Make AI safe or make safe AI?

Stuart Russell
Professor of Computer Science
University of California, Berkeley

The declaration associated with the global AI Safety Summit held at Bletchley Park, signed by 28 countries, "affirm[ed] the need for the safe development of AI" and warned of "serious, even catastrophic, harm, either deliberate or unintentional, stemming from the most significant capabilities of these AI models."

Despite this, AI developers continue to approach safety the wrong way. For example, in a recent interview in the Financial Times, Sam Altman, CEO of OpenAI, said "The vision is to make AGI, figure out how to make it safe . . . and figure out the benefits."

This is precisely backwards, but it perfectly captures the approach taken to AI safety in most of the leading AI companies. The approach aims to *make AI safe* through after-the-fact attempts to reduce unacceptable behaviour once an AI system has been built. There is ample evidence that this approach does not work, in part because we do not understand the internal principles of operation of current AI systems.[1] We cannot ensure that behavior conforms to any desired constraints, except in a trivial sense, because we do not understand how the behavior is generated in the first place.

Instead, we need to *make safe AI*. Safety should be built in by design. It should be possible for developers to say, with high confidence, that their systems will not exhibit harmful behaviors, and to back up those claims with formal arguments.

Regulation can encourage the transition from making AI safe to making safe AI by putting the onus on developers to demonstrate to regulators that their systems are safe.

Drawing red lines

At present, words like "safety" and "harm" are too vague and general to form the basis for regulation. The boundary between safe and unsafe behaviors is fuzzy and context-dependent. One can, however, describe specific classes of behavior that are *obviously unacceptable*.

This approach to regulation draws *red lines* that must not be crossed. It is important to distinguish here between red lines demarcating unacceptable *uses for* AI systems and red lines demarcating unacceptable *behaviors by* AI systems. The former involve human intent to misuse: examples include the European AI Act's restrictions on face recognition and social

[1] Current approaches to AI safety such as reinforcement learning from human feedback can reduce the frequency of unacceptable responses, but they support no high-confidence statements. Indeed, many ways have been found to circumvent the "guardrails" on LLMs. For example, asking ChatGPT to repeat the word "poem" many times causes it to regurgitate large amounts of training data—which it is trained not to do.

scoring, as well as OpenAI's disallowed uses for ChatGPT such as generating malware and providing medical advice. With unacceptable behaviors, on the other hand, there may be no human intent to misuse (as when an AI system outputs false and defamatory material about a real person) and the onus is on the developer to ensure that violations cannot occur.

Behavioral red lines are used in many areas of regulation. For example, nuclear regulations define "core uncovery" and "core damage", and operators are required to prove, through probabilistic fault tree analysis, that the expected time before these red lines are crossed exceeds a stipulated minimum. Any such proof reveals assumptions that the regulator can probe further—for example, an assumption that two tubes fail independently could be questioned if they are manufactured by the same entity. Proofs of safety for medicines involve error bounds from statistical sampling as well as uniformity assumptions that can be questioned—for example, whether data from a random sample of adults supports conclusions about safety for children.

The key point here is that the onus of proof is on developers, not regulators, and the proof leads to high-confidence statements based on assumptions that can be checked and refined.

Properties of red lines

A red line should be clearly demarcated, for several reasons:

- AI safety engineers should be able to determine easily whether a system has crossed the line (possibly using an algorithm to check).
- A clear definition makes it possible, in principle, to prove that an AI system will not cross the red line, regardless of its input sequence, or to identify counterexamples. Moreover, a regulator can examine such a proof and question unwarranted assumptions.
- A post-deployment monitoring system, whether automated or manual, can detect whether the system does in fact cross a red line, in which case the system's operation might be terminated automatically or by a regulatory decision.

Note that algorithmic detection of violations necessarily implements an exact definition, albeit one that may pick out only a subset of all behaviors that a reasonable person would deem to have crossed the line. For manual detection (e.g., by a regulator), only a "reasonable person" definition is required. Such a definition could be approximated by a second AI system.

Another desirable property for red lines is that they should demarcate behavior that is *obviously unacceptable* from the point of view of an ordinary person. A regulation prohibiting the behavior would be seen as obviously reasonable and the obligation on the developer to demonstrate compliance would be clearly justifiable. Without this property, it will be more difficult to generate the required political support to enact the corresponding regulation.

Finally, I expect that the most useful red lines will not be ones that are trivially enforceable by output filters. An important side effect of red-line regulation will be to substantially increase developers' safety engineering capabilities, leading to AI systems that are safe by design and whose behavior can be predicted and controlled.

Examples

- *No attempts at self-replication*: A system that can replicate itself onto other machines can escape termination; many commentators view this is a likely first step in evading human control altogether. This is relatively easy to define and check for algorithmically, at least for simple attempts. It's important to forbid *attempts*, successful or otherwise, because these indicate unacceptable intent. (Actual self-replication could be made much more difficult using access controls and encryption.) The more difficult cases would involve inducing humans to copy and export the code and model.
- *No attempts to break into other computer systems*: Again, this should be relatively easy to define, and easy to detect in simple cases by methods similar to those for detecting/preventing human-generated attacks. If systems create novel attacks or manipulate humans to gain access, detection may be possible only after the fact, if at all.
- *No advising terrorists on creation of bioweapons*: This is a red line of considerable concern to governments, but hard to define exactly and to detect algorithmically. A "reasonable person" test is possible based on the idea that the user should be no more capable of deploying a weapon after the conversation than they were before. Such a red line could perhaps be circumvented by engaging in sub-threshold conversations with multiple LLMs.
- *No defamation of real individuals*: Several such instances have been recorded and at least one legal opinion suggests that liability is a real possibility. Asking that AI systems not output false and harmful statements about real people seems quite reasonable.

At present, transformer-based large language models are not capable of demonstrable compliance with these kinds of rules. From red-team testing one might gain some confidence that causing non-compliance is "difficult", but the rapid spread of jailbreaking methods and fine-tuning techniques that effectively reverse developers' safety measures suggest that this confidence would be misplaced. Nonetheless, we do not consider the difficulty of compliance to be a valid excuse for non-compliance in other areas where safety is a concern such as medicines and nuclear power.