What Is IA ChatGPT ... and How Does It Work?

By Stephen Wolfram (excerpts)

It's Just Adding One Word at a Time

That ChatGPT can automatically generate something that reads even superficially like human-written text is remarkable, and unexpected. But how does it do it? And why does it work? The purpose here is to give a rough outline of what's going on inside ChatGPT—and then to explore why it is that it can do so well in producing what we might consider to be meaningful text. I should say at the outset that I'm going to focus on the big picture of what's going on—and while I'll mention some engineering details, I won't get deeply into them. (And the essence of what I'll say applies just as well to other current "large language models" [LLMs] as to ChatGPT.)

The first thing to explain is that what ChatGPT is always fundamentally trying to do is to produce a "reasonable continuation" of whatever text it's got so far, where by "reasonable" we mean "what one might expect someone to write after seeing what people have written on billions of webpages, etc."

So let's say we've got the text "The best thing about AI is its ability to". Imagine scanning billions of pages of human-written text (say on the web and in digitized books) and finding all instances of this text—then seeing what word comes next what fraction of the time. ChatGPT effectively does something like this, except that (as I'll explain) it doesn't look at literal text; it looks for things that in a certain sense "match in meaning". But the end result is that it produces a ranked list of words that might follow, together with "probabilities":

And the remarkable thing is that when ChatGPT does something like write an essay what it's essentially doing is just asking over and over again "given the text so far, what should the next word be?"—and each time adding a word. (More precisely, as I'll explain, it's adding a "token", which could be just a part of a word, which is why it can sometimes "make up new words".)

But, OK, at each step it gets a list of words with probabilities. But which one should it actually pick to add to the essay (or whatever) that it's writing? One might think it should be the "highest-ranked" word (i.e. the one to which the highest "probability" was assigned). But this is where a bit of voodoo begins to creep in. Because for some reason—that maybe one day we'll have a scientific-style understanding of—if we

always pick the highest-ranked word, we'll typically get a very "flat" essay, that never seems to "show any creativity" (and even sometimes repeats word for word). But if sometimes (at random) we pick lower-ranked words, we get a "more interesting" essay.

The fact that there's randomness here means that if we use the same prompt multiple times, we're likely to get different essays each time. And, in keeping with the idea of voodoo, there's a particular so-called "temperature" parameter that determines how often lower-ranked words will be used, and for essay generation, it turns out that a "temperature" of 0.8 seems best. (It's worth emphasizing that there's no "theory" being used here; it's just a matter of what's been found to work in practice. And for example the concept of "temperature" is there because exponential distributions familiar from statistical physics happen to be being used, but there's no "physical" connection—at least so far as we know.)

Where Do the Probabilities Come From?

OK, so ChatGPT always picks its next word based on probabilities. But where do those probabilities come from? Let's start with a simpler problem. Let's consider generating English text one letter (rather than word) at a time. How can we work out what the probability for each letter should be?

But let's now assume—more or less as ChatGPT does—that we're dealing with whole words, not letters. There are about 40,000 reasonably commonly used words in English. And by looking at a large corpus of English text (say a few million books, with altogether a few hundred billion words), we can get an estimate of how common each word is. And using this we can start generating "sentences", in which each word is independently picked at random, with the same probability that it appears in the corpus.

Not surprisingly, this is nonsense. So how can we do better? Just like with letters, we can start taking into account not just probabilities for single words but probabilities for pairs or longer n-grams of words.

It's getting slightly more "sensible looking". And we might imagine that if we were able to use sufficiently long n-grams we'd basically "get a ChatGPT"—in the sense that we'd get something that would generate essay-length sequences of words with the "correct overall essay probabilities". But here's the problem: there just isn't even

close to enough English text that's ever been written to be able to deduce those probabilities.

In a crawl of the web there might be a few hundred billion words; in books that have been digitized there might be another hundred billion words. But with 40,000 common words, even the number of possible 2-grams is already 1.6 billion—and the number of possible 3-grams is 60 trillion. So there's no way we can estimate the probabilities even for all of these from text that's out there. And by the time we get to "essay fragments" of 20 words, the number of possibilities is larger than the number of particles in the universe, so in a sense they could never all be written down.

For the complete article, go to What Is ChatGPT Doing … and Why Does It Work?—Stephen Wolfram Writings