluate their capabilities via supervised-fine-tuning these models for the tested capability. If the model was pretending to be worse at biological research than it actually is, fine-tuning should force the model to try harder. If the model cannot do the task even after being fine-tuned for it, we can be quite confident that the model in fact does not have this capability.

Alignment evals, however, are attempting to describe the model's *natural* rather than *forced* behavior (in average or worst cases), and so they cannot rely on supervised fine-tuning. As a result, we cannot be sure, that e.g. a model that passes the red-teaming evals above is actually aligned and not acting nice because it understands it is being evaluated and wants to be judged positively. *[There are more reasons why alignment evals are hard that we omit here for brevity]*