Can an A.I. Make Plans?

Cal Newport

Last summer, AdamYedidia, a user on a Web forum called LessWrong, published a post titled "Chess as a Case Study in Hidden Capabilities in ChatGPT." He started by noting that the Internet is filled with funny videos of ChatGPT playing bad chess: in one popular clip, the A.I. confidently and illegally moves a pawn backward. But many of these videos were made using the original version of OpenAI's chatbot, which was released to the public in late November, 2022, and was based on the GPT-3.5 large language model. Last March, OpenAI introduced an enhanced version of ChatGPT based on the more powerful GPT-4. As the post demonstrated, this new model, if prompted correctly, could play a surprisingly decent game of chess, achieving something like an Elo rating of 1000—better than roughly fifty per cent of ranked players. "ChatGPT has fully internalized the rules of chess," he asserted. It was "not relying on memorization or other, shallower patterns."

This distinction matters. When large language models first vaulted into the public consciousness, scientists and journalists struggled to find metaphors to help explain their eerie facility with text. Many eventually settled on the idea that these models "mix and match" the incomprehensibly large quantities of text they digest during their training. When you ask ChatGPT to write a poem about the infinitude of prime numbers, you can assume that, during its training, it encountered many examples of both prime-number proofs and rhyming poetry, allowing it to combine information from the former with the patterns observed in the latter. ("I'll start by noting Euclid's proof, / Which shows that primes aren't just aloof.") Similarly, when you ask a large language model, or L.L.M., to summarize an earnings report, it will know where the main points in such documents can typically be found, and then will rearrange them to create a smooth recapitulation. In this view, these technologies play the role of redactor, helping us to make better use of our existing thoughts.

But after the advent of GPT-4—which was soon followed by other next-generation A.I. models, including Google's PaLM-2 and Anthropic's Claude 2.1—the mix-and-match metaphor began to falter. As the LessWrong post emphasizes, a large language model that can play solid novice-level chess probably isn't just copying moves that it encountered while ingesting books about chess. It seems likely that, in some hard-to-conceptualize sense, it "understands" the rules of the game—a deeper accomplishment. Other examples of apparent L.L.M. reasoning soon followed, including acing SAT exams, solving riddles, programming video games from scratch, and explaining jokes. The implications here are potentially profound. During a talk at M.I.T., Sébastien Bubeck, a Microsoft researcher who was part of a team that systematically studied the abilities of GPT-4, described these developments: "If your perspective is, 'What I care about is to solve problems, to think abstractly, to comprehend complex ideas, to reason on new elements that arrive at me,' then I think you have to call GPT-4 intelligent," he said.

Yet intertwined with this narrative of uneasy astonishment is an intriguing counterpoint. There remain some surprisingly simple tasks that continue to stymie L.L.M.s. In his M.I.T. talk, Bubeck described giving GPT-4 the math equation "7 x 4 + 8 x 8 = 92." He then asked it to modify exactly one number on the left-hand side so that the equation would instead evaluate to 106. For a person, this problem is straightforward: change "7 x 4" to "7 x 6." But GPT-4 couldn't figure it out, and provided an answer that was clearly wrong. "The arithmetic is shaky," Bubeck said.

How can these powerful systems beat us in chess but falter on basic math? This paradox reflects more than just an idiosyncratic design quirk. It points toward something fundamental about how large language models think. Given the predicted importance of these tools in our lives, it's worth taking a moment to pull on this thread. To better understand what to expect from A.I. systems in the future, in other words, we should start by better understanding what the dominant systems of today still cannot do.

How does the human brain tackle a math problem like the one that Bubeck used to stump GPT-4? In his M.I.T. talk, he described how our thinking might unfold. Once we recognize that our goal is to increase the sum on the right side of the equation by fourteen, we begin searching for promising options on the left side. "I look at the left, I see a seven," Bubeck said. "Then I have kind of a eureka moment. Ah! Fourteen is seven times two. O.K., so if it's seven times two, then I need to turn this four into a six."

To us, this type of thinking is natural—it's just how we figure things out. We might overlook, therefore, the degree to which such reasoning depends on anticipation. To solve our math problem, we have to look into the future and assess the impact of various changes that we might make. The reason the "7 x 4" quickly catches our attention is that we intuitively simulate what will happen if we increase the number of sevens. "It was through some kind of planning," Bubeck concluded, of his solution process. "I was thinking ahead about what I'm gonna need."

We deploy this cognitive strategy constantly in our daily lives. When holding a serious conversation, we simulate how different replies might shift the mood—just as, when navigating a supermarket checkout, we predict how slowly the various lines will likely progress. Goal-directed behavior more generally almost always requires us to look into the future to test how much various actions might move us closer to our objectives. This holds true whether we're pondering life's big decisions, such as whether to move or have kids, or answering the small but insistent queries that propel our workdays forward, such as which to-do-list item to tackle next.

Presumably, for an artificial intelligence to achieve something like human cognition, it would also need to master this kind of planning. In "2001: A Space Odyssey," the self-aware supercomputer *HAL* 9000 refuses Dave's request to "open the pod bay doors" because, we can assume, it simulates the possible consequences of this action and doesn't like what it discovers. The ability to consider the future is inextricable from our colloquial understanding of real intelligence. All of which points to the importance of GPT-4's difficulty with Bubeck's math equation. The A.I.'s struggle here was not a fluke. As it turns out, a growing body of research finds that these cutting-edge systems consistently fail at the fundamental task of thinking ahead.

Consider, for example, the research paper that Bubeck was presenting in his M.I.T. talk. He and his team at Microsoft Research ran a pre-release version of GPT-4 through a series of systematic intelligence tests. In most areas, the model's performance was "remarkable." But tasks that involved planning were a notable exception. The researchers provided GPT-4 with the rules of Towers of Hanoi, a simple puzzle game in which you move disks of various sizes between three rods, shifting them one at a time without ever placing a larger disk above a smaller one. They then asked the model to tackle a straightforward instance of the game that can be solved in five moves. GPT-4 provided an incorrect answer. As the researchers noted, success in this puzzle requires you to look ahead, asking whether your current move might lead you to a future dead end.

In another example, the researchers asked GPT-4 to write a short poem in which the last line uses the same words as the first, but in reverse order. Furthermore, they specified that all of the lines of the poem needed to make sense in both grammar and content. For example:

> Darkness requires light,
> Toward this truth our imagination takes flight.
> But let us not forget, as we take solace,
> Light requires Darkness.

Humans can easily handle this task: the above poem, terrible as it is, satisfies the prompt and took me less than a minute to compose. GPT-4, on the other hand, stumbled. When Bubeck's team asked it to attempt the assignment, the chatbot started its poem with the line "I heard his voice across the crowd"—an ill-advised decision that led, inevitably, to the nonsensical concluding line "Crowd the across voice his heard I." To succeed in this poem-writing challenge, you need to think about writing your last line before you compose your first. GPT-4 wasn't able to peer into the future that way. "The model relies on a local and greedy process of generating the next word, without any global or deep understanding of the task or the output," the researchers wrote.

Bubeck's team wasn't the only one to explore the planning struggle. In December, a paper presented at Neural Information Processing Systems, a prominent artificial-intelligence conference, asked several L.L.M.s to tackle "commonsense planning tasks," including rearranging colored blocks into stacks ordered in specific ways and coming up with efficient schedules for shipping goods through a network of cities and connecting roads. In all cases, the problems were designed to be easily solvable by people, but also to require the ability to look ahead to understand how current moves might alter what's possible later. Of the models tested, GPT-4 performed best; even it was able to achieve only a twelve-per-cent success rate.

These problems with planning aren't superficial. They can't be fixed by making L.L.M.s bigger, or by changing how they're trained. They reflect something fundamental about the way these models operate.

A system like GPT-4 is outrageously complicated, but one way to understand it is as a supercharged word predictor. You feed it input, in the form of text, and it outputs, one at a time, a string of words that it predicts will extend the input in a rational manner. (If you give a large language model the input "Mary had a little," it will likely output "lamb.") A.I. applications like ChatGPT are wrapped around large language models such as GPT-4. To generate a long response to your prompt, ChatGPT repeatedly invokes its underlying model, growing the output one word at a time.

To choose their words, language models start by running their input through a series of pattern recognizers, arranged into sequential layers. As the text proceeds through this exegetical assembly line, the model incrementally builds up a sophisticated internal representation of what it's being asked about. It might help to imagine that the model has a vast checklist containing billions of possible properties; as the input text is processed by the model, it is checking off all of the properties that seem to apply. For example, if you provide GPT-4 with a description of a chessboard and ask it to make a move, the model might check off properties indicating that the input is about a game, that the game is chess, and that the user is asking for a move. Some properties might be related to more specific information, such as the fact that the board described in the input has a white knight on space E3; others might encode abstract observations, like the role that the white knight in space E3 is playing in protecting its king.