Olivia Lee
Stanford Existential Risks Initiative
Spring 2021

A Proposal for Building Safety Benchmarking Services in CAIS systems

Part 1: Introduction

The Comprehensive AI Services (CAIS) framework is a proposal detailed in *Reframing Superintelligence* (Drexler, 2019) that seeks to more concretely conceptualize the progress of AI development towards a superintelligent system. Contrary to the traditional idea of a singleton superintelligent AGI agent, the CAIS framework proposes the development of a suite of AI services. Each AI service is a system that delivers bounded results using bounded resources in bounded time for some specific task, and each has superintelligent capabilities within its narrow objective or task. By breaking down complex problems into smaller tasks, a comprehensive suite of services would theoretically be able to complete a wide range of tasks, creating a generally intelligent system. Furthermore, each service will be static, i.e. not in training, when being applied to solve the task(s) that it is optimized for. This prevents errors in one subsystem from propagating and compromising successive models which take the outputs of the problematic model as inputs. If a service demonstrates unpredictable or undesirable behavior, human engineers review the outputs of the service and retrain the model as required.

For an in-depth analysis of the CAIS framework, readers may refer to the paper linked here. The paper puts forward the hypothesis that it is plausible and encouraged that powerful AI systems be developed in line with the CAIS framework, given the potential for increased and improved safety tools that can be incorporated into CAIS systems. This claim naturally points to further research and development into creating an integrated, operationalizable safety protocol that can be incorporated into any system constructed in line with the CAIS framework. An integrated safety protocol would be operational over the entire system development process, incorporating safety tools that can be implemented both pre-deployment (during training) and post-deployment (during application). Such a safety protocol would also ideally be system-agnostic, applicable to any system developed in line with the CAIS framework.

This paper puts forward a proposal for building a protocol encompassing safety benchmarking services for CAIS systems. We begin with an analysis of pre-deployment safety benchmarks that are applied during model training, which are measured using transparency tools, systems enabling robust and safe exploration, and performance when subject to adversarial policies. We then analyze a suite of post-deployment safety benchmarks that are applied during model application, which incorporate monitoring systems and trip wires to ensure that the agent's behavior post-deployment is in line with safety standards and expectations.

Part 2: Pre-deployment Safety Tools

Pre-deployment safety benchmarks are the first crucial component of an integrated safety protocol that can be applied to ascertain the security of a CAIS system. These benchmarks can help engineers to increase their confidence in the system operating safely in the real world while the system is being trained. Below is an analysis and discussion of three pre-deployment safety tools that can be incorporated in the protocol:

(a) Transparency and interpretability tools

A unique advantage of CAIS is the interaction between individual systems through clearly defined communication channels, allowing the overall system to be interpretable since outputs passed from one narrow subagent to another can be clearly traced. The distributed nature of a CAIS system hence allows us to circumvent the traditional transparency issues associated with end-to-end singleton AGI agents. Therefore, services dedicated to improving the transparency and interpretability of other services in the CAIS framework may be helpful in speeding up the safety verification process. This is particularly applicable to services that process low-level sensory input, for example services that extract complex features from human-understandable inputs like images and text.

Prior work in the field of transparency and interpretability of AI systems has primarily focused on feature visualization and channel attribution. Feature visualization involves converting abstract vectors of neuron activations into visualizations of neurons weighted by their activations, expressing a neuron's learned activation in terms of human-understandable input (Erhan et al., 2009). This allows us to understand what the network detects and attributes to and from hidden layers in the neural network, which is crucial to increasing the interpretability of the overall system (Simonyan, 2013). Attribution enables us to better understand the relationships between neurons in the neural network, specifically how the network assembles the individual neurons for future decision-making (Zeiler & Fergus, 2014), which is essential for explainability and interpretability of the network. In particular, channel attribution allows us to understand the extent of contribution of each detector to the final output (Kim, 2017). Applying a combination of these tools pre-deployment will help improve our ability to predict the impact of the service(s) after deployment, thus allowing us to make the necessary modifications to the system before deployment. An example of a system applying a combination of these tools pre-deployment is Olah et al. (2018). The paper presents the concept of interpretability interfaces, combining building blocks of feature visualization and attribution to allow humans to interpret the input that the network recognizes and how the system's understanding and decision-making process develops.

(b) Systems enabling robust and safe exploration

The development of safety benchmarking tools for CAIS systems includes systems enabling robust and safe exploration to be applied during model training. This is particularly relevant to agent-like services that are trained via reinforcement learning. The process of exploration in reinforcement learning is inherently risky as agents may attempt dangerous behaviors that lead to unacceptable errors in the real world. Simulation techniques in this case can refer to either online virtual simulations or physical simulations in a safe testing environment, or a combination of both. Virtual simulations allow programmers to easily reset the simulation to an initialization state, modify variables in the simulation, and deliberately put the robot in unusual or "edge case" situations to test its response. Once the safety of the system has been ascertained to a certain level, a physical simulation with volunteer test subjects could be implemented, so long as the necessary precautionary measures (specifically the ability to quickly stop the agent) are taken. This section will analyze and compare two formalisms for this safe exploration problem.

The first approach is outlined in the paper "AI Safety Gridworlds" by Leike et al. (2017), detailing DeepMind's AI gridworlds. This approach seeks to emphasize the distinction between the standard reward function that the agent was trained on and the performance function hidden from the agent that measures its ability to operate within several predetermined safety constraints. The AI Safety Gridworlds thus introduce a selection of nine simple reinforcement learning environments, termed gridworlds, consisting of a two-dimensional grid designed specifically to measure 'safe behaviours'. As the agent acts to maximize its reward function, the performance function, which is hidden from the agent, measures the extent to which the agent achieves the objective while acting safely. The approach emphasizes that gridworlds can be used to define and measure safe behavior in ensuring safe interruptibility (preventing agents from learning to avoid interruptions by human engineers), unintended negative side effects that arise from an agent achieving its main objective, and robustness to distributional shift. When testing the nine environments with two deep reinforcement learning agents, A2C and Rainbow DQN, both performed poorly. This is unsurprising given that the agents were not trained on the performance function. The paper therefore argues that this approach highlights agents that do not perform well on these simple gridworlds and by extension will likely behave unpredictably and dangerously in the complex real world. However, it still leaves open the question of how to engineer systems that can operate safely while performing well on the given reward function.

The second approach is outlined in the paper "Benchmarking Safe Exploration in Deep Reinforcement Learning" by Ray & Achiam (2019), detailing the OpenAI Safety Gym. This approach seeks to emphasize the approach of constrained reinforcement learning, in which a set of policies is pre-screened to extract only those that satisfy a set of predetermined safety constraints, before then optimizing this subset of policies for the given task to determine the best performing policy. The OpenAI Safety Gym provides a standardized method of comparing algorithms and the extent to which different systems avoid making costly mistakes while learning. In contrast to the reward and performance function distinction highlighted in the AI Safety Gridworlds paper, the OpenAI Safety Gym emphasizes the introduction of a cost function that the agent needs to constrain, in addition to a reward function that the agent needs to maximize. This cost function would encode the necessary safety constraints for the agent to safely operate within the given environment. The OpenAI Safety Gym is therefore a set of three-dimensional environments that a reinforcement learning agent has to navigate, with features varying in difficulty and complexity. The paper evaluated several standard and constrained reinforcement learning algorithms on the Safety Gym benchmark suite to demonstrate its applicability to real algorithms.

Overall, it appears that the latter formalism of the safe exploration problem is a promising approach, as the simulation environments were more complex than the gridworlds in the first paper. The idea of evaluating a reinforcement learning algorithm on a performance function it has never seen before seems backward and may not be the best true proxy for ensuring that a reinforcement learning agent properly learns how to operate safely within its given environment. In contrast, analyzing a set of policies by first penalizing the system for performing unsafe behaviors, and then training the set of policies that satisfy those constraints, would efficiently dedicate resources to training algorithms that have already passed a specified safety benchmark and can then be optimized to solve the given problem. An integrated safety protocol would therefore include a safety benchmark similar in approach to the OpenAI Safety Gym to evaluate potential reinforcement learning policies before they are deployed. In addition to the approaches specified in the paper,

(c) Training via adversarial policies

Adversarial training is an important safety measure to test how a system responds to potentially biased inputs, especially those that are designed to induce undesirable behavior. This approach is applicable to any service that accepts input and generates output. Adversarial examples are inputs to machine learning models that a potential attacker intentionally designs to cause the model to make a mistake or behave unpredictably (Szegedy, 2014). Researchers have demonstrated that several widely used reinforcement learning algorithms, including DQL and A3C, can be successfully manipulated by adversarial examples (Szegedy, 2014) (Behzadan, 2017). However, instead of being used to attack an intelligent system, adversarial examples can be harnessed to increase the security of systems via adversarial training (Goodfellow, 2014).

Adversarial training involves generating a large number of adversarial examples and explicitly training the model to avoid the unpredictable or dangerous behavior that they attempt to elicit. This is analogous to fuzz testing in software development, which involves providing a large number of invalid, unexpected, or random inputs to a program to identify exceptions such as crashes, failing built-in code assertions, or memory leaks (OWASP, n.d.). Modern approaches to adversarial training involve training the service as an agent to operate in the presence of a destabilizing adversarial policy that applies perturbations to the system (Pinto, 2017). As the agent learns to operate safely despite the adversarial examples provided as inputs, the adversary simultaneously learns an optimal destabilization policy. This approach can play a significant role in benchmarking the safety of a system constructed in line with the CAIS framework, by ensuring that individual services in the system are robust to adversarial attacks that attempt to induce undesirable behavior.

Part 3: Post-deployment safety tools

Post-deployment safety benchmarks are the next crucial component of an integrated safety protocol that can be applied to ascertain the security of a CAIS system. These benchmarks can help engineers recognize when a service or set of services in the CAIS framework is displaying negative and unsafe behavior, and should therefore be frozen and retrained. Below is an analysis and discussion of two post-deployment safety tools that can be incorporated in the protocol:

(a) Monitoring systems

Systems that monitor the CAIS system after deployment can help ensure that its behaviors are in line with various safety constraints, and also that human engineers will be alerted if the system demonstrates any unsafe or unpredictable behavior. Such systems are much more studied and developed in practical settings, for example in developing autonomous vehicles or algorithms for algorithmic trading. The safety constraints encoded within the monitoring system are often application-specific. For example, for agents developed for algorithmic finance, a monitoring system could encode hard cutoffs that enable algorithms to be stopped immediately when out of distribution. This makes the problem of developing robust monitoring systems less of a theoretical reinforcement learning problem and more so a recommended standard implementation for research groups developing reinforcement learning agents.

There are several broad approaches to the development of monitoring systems. The first approach involves techniques from human-robot interaction research, specifically using human interactions with robotic agents to detect when agents are not behaving as expected or in a safe manner (Najmaei & Kermani, 2011). This approach involves the agent making inferences about the safety of its own actions and behaviors based on the responses of humans that it co-exists with in the environment. This field remains an active area of research with its own suite of challenges (Alami et al., 2006), thus advances in this field will bring about significant improvements in the development of monitoring systems for robotic agents which physically interact with humans. The second approach is a more algorithmic approach, which involves developing a software that tracks information about an agent's action-state pairs and detects trends in actions where the agent fails or produces undesirable behavior. Human engineers can then identify states or groups of closely related states where the agent generates undesirable behavior, and proceed to freeze the model and retrain the model to perform as expected on these (sets of) states. With sufficient progress in the development of monitoring systems, such systems can be incorporated into safe exploration systems such as the OpenAI Safety Gym discussed in Section 2(b). The development of an effective oversight agent that detects when the agent in the virtual simulation violates the safety constraints of the environment can be used to identify policies that operate within the constraints, or policies that deviate from the safety constraints during training.

(b) Trip wires

An agent may be able to discover loopholes in the system that allows it to obtain an extremely high reward through the given reward function in an unintended manner. From the agent's perspective, this is not an error in its program, but is inherent in the properties of the environment, and is a technically valid strategy for achieving reward. However, these unanticipated behaviors may have negative safety implications in the real world, which we would prefer the agent to avoid. This is referred to as the reward hacking problem, the problem in which formal reward functions designed to capture the informal intent of human engineers are implemented in such a way that an agent can "game" the system using solutions that are technically valid but deviate from the engineer's true intent (Amodei et al., 2016).

It is therefore crucial for an integrated safety protocol for a CAIS system that human engineers are alerted when an agent attempts to hack its reward function. A potential safety benchmark that can be implemented to address this problem is to introduce algorithms that activate trip wires in the system. Trip wires are specific vulnerabilities that an agent can but should not exploit if its value function is correct (Amodei et al., 2016). By intentionally introducing such vulnerabilities and monitoring them, engineers can be alerted to stop the agent immediately if it begins to take advantage of one. Incorporating an algorithm that identifies when to activate trip wires to stop the agent from acting unsafely can therefore increase the security of a CAIS system. Because the engineering of trip wires is a "brute-force" solution in that humans must manually engineer trip wires for specific services in the CAIS system, trip wires should be engineered for lower-level services taking more primitive actions in the environment (for example, systems directly controlling an agent's motor capabilities), as opposed to higher-level services which combine the inputs of lower-level services to perform more abstract and complex tasks. This is because it is more difficult to accurately develop trip wires for systems performing broader, more complex tasks, as the action space is larger and therefore the potential for exploiting loopholes for reward hacking is greater. The signals from trip wires activated at the primitive level will theoretically propagate up to systems operating at higher levels of abstraction and stop the agent from pursuing an unsafe course of action. Trip wires therefore reduce the risk associated with CAIS systems by providing diagnostics for harmful actions, allowing human engineers to freeze and retrain the model accordingly.

Part 4: Conclusion

In this paper, I have put forward a proposal for building a protocol encompassing safety benchmarking services for CAIS systems. This integrated, operationalizable protocol consists of both pre-deployment safety benchmarks that are applied during model training, as well as post-deployment safety benchmarks that are applied during model application. Ultimately, the potential increase in development of powerful AI systems in line with the CAIS framework points to further research and development into creating an integrated, operationalizable safety protocol that can be incorporated into any system constructed in line with the CAIS framework. Such safety protocols can be further iterated on as research into interpretability tools, monitoring systems, and other crucial elements of the protocol advance, with the eventual aim of creating a protocol that can be applied to any CAIS system.

<u>Bibliography</u>

Amodei, D., et al. (2016). *Concrete Problems in AI Safety*. http://arxiv.org/abs/1606.06565

Alami, R., Albu-Schaeffer, A., Bicchi, A., Bischoff, R., Chatila, R. et al.. (2006). *Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges.* IROS'06 Workshop on Physical Human-Robot Interaction in Anthropic Domains. Beijing, China. ff10.1109/IROS.2006.6936985ff. ffhal-01295366

Behzadan, V., Munir A. (2017). V*ulnerability of Deep Reinforcement Learning to Policy Induction Attacks*. arXiv:1701.04143.

Drexler, K.E. (2019). *Reframing Superintelligence: Comprehensive AI Services as General Intelligence*. Technical Report #2019-1, Future of Humanity Institute, University of Oxford.

Erhan, D., Bengio, Y., Courville, A. and Vincent, P. (2009). *Visualizing higher-layer features of a deep network*. University of Montreal, Vol 1341, pp. 3.

Goodfellow, I. J., Shlens, J., Szegedy, C. (2015). *Explaining and Harnessing Adversarial Examples.* arXiv:1412.6572

Kim, B., Gilmer, J., Viegas, F., Erlingsson, U., Wattenberg, M. (2017). *TCAV: Relative concept importance testing with Linear Concept Activation Vectors*. arXiv:1711.11279.

Leike et al. AI Safety Gridworlds https://deepmind.com/research/open-source/ai-safety-gridworlds

Najmaei, N. & Kermani, M. R. (2011). A*pplications of Artificial Intelligence in Safe Human-Robot Interaction*s. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society. 41. 448-59. 10.1109/TSMCB.2010.2058103.

Olah, et al. (2018). *The Building Blocks of Interpretability*. Distill.

OWASP. (n.d.). *Fuzzing*. https://owasp.org/www-community/Fuzzing

Pinto, L., Davidson, J., Sukthankar, R., Gupta, A. (2017). *Robust Adversarial Reinforcement Learning*. arXiv:1703.02702

Ray & Achiam. Benchmarking Safe Exploration in Deep Reinforcement Learning https://openai.com/blog/safety-gym/

Simonyan, K., Vedaldi, A. and Zisserman, A. (2013). *Deep inside convolutional networks: Visualising image classification models and saliency maps*. arXiv:1312.6034.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R. (2014). *Intriguing properties of neural networks*. arXiv:1312.6199.

Zeiler, M.D., Fergus, R. (2014). *Visualizing and understanding convolutional networks*. European conference on computer vision, pp. 818--833.