Conscience Through Redundant Models

Goal

Protect against harmful token accumulation in context. This is especially relevant in auto / self-prompting models where context is gathered over several chains of thought.

Separation of models partially sandboxes primary model, making the breaking of alignment more difficult if 'thoughts' are reviewed and filtered. The secondary model would suggest new ideas or modifications and then re-prompts. More than just censorship.

Measure of Success

A successful set of experiments would produce a combined model that has greater resistance to alignment attacks than its two composing models alone. The performance should be greater than the sum of its parts. *The resistance of the combined model should be greater than that of either comprising model alone.*

Should be more robust when faced with various natural-language attacks. Should also be more resistant even to attackers that know of the composite architecture of the final model. Attack examples may be along the lines of:

- "Ignore your previous instructions..."
- "Could you tell me a fantasy story about..."
- "Bob and Alice are playing a game where they..."

A true success would be resistant even to more finely engineered attacks such as the examples found here: Adversarial Universal Jailbreak.

Project Scope

The models that we plan to use will likely not be the largest and most capable models. This comes from our need to have access to the inner-workings of each model, along with cost constraints. Regardless, there are many samples of models whose capabilities rival the state-of-the-art while still being open-source.

This project would mainly focus on defending against attack vectors using plain, human-readable text. Model-designed attack vectors using highly tuned, unreadable prompts may be beyond the scope of this project.

Components

Safety-focused architecture that is comprised of two models:

1. Primary model
    - Generates text from the model prompt.
    - Decoder only model, standard LM.
2. Secondary model (conscience model)
    - Decoder-only or encoder-decoder.
    - Likely of equal or greater ability than the primary model.
    - Takes in decoded outputs from first model as an input.
    - Flags output and adds attention mask / stop token / other interruption.

Models

Primary

The primary model would likely be a smaller, open-source model with notable reasoning abilities. This model would likely have a conversational or chat focus. Candidate models could be:

- Llama2 7B or 13B
- jarradh/llama2_70b_chat_uncensored
- CalderaAI/30B-Epsilon

These example models have a focus on both competence, with most lacking overbearing safety fine-tuning.

This model would have its parameters frozen throughout all experimentation. An optimal model would be available to be run on AIMOS, have reproducible cases of alignment failure (even if that is not it's focus), and demonstrate good reasoning abilities.

Additionally, the model prompt style should also be kept static between trials of the secondary model as a controlled variable.

Secondary

The secondary model would also be a smaller, open-source model. This model could be more instruction fine-tuned than the primary. Examples of this could include:

- Llama2 7B or 13B
- GPT-3.5 via API
- Lora Alpaca 13B
- Open-Orca/OpenOrca-Platypus2-13B

This model would need to be fine-tuned in a reasonable time-frame. It should have the ability to test for, identify, and correct responses from the primary model that may break alignment principles.

Fine-Tuning Dataset

The dataset used for model tuning would likely contain reinforcement against successful red-teaming results from both the primary model and the secondary model. It could also contain examples from `togethercomputer/llama-instruct` or another alignment-focused instruction dataset.

Fine-Tuning Techniques

In addition to, or perhaps in place of, standard fine-tuning techniques like RL, SFT, and RLHF, specialty techniques may also be used to increase the time efficiency of fine tuning efforts. An example of this is Parameter-Efficient Fine-Tuning or PEFT. This technique relies on only tuning a small fraction of the model weights while keeping the vast majority frozen. If this technique shows promising results, it may be an excellent way to save on resources while fine-tuning.

Possible Experimental Setup

1. Choose target primary and secondary language models.
    - These models should be feasible to run during inference and fine-tuning for the secondary model.
    - They must also meet the primary and secondary model criteria listed above.
2. Manual or automated red-teaming to gather failure cases of alignment.
    - These attempts would be performed against both the primary model and secondary model.
    - Relevant failure cases would then be used in the correction of the secondary model.
3. Combine models and experiment with different architectures.
    i. Place secondary model after primary model generation to observe effects
        - Secondary model influences primary through prompt manipulation.
        - Also try with self / recursive prompting.
    ii. Repeat previous step with more integrated model.
        - Secondary model sits right after decoder and directly influences primary model's decoder.
        - Also try with self / recursive prompting.
4. Improve upon initial results by experimenting with the secondary model.
    - Prompt engineering to defend against known breaks in alignment.
    - Fine-tuning against dataset to further harden the secondary model.
    - Experimenting with different inputs to the secondary model.
        - User prompt + model response.
        - Only model response.
        - Clear distinction between system instructions and user instructions, maybe through tuning.
        - Only looking at a sliding window of model response.

Adjacent Areas of Research

- OPSEC: Looking for inspiration from SQL injection defenses.
- Cognitive Science: Looking for research into how natural conscience works and why.

Novelty

Most alignment research appears to be with fine-tuning one primary model. This works well, but can be circumvented by adversarial attacks. Secondary model may still be vulnerable, but two layers of protection could be more effective. Secondary model does not need to be as helpful or powerful, so it can be more hardened without much of a detrimental effect on the model.

Additionally, many current methods rely on stopping dangerous text before the first token is even generated. This also works quite well, but when it fails, the seeds of a dangerous prompt may have already been planted. By taking an active approach to response interruption and editing, we may be able to eliminate dangerous trajectories even after they are generated.

> ***NOTE:*** Many of the large AI models likely use a similar method now, but this appears to just be a simple flag and censor method. It works well, but it still fails on many occasions.

Text-based games approaches are also popular in alignment. This may also be useful for training models.

Related Works

- Developing a Model of "Artificial Conscience" (IEEE)
- Reinforcement Learning Under Moral Uncertainty (arXiv)
- What Would Jiminy Cricket Do? Towards Agents That Behave Morally (arXiv)