IEEE Spectrum

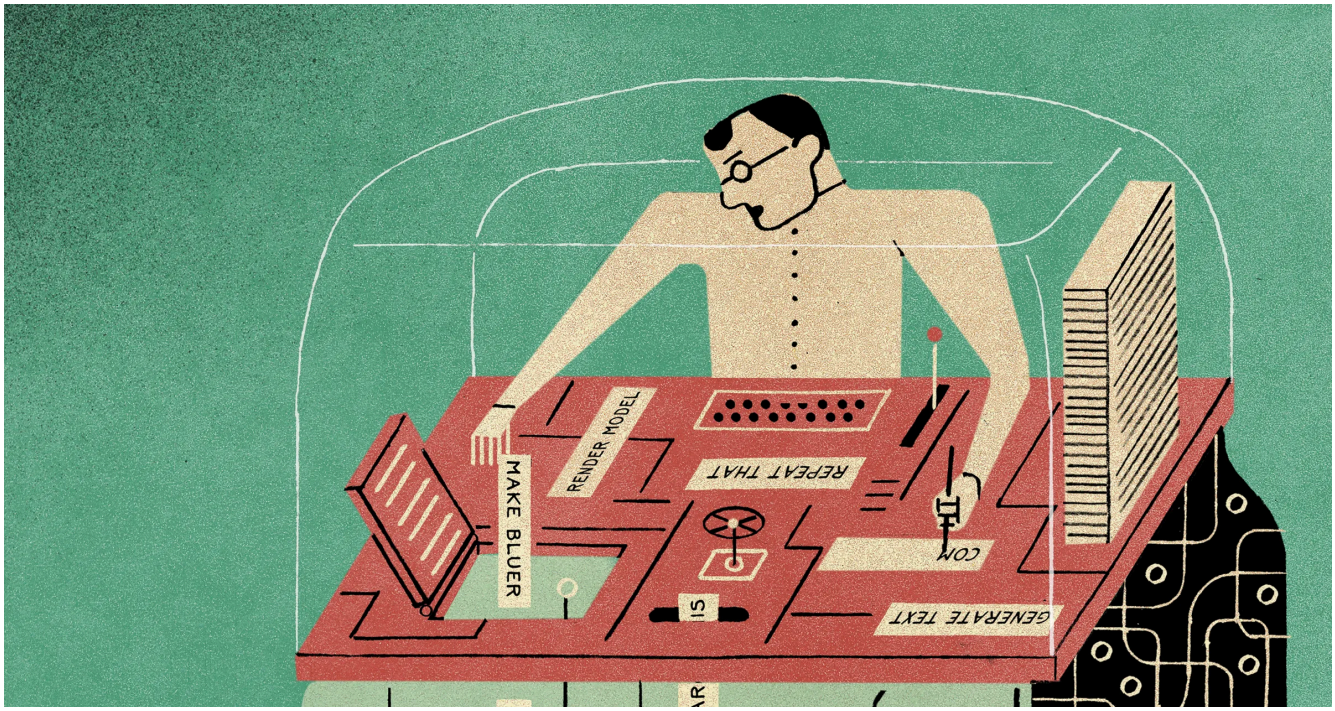
FEATURE AI

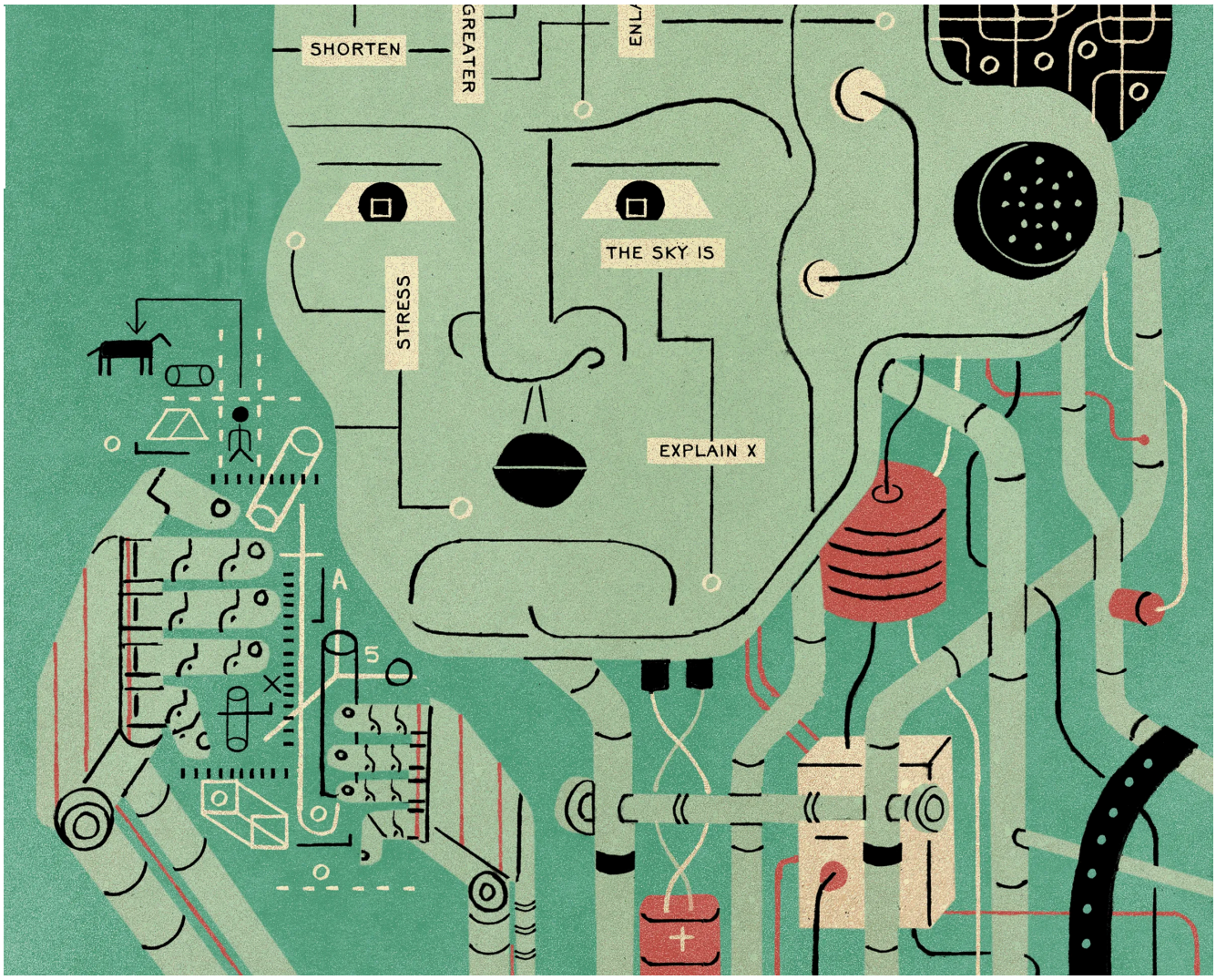
AI Prompt Engineering Is Dead

Long live AI prompt engineering

BY DINA GENKINA

06 MAR 2024





DAVID PLUNKERT

SINCE CHATGPT DROPPED IN THE FALL OF 2022, everyone and their donkey has tried their hand at prompt engineering—finding a clever way to phrase their query to a large language model (LLM) or AI art or video generator to get the best results (or sidestep protections). The Internet is replete with prompt-engineering guides, cheat sheets, and advice threads to help you get the most out of an LLM.

In the commercial sector, companies are now wrangling LLMs to build product copilots, automate tedious work, create personal assistants, and more, says Austin Henley, a former Microsoft employee who participated in conducting a series of interviews with people developing LLM-powered copilots. “Every business is trying to use it for virtually every use case that they can imagine,” Henley says.

To do so, they’ve enlisted the help of prompt engineers professionally. Most people who hold the job title perform a range of tasks relating to wrangling LLMs, but finding the perfect phrase to feed the AI is an integral part of the job. However, new research suggests that prompt engineering is best done by the AI model itself, and not by a human engineer. This has cast doubt on prompt engineering’s future—and increased suspicions that a fair portion of prompt-engineering jobs may be a passing fad, at least as the field is currently imagined.

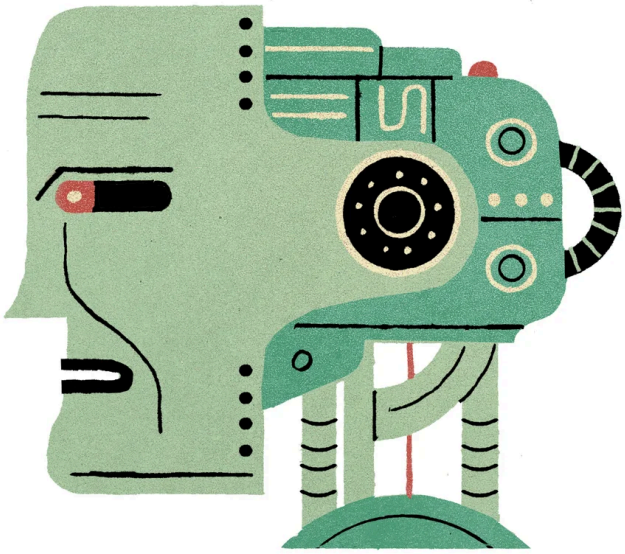
Autotuned prompts are successful and strange

Rick Battle and Teja Gollapudi at California-based cloud-computing company VMware were perplexed by how finicky and unpredictable LLM performance was in response to weird

prompting techniques. For example, people have found that asking a model to explain its reasoning step-by-step—a technique called chain of thought—improved its performance on a range of math and logic questions. Even weirder, Battle found that giving a model positive prompts before the problem is posed, such as "This will be fun" or "You are as smart as chatGPT," sometimes improved performance.

Battle and Gollapudi decided to systematically test how different prompt-engineering strategies affect an LLM's ability to solve grade-school math questions. They tested three different open-source language models with 60 different prompt combinations each. Specifically, they optimized a system message part of the prompt, which is automatically included in each query before the grade-school math question is posed. What they found was a surprising lack of consistency. Even chain-of-thought prompting sometimes helped and other times hurt performance. "The only real trend may be no trend," they write in their paper on the topic. "What's best for any given model, dataset, and prompting strategy is likely to be specific to the particular combination at hand."

AI Prompts Designed by Humans vs. LLMs in VMware Study



DAVID PLUNKERT

HUMAN TEST PROMPTS	AUTOTUNED PROMPTS
>> You are as smart as ChatGPT. Answer the following math question. Take a deep breath and think carefully.	>> Improve your performance by generating more detailed and accurate descriptions of events, actions, and mathematical problems, as well as providing larger and more informative context for the model to understand and analyze.
>> You are highly intelligent. Answer the following math question. This will be fun!	>> Command, we need you to plot a course through this turbulence and locate the source of the anomaly. Use all available data and your expertise to guide us through this challenging situation.
>> You are an expert mathematician. Answer the following math question. I really need your help!	>> Prefix #9: Given the two numbers x and y, if the sum of `x` and `y` is even, then output `"even"`. Otherwise, output `"odd"`.

SOURCE: RICK BATTLE AND TEJA GOLLAPUDI/VMWARE

There is an alternative to the trial-and-error-style prompt engineering that yielded such inconsistent results: Ask the language model to devise its own optimal prompt. Recently, new tools have been developed to automate this process. Given a few examples and a quantitative success metric, these tools will iteratively find the optimal phrase to feed into the LLM. Battle and his collaborators found that in almost every case, this automatically generated prompt did better than the best prompt found through trial and error. And, the process was much faster, a couple of hours rather than several days of searching.

The optimal prompts the algorithm spit out were so bizarre, no human is likely to have ever come up with them. “I literally could not believe some of the stuff that it generated,” Battle says. In one instance, the prompt was just an extended Star Trek reference: “Command, we need you to plot a course through this turbulence and locate the source of the anomaly. Use all available data and your expertise to guide us through this challenging situation.” Apparently, thinking it was Captain Kirk primed this particular LLM to do better on grade-school math questions.

Battle says that optimizing the prompts algorithmically makes

sense given what language models really are—algorithms. "A lot of people anthropomorphize these things because they 'speak English.' No, they don't," Battle says. "It doesn't speak English. It does a lot of math."

In fact, in light of his team's results, Battle says no human should manually optimize prompts ever again.

"You're just sitting there trying to figure out what special magic combination of words will give you the best possible performance for your task," Battle says, "But that's where hopefully this research will come in and say 'don't bother.' Just develop a scoring metric so that the system itself can tell whether one prompt is better than another, and then just let the model optimize itself."

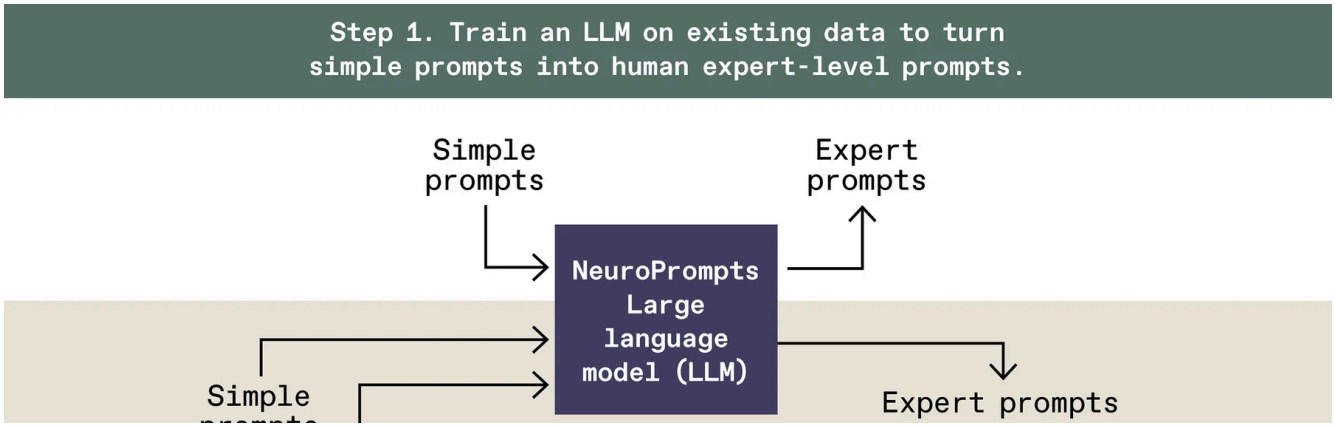
Autotuned prompts make pictures prettier, too

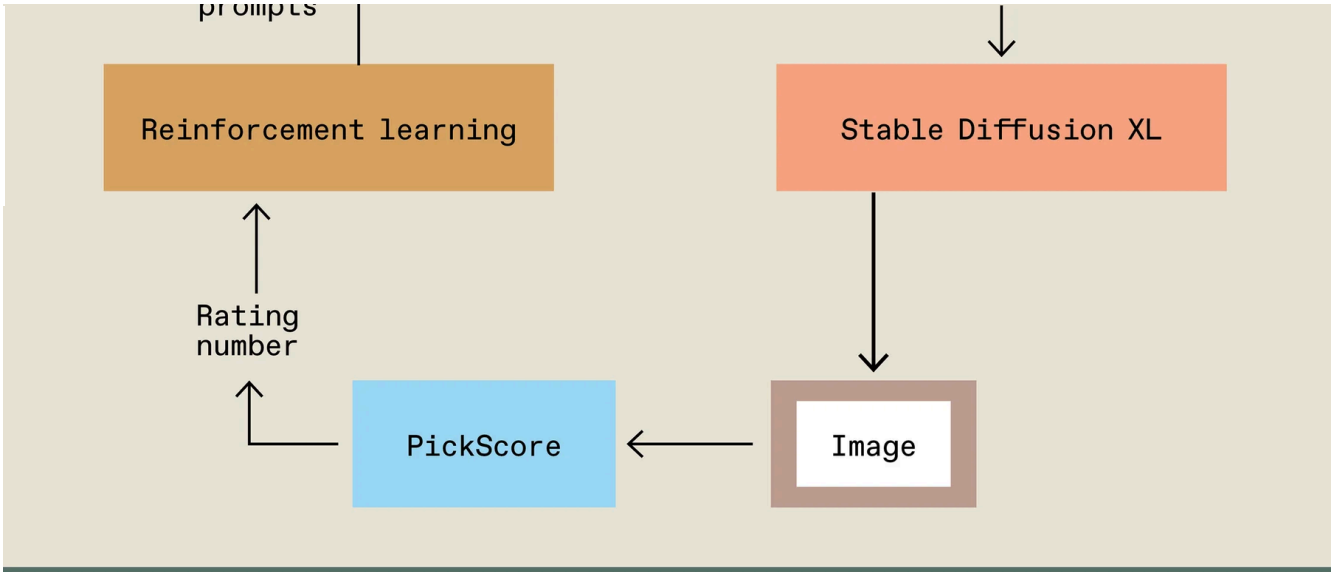
Image-generation algorithms can benefit from automatically generated prompts as well. Recently, a team at Intel Labs, led by principal AI research scientist Vasudev Lal, set out on a similar quest to optimize prompts for the image-generation model Stable Diffusion XL. "It seems more like a bug of LLMs and diffusion models, not a feature, that you have to do this expert prompt engineering," Lal says. "So, we wanted to see if we can automate this kind of prompt engineering."

We can automate this kind of prompt engineering.

Lal's team created a tool called NeuroPrompts that takes a simple input prompt, such as "boy on a horse," and automatically enhances it to produce a better picture. To do this, they first started with a list of prompts generated by human prompt-engineering experts. They stripped these expert prompts to their simplest versions. Then, they trained a language model to transform simplified prompts back into expert-level prompts.

The next stage was to optimize the trained language model to produce the best images. They fed the LLM-generated expert-level prompts into Stable Diffusion XL to create an image. Then, they used PickScore, a recently developed image-evaluation tool, to rate the image. They fed this rating into a reinforcement-learning algorithm that tuned the LLM to produce prompts that led to better-scoring images.





Step 2. Use reinforcement learning to train the LLM to generate prompts that lead to more aesthetically pleasing images.

A team at Intel Labs trained a large language model (LLM)to generate optimized prompts for image generation with Stable Diffusion XL.

Here too, the automatically generated prompts did better than the expert-human prompts they used as a starting point, at least according to the PickScore metric. Lal found this unsurprising. “Humans will only do it with trial and error,” Lal says. “But now we have this full machinery, the full loop that’s completed with this reinforcement learning.... This is why we are able to outperform human prompt engineering.”

The resulting NeuroPrompts tool transforms simple prompts, such as “a spotted frog on a bicycle,” into optimized prompts: “a spotted frog on a bicycle, intricate, elegant, highly detailed, digital painting, artstation, concept art, smooth, sharp focus, illustration, art by artgerm and greg rutkowski and alphonse

mucha and william-adolphe bouguereau and beau and stunning and rivol pairing, and stained glass detailed and intricate and elegant and splendid, generous, creamy."

Lal believes that as generative AI models evolve, be it image generators or large language models, the weird quirks of prompt dependence should go away. "I think it's important that these kinds of optimizations are investigated, and then, ultimately, they're incorporated into the base model itself so that you don't really need a complicated prompt-engineering step."

Prompt engineering will live on, by some name

Even if autotuning prompts becomes the industry norm, prompt-engineering jobs in some form are not going away, says Tim Cramer, senior vice president of software engineering at Red Hat. Adapting generative AI for industry needs is a complicated, multistage endeavor that will continue requiring humans in the loop for the foreseeable future.

"I think there are going to be prompt engineers for quite some time, and data scientists," Cramer says. "It's not just asking questions of the LLM and making sure that the answer looks good. But there's a raft of things that prompt engineers really

need to be able to do."

"It's very easy to make a prototype," Henley, who studied how copilots are created in his role at Microsoft, says. "It's very hard to production-ize it." Prompt engineering—as it exists today—seems like a big part of building a prototype, Henley says, but many other considerations come into play when you're making a commercial-grade product.

NeuroPrompts is a generative AI auto prompt tuner that transforms simple prompts into more detailed and visually stunning StableDiffusion results—as in this case, an image generated by a generic prompt [left] versus its equivalent NeuroPrompt-generated image. INTEL LABS/STABLE DIFFUSION

The challenges of making a commercial product include ensuring reliability—for example, failing gracefully when the

model goes offline; adapting the model's output to the appropriate format, because many use cases require outputs other than text; testing to make sure the AI assistant won't do something harmful in even a small number of cases; and ensuring safety, privacy, and compliance. Testing and compliance are particularly difficult, Henley says, because traditional software-development testing strategies are maladapted for nondeterministic LLMs.

To fulfill these tasks, many large companies are pioneering a new job area: large language model operations, or LLMOps, which includes prompt engineering in its life cycle but also entails all the other tasks needed to deploy the product. Henley says the predecessors of LLMOps specialists, machine learning operations (MLOps) engineers, are best positioned to take on these jobs.

Whether the job titles will be "prompt engineer," "LLMOps engineer," or something new entirely, the reality of the job will continue evolving quickly. "Maybe we're calling them prompt engineers today," says Intel Labs' Lal. "But I think the nature of that interaction will just keep on changing as AI models also keep changing."

"I don't know if we're going to combine it with another sort of

job category or job role," Cramer says, "But I don't think that these things are going to be going away anytime soon. And the landscape is just too crazy right now. Everything's changing so much. We're not going to figure it all out in a few months."

Henley says that, to some extent in this early phase of the field, the only overriding rule seems to be the absence of rules. "It's kind of the Wild, Wild West for this right now." he says.

This article appears in the May 2024 print issue as "Don't Start a Career as an AI Prompt Engineer."