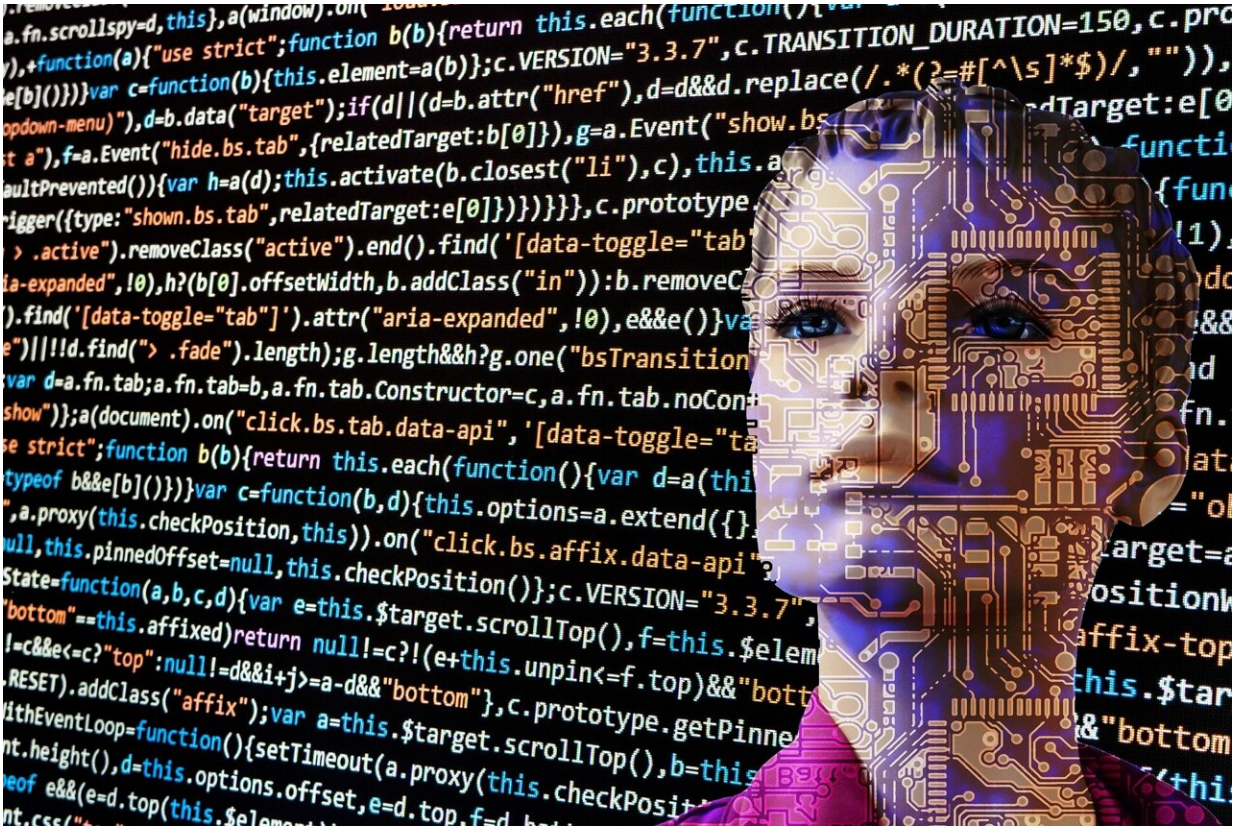
Researchers develop AI-driven machine-checking method for verifying software code

January 4 2024



Credit: Pixabay/CC0 Public Domain

A team of computer scientists led by the University of Massachusetts Amherst recently announced a new method for automatically generating whole proofs that can be used to prevent software bugs and verify that

the underlying code is correct.

This new method, called Baldur, leverages the artificial intelligence power of large language models (LLMs), and when combined with the state-of-the-art tool Thor, yields unprecedented efficacy of nearly 66%. The team was recently awarded a Distinguished Paper award at the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering.

"We have unfortunately come to expect that our software is buggy, despite the fact that it is everywhere and we all use it every day," says Yuriy Brun, professor in the Manning College of Information and Computer Sciences at UMass Amherst and the paper's senior author.

The effects of buggy software can range anywhere from the annoying—glitchy formatting or sudden crashes—to potentially catastrophic when it comes to security breaches or the precision software used for space exploration or for controlling health care devices.

Of course, there have been methods for checking software for as long as it has existed. One popular method is the simplest: you have a human being go through the code, line by line, manually verifying that there are no errors. Or you can run the code and check it against what you expect it to do. If, for instance, you expect your word-processing software to break the line every time you press the "return" key, but it instead outputs a question mark, then you know something in the code is wrong.

The problem with both methods is that they are prone to human error, and checking against every possible glitch is extraordinarily time-consuming, expensive and infeasible for anything but trivial systems.

A much more thorough, but harder, method is to generate a mathematical proof showing that the code does what it is expected to do,

and then use a theorem prover to make sure that the proof is also correct. This method is called machine-checking.

But manually writing these proofs is incredibly time-consuming and requires extensive expertise. "These proofs can be many times longer than the software code itself," says Emily First, the paper's lead author who completed this research as part of her doctoral dissertation at UMass Amherst.

With the rise of LLMs, of which ChatGPT is the most famous example, a possible solution is to try to generate such proofs automatically. However, "one of the biggest challenges with LLMs is that they're not always correct," says Brun. "Instead of crashing and letting you know that something is wrong, they tend to 'fail silently,' producing an incorrect answer but presenting it as if it's correct. And, often, the worst thing you can do is to fail silently."

This is where Baldur comes in.

First, whose team performed its work at Google, used Minerva, an LLM trained on a large corpus of natural-language text, and then fine-tuned it on 118GB of mathematical scientific papers and webpages containing mathematical expressions.

Next, she further fine-tuned the LLM on a language, called Isabelle/HOL, in which the mathematical proofs are written. Baldur then generated an entire proof and worked in tandem with the theorem prover to check its work. When the theorem prover caught an error, it fed the proof, as well as information about the error, back into the LLM, so that it can learn from its mistake and generate a new and hopefully error-free proof.

This process yields a remarkable increase in accuracy. The state-of-the-

art tool for automatically generating proofs is called Thor, which can generate proofs 57% of the time. When Baldur (Thor's brother, according to Norse mythology) is paired with Thor, the two can generate proofs 65.7% of the time.

Though there is still a large degree of error, Baldur is by far the most effective and efficient way yet devised to verify software correctness, and as the capabilities of AI are increasingly extended and refined, so should Baldur's effectiveness grow.

The paper is published as part of the *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*.

More information: Emily First et al, Baldur: Whole-Proof Generation and Repair with Large Language Models, *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (2023). DOI: 10.1145/3611643.3616243