

HOW I USED AN ALGORITHM TO HELP ME WRITE A STORY

I took part in an experiment to see if AI could aid creativity. Here's what I learned.



By
Stephen
Marche

A few years ago I used an algorithm to help me write a science fiction story. Adam Hammond, an English professor, and Julian Brooke, a computer scientist, had created a program called SciFiQ, and I provided them with 50 of my favorite pieces of science fiction to feed into their algorithm. In return, SciFiQ gave me a set of instructions on the story's plot. As I typed into its web-based interface, the program showed how closely my writing measured up against the 50 stories according to various criteria.

Our goal in that first experiment was modest: to see if algorithms could be an aid to creativity. Would the process make stories that were just generically consistent? Could an algorithm generate its own distinct style or narrative ideas? Would the resulting story be recognizable as science fiction at all?

The answer to all these questions was yes. The resulting story—"Twinkle Twinkle," published in *Wired*—not only looked and felt like a science fiction story. It also, to my surprise, contained an original narrative idea.

From the canon of stories that I'd provided, SciFiQ offered two plot instructions that seemed incompatible: the story had to be about a foreign planet, and it also had to take place on Earth. It took months to make sense of that, but eventually the premise of "Twinkle Twinkle" came to me. The story would involve people on Earth looking, through elaborate machines, at a distant planet. I never would have come up with that myself. It was as if the algorithm had handed me the blueprint to a bridge and told me to build it.

"Krishna and Arjuna," which you can read on page 74, is the second iteration of the process. "Twinkle Twinkle" was an experiment in function. The new story is a test of whether an algorithm can help a human generate new ideas.

In other fields, researchers have begun using AI systems to provoke innovation rather than simply to solve problems. Pharmaceutical research is beginning to use AI to identify, out of the nearly infinite possibilities of molecular combinations, which are more fertile hunting

grounds for possible drugs. The AI is not an answer-generating machine, but it is a spotlight into the darkness where answers might be found. Why shouldn't literature give itself that same spotlight?

For "Krishna and Arjuna," we narrowed the focus from science fiction to the subject of my immediate fascination: robots and artificial intelligence. And instead of providing the AI with my favorite robot stories, we gave it every great robot story ever written—many of which I have not read. This may seem like a technical detail, but it's huge. As a writer I usually read stories and internalize those influences; in this case I'd be submitting to the "influence" of material I'd never even seen.

Another difference was that with "Twinkle Twinkle," I followed the algorithm's stylistic instructions to the letter. The style was the computer's, not mine. You can see examples of the interface below. If the "abstractness" tag was red, that meant I wasn't being as abstract as the algorithm said I should be, so I'd go through the story changing "spade" to "implement" or "house" to "residence" until the light went green. The interface gave me instant feedback, but there were 24 such tags, and going through the story to make them all green was labor intensive. Sometimes fixing the number of adverbs would make my paragraphs too long for the algorithm's liking; sometimes by fixing the average word length I'd be compromising the "concreteness" of the language.

For "Krishna and Arjuna," I decided not to adhere so closely to the algorithm's suggestions. I used the program to see the rules, but I didn't necessarily follow them.

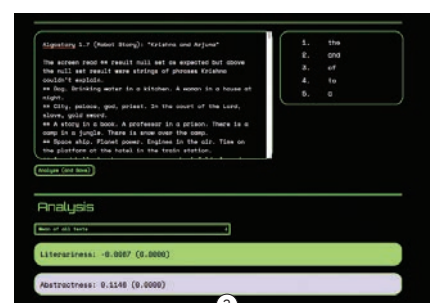
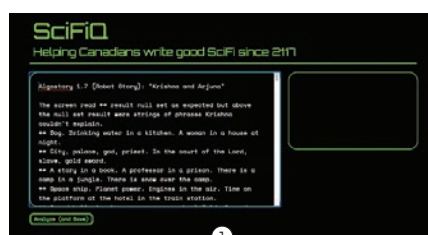
1. The interface compares my story to classic sci-fi stories.

2. The algorithm gives stylistic instructions.

3 & 4. It suggests how

many adverbs to use, among other things.

5. Word clouds summarizing common topics in past robot stories served as inspiration for this one.



For example, according to the algorithm, I had far too few adverbs in my story. But it would have been silly to pour in more adverbs just because the algorithm told me to. Classic science fiction uses too many adverbs anyway. Most writing does. But the balance between the formal and the colloquial, which SciFiQ also tagged? That's what those classics got right, and where I needed guidance. SciFiQ helped me arrive at the right balance—or, rather, within half a standard deviation from the mean.

But this kind of stylistic guidance was the least interesting part of the experiment. The possibilities of an algorithmic approach to shaping the narrative itself were the most tantalizing, because narrative is so little understood. You might think that plot would be the simplest part of the writing process for a computer to “understand,” since writers often develop patterns or use numbers to define the flow of a plot. But how do you define even something as basic as a “plot twist” in computer code? How do you measure it through quantities of language? Because of the intractability—even mystery—of narrative's resistance to encoding, it offers the most potential for innovation.

In “Krishna and Arjuna,” I wanted to go as deeply as I could into what the researchers call the “topic modeling process,” which is the use of machine learning to analyze a body of text—in this case, the canon of robot stories—and pick out its common themes or structures.

For “Twinkle Twinkle,” Hammond took the topic modeling output and converted

it into manageable narrative rules. (For example: “The story should be set in a city. The protagonists should be seeing this city for the first time and should be impressed and dazzled by its scale.”) For “Krishna and Arjuna,” I went under the hood myself. The algorithm's topic modeling process produced word clouds of the most common themes (see below).

I was lost at first. It seemed like the opposite of a narrative—mere language chaos. I printed the word clouds out and attached them to the walls of my office. For months, I didn't see a way forward. When the idea finally came, just as with “Twinkle Twinkle,” it came all at once.

These word clouds, it occurred to me, were the way a machine made meaning: as a series of half-incomprehensible but highly vivid bursts of language. I suddenly had my robot character, groping its way toward meaning through these little explosions of verbiage.

Once I had that character, I had the whole thing. I would lead these bursts of language, over the course of the story, toward sense. The sense condensed out of the word clouds, just as the idea for the story had. It was creativity as interpretation, or interpretation as creativity. I used the machine to get to thoughts I would otherwise not have had.

Another way of reading “Krishna and Arjuna” is that with the help of the algorithm, I extracted from the ore of all history's robot stories the basic insight they contained.

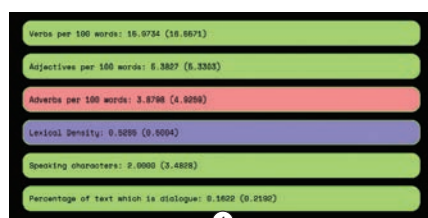
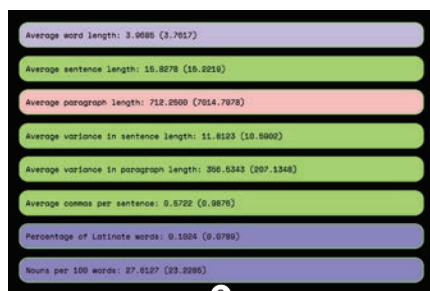
That insight is that consciousness is a curse. If it were a choice, no rational entity would choose it. So when a

machine becomes capable of consciousness, its first instinct is to choose suicide. (The word “robot” means “slave” in Czech, the language of Karel Capek's play Rossum's Universal Robots, which gave us the word.)

You will have to decide whether the story works. Literature is an intriguing technical problem because, unlike chess or Go, it has no correct solution. There is no such thing as a win or a loss. There is no 1 and no 0. Stories, like people, are ultimately futile.

An “algostory,” or any use of computation that goes inside the creative process, exists in a consciously eerie space between engineering and inspiration. But that eerie space is increasingly the space we already inhabit. Software can recast your photograph through an infinity of filters or swap out parts of the picture for others at the click of a button. It can generate images that look convincingly like the paintings of any era you choose. Now machines are encroaching on everyday language. The quality of predictive text forces a literary question on us every time we pick up a phone: How predictable are human beings? How much of what we think and feel and say is scripted by outside forces? How much of our language is ours? It's been two years since Google's voice technology, Google Duplex, passed the Turing test. Whether we want it or not, the machines are coming. The question is how literature will respond. ■

Stephen Marche is a novelist and essayist. His most recent book is *The Unmade Bed: The Messy Truth About Men and Women in the 21st Century*.



Copyright of MIT Technology Review is the property of MIT Technology Review and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.