An Excerpt from Chapter Two of *Non-Computable You*

https://mindmatters.ai/podcast/ep211

Austin Egbert:

Greetings and welcome to Mind Matters News. Non-Computable You by our host, Robert J. Marks, is now available in audiobook form on Audible at Amazon.com. This week we have an excerpt from the start of chapter two as read by Larry Nobles. If you'd like to hear some from chapter one, check out last week's episode. Enjoy.

Larry Nobles:

Chapter Two: Can AI Be Creative?

Larry Nobles:

"Computers are useless. They can only give you answers." Pablo Picasso

Larry Nobles:

Some have claimed AI is creative, but creativity is a fuzzy term. To talk fruitfully about creativity, the term must be defined so that everyone is talking about the same thing and no one is bending the meaning to fit his purpose. In this and subsequent chapters, we'll explore what creativity is. And in the end, it'll become clear that properly defined, AI is no more creative than a pencil.

Larry Nobles:

Creativity, Originating Something New

Larry Nobles:

Lady Ada Lovelace, 1815 to 1852, daughter of poet George Gordon, Lord Byron, was the first computer programmer writing algorithms for a machine that was planned but never built. Footnote: Lovelace is often credited with writing an algorithm for Charles Babbage's Analytical Engine, a machine that was planned but never built. There is some controversy as to whether Lovelace or Babbage wrote this first program. In any case, Lovelace undoubtedly was involved to an extensive degree in the very earliest computer programs, and she was also the first to say that a computer could be programmed to do more than merely compute. End of footnote.

Larry Nobles:

Lovelace also was quite possibly the first to note that computers will not be creative, that is they cannot create something new. She wrote in 1842 that "The computer has no pretensions whatever to originate anything. It can do only whatever we know how to order it to perform." Alan Turing disagreed.

Larry Nobles:

Turing is often called the father of computer science, having established the idea for modern computers in the 1930s. Turing argued that "We can't even be sure that humans create because humans do nothing new under the sun, but they do surprise us." Likewise, he said, "Machines take me by surprise with great frequency, so perhaps, he argued, "it is the element of surprise that's relevant, not the ability to originate something new." Machines can surprise us if they're programmed by humans to surprise us, or if the programmers made a mistake and thus experienced an unexpected outcome.

Larry Nobles:

Footnote: Turing concedes that when machines surprise him, it tends to be because of traceable human error in his calculations. He also anticipates the objection that machine surprises are due to some creative mental act on my part and reflect no credit on the machine. But he doesn't answer this objection except to say it leads back to the question of consciousness, which "we must consider closed." End of footnote.

Larry Nobles:

Often though, surprise occurs as a result of successful implementation of a computer search that explores a myriad of solutions for a problem. The solution chosen by the computer can be unexpected. The computer code that searches among different solutions, though, is not creative. The creativity credit belongs to the computer programmer who chose the set of solutions to be explored. Shortly, we'll give examples from computer searches for making the best move in the game of GO, that's G-O, and for simulated swarms. Both results are surprising and unexpected, but there's no creativity contributed from computer code.

Larry Nobles:

The Flawed Turing Test

Larry Nobles:

Alan Turing, an atheist, wanted to show that we are machines and that computers could be creative. Turing equated intelligence with problem-solving, didn't consider questions of consciousness and emotion, and referred to people as human computers. Turing's version of The Imitation Game was proposed to show that computers could duplicate the conversational human. This is why the biographical movie starring Benedict Cumberbatch as Turing was titled The Imitation Game.

Larry Nobles:

How can computers imitate humans according to Turing? The Imitation Game, which came to be called the Turing test, simply asked whether in a conversational exchange using text, that is an exchange in which the participants are hidden from each other, a sufficiently sophisticated computer can be distinguished from a human. If a questioner gets lucid, human-sounding answers from the computer and believes the computer is in fact a human typing in answers from another room, then the test has been passed. Incidentally, the converse of a Turing test is easy. Simply ask it to calculate the cube root of 12 out to 10 significant figures. If the answer is almost immediate, you're talking to a computer.

Larry Nobles:

There are those who claim that today's chatbots have passed the Turing test. Chatbots are computer programs that respond in a human-like way in text-based exchanges. You might have interacted with a chatbot that popped up on your screen to offer help or answer questions. Chatbots are useful to business because they can function as digital help desks, but they are limited in the questions they can answer and can only respond as they are programmed to respond.

Larry Nobles:

Most famously, a chatbot known as Eugene Goostman reportedly passed the Turing test. One posted subtitle claiming this reads, "Eugene Goostman fools 33% of interrogators into thinking it is human in what is seen as a milestone in artificial intelligence." But those making a test often cheat to tweak the outcome. That's what Goostman's programmers did. Goostman supposedly was a Ukrainian, which meant English was not his first language, so his poor grammar could be excused. He was supposedly a teenager, so any lack of depth of understanding could be chalked up to his naive intellect. Likewise, his tendency to be silly and deflect questions could be considered typical teenage behavior. In short, Goostman's software character was gamed to pass the Turing test.

Larry Nobles:

Here are a couple of examples of an exchange between Goostman and his questioner. Goostman is first asked a question proposed in the first line of the lyrics of the song Close to You popularized by The Carpenters. Questioner: Why do birds suddenly appear? Goostman: Because two plus two is five. By the way, what's your occupation? I mean, could you tell me about your work? We see in this an often used cheat in chatbots. If an answer is not known, the chatbot responds with an unrelated question to derail the conversation's direction.

Larry Nobles:

Here's another exchange with Eugene Goostman. Questioner: It is okay. I get sick of sick people. How is your stomach feeling today? Is it upset maybe? Goostman: I think you can't smile at all. I bet you work in a funeral agency. As you can see, Goostman's answers here are elusively non-responsive.

Larry Nobles:

Selmer Bringsjord correctly notes the Turing test is gamed by programmers. Gamed here is a nice word for being an elusive cheat. As Bringsjord writes, "Though progress toward Turing's dream is being made, it's coming only on the strength of clever but shallow trickery." When gaming the system, chatbots can deflect detection by answering questions with other questions, giving evasive answers, or admitting ignorance. They display general intellectual shallowness as regards, creativity, and depth of understanding. Goostman answered questions with questions like, by the way, what's your occupation? He also tried to change topics with conversational whiplash responses, like I bet you work at a funeral agency. These are examples of the clever but shallow trickery Bringsjord criticized.

Larry Nobles:

What then do Turing tests prove? Only that clever programmers can trick gullible or uninitiated people into believing they're interacting with a human. Mistaking something for human doesn't make it human. Programming to shallowly mimic thought is not the same thing as thinking. Rambling randomness, such as the change of topic questions Goostman spit out, does not display creativity. "I propose to consider the question, can machines think?" Turing said. Ironically, Turing not only failed in his attempt to show that machines can be conversationally creative, but also developed computer science that shows humans are non-computable.

Larry Nobles:

The Lovelace Test for Creativity

Larry Nobles:

Bringsjord and his colleagues have proposed the Lovelace test as a substitute for the flawed Turing test. The test is named after Lady Lovelace. Bringsjord defined software creativity as passing the Lovelace test if the program does something that cannot be explained by the programmer or an expert in computer code. Computer programs can generate unexpected and surprising results. Results from computer programs are often unanticipated. But the question is does the computer create a result that the programmer looking back cannot explain?

Larry Nobles:

When it comes to assessing creativity and therefore consciousness and humanness, the Lovelace test is a much better test than the Turing test. If AI truly produces something surprising which cannot be explained by the programmers, then the Lovelace test will have been passed and we might in fact be looking at creativity. So far, however, no AI has passed the Lovelace test. There have been many cases where a machine looked as if it were creative, but on closer inspection, the appearance of creative content fades. Here are a couple of examples.

Larry Nobles:

AlphaGo. A computer program named AlphaGo was taught to play Go, the most difficult of all popular board games. AlphaGo was an impressively monumental contribution to machine intelligence. AI already had mastered tic-tac-toe, then the more complicated game of checkers, and then the still more complicated game of chess. Conquest of Go remained an unmet goal of AI until it was finally achieved by AlphaGo.

Larry Nobles:

In a match against human World Champion Lisa Dole in 2016, AlphaGo made a surprising move. Those who understood the game described the move as ingenious and unlike anything a human would ever do. Were we seeing the human attribute of creativity in AlphaGo beyond the intent of the programmers? Does this act pass the Lovelace test? The programmers of AlphaGo claim they didn't anticipate the unconventional move. This is probably true. But AlphaGo is trained to play Go by the programmers.

Larry Nobles:

Go is a board game with fixed rules in a static, never-changing arena, and that's what the AI did and did well. It applied programmed rules within a narrow rule-bound game. AlphaGo was trained to play Go, and that's what it did. So no, the Lovelace test was not passed. If the AlphaGo AI were to perform a task not programmed, like beating all comers at the simple game of Parcheesi, the Lovelace test would be passed. But as it stands, AlphaGo is not creative. It can only perform the task it was trained for, namely playing Go. If asked, AlphaGo is unable to even explain the rules of Go. This said, AI can appear smart when it generates a surprising result. But surprise does not equate to creativity.

Larry Nobles:

When a computer program is asked to search through a billion designs to find the best, the result can be a surprise. But that isn't creativity. The computer program has done exactly what it was programmed to do.

Larry Nobles:

The Sacrificial Dweeb. Here's another example from my personal experience. The Office of Naval Research contracted Ben Thompson of Penn State's Applied Research Lab and me and asked us to evolve swarm behavior. As we saw in chapter one, simple swarm rules can result in unexpected swarm behavior, like stacking Skittles. Given simple rules, finding the corresponding emergent behavior is easy. Just run a simulation. But the inverse design problem is a more difficult one. If you want a swarm to perform some task, what simple rules should the swarm bugs follow?

Larry Nobles:

To solve this problem, we applied an evolutionary computing, AI. This process ended up looking at thousands of possible rules to find a set that gave the closest solution to the desired performance. One problem we looked at involved a predator prey swarm. All action took place in a closed, square virtual room. Predators called Bullies ran around chasing prey called Dweebs. Bullies captured Dweebs and killed them. We wondered what performance would be if the goal was maximizing the survival time of the Dweeb swarm. The swarm's survival time was measured up to when the last Dweeb was killed.

Larry Nobles:

After running the evolutionary search, we were surprised by the result. The Dweebs submitted themselves to self-sacrifice in order to maximize the overall life of the swarm. This is what we saw. A single Dweeb captured the attention of all the Bullies who chased the Dweeb in circles around the room. Around and around they went, adding seconds to the overall life of the swarm. During the chase, all the other Dweebs huddled in the corner of the room shaking with what appeared to be fear. Eventually, the pursuing Bullies killed the sacrificial Dweeb and pandemonium broke out as the surviving Dweebs scattered in fear.

Larry Nobles:

Eventually, another sacrificial Dweeb was identified, and the process repeated. The new sacrificial Dweeb kept the Bullies running around in circles while the remaining Dweebs coward in the corner. The sacrificial Dweeb result was unexpected. A complete surprise. There was nothing written in the evolutionary computer code explicitly calling for these sacrificial Dweebs. Is this an example of AI doing something we hadn't programmed it to do? Did it pass the Lovelace test? Absolutely not.

Larry Nobles:

We had programmed the computer to sort through millions of strategies that would maximize the life of the Dweeb swarm, and that's what the computer did. It evaluated options and chose the best one. The result was a surprise, but does not pass the Lovelace test for creativity. The program did exactly what it was written to do, and the seemingly frightened Dweebs were not in reality shaking with fear. Humans tend to project human emotions onto non-sentient things. The Dweebs were rapidly adjusting to stay as far away as possible from the closest Bully. They were programmed to do this.

Larry Nobles:

If the sacrificial Dweeb action and the unexpected Go move against Lisa Dole do not pass the Lovelace test, what would? The answer is anything outside of what the code was programmed to do. Here's an example from the predatory prey swarm example. The Lovelace test would be passed if some Dweebs became aggressive and started attacking and killing lone Bullies, a potential action we didn't program into the suite of possible strategies. But that didn't happen. And because the ability of a Dweeb to kill a Bully is not written into the code, it will never happen. Likewise, without additional programming, AlphaGo will never engage opponent Lisa Dole in trash talk or psychoanalyze Dole to get a game edge. Either of those things would be sufficiently creative to pass the Lovelace test. But remember, the AlphaGo software as written couldn't even provide an explanation of its own programmed behavior, the game of Go.

Austin Egbert:

You've been listening to an excerpt from Chapter Two of Non-Computable You by Robert J. Marks. You can find the rest of the book in print, Kindle, and audio formats on Amazon.com. Oh, and be sure to leave us an enthusiastic five-star review so we know you enjoyed it. Thanks for listening. And until next time, be of good cheer.