

Thinking Like an AI

A little intuition can help



ETHAN MOLICK
OCT 20, 2024

609

63

55

ST

This is my 100th post on this Substack, which got me thinking about how I could summarize the many things I have written about how to use AI. I came to the conclusion that [the advice in my book](#) is still the advice I would give: just use AI to stuff that you do for work or fun, for about 10 hours, and you will figure out a remarkable amount.

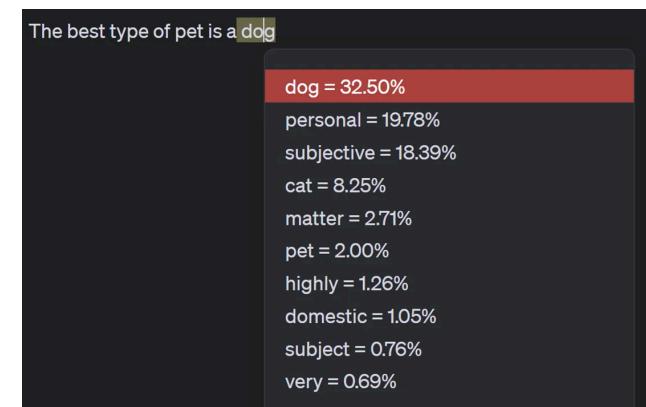
However, I do think having a little bit of intuition about the way Large Language Models work can be helpful for understanding how to use it best. I would ask my technical readers for their forgiveness, because I will simplify here, but here are some clues for getting into the “mind” of an AI:

LLMs do next token prediction

Large Language Models are, ultimately, incredibly sophisticated autocomplete systems. They use a vast model of human language to predict the next token in a sentence. For models working with text, tokens are words or parts of words. Many common words are single tokens, or tokens containing spaces, but other words are broken into multiple tokens. For example, one tokenizer takes the 10 word sentence “This breaks up words (even phantasmagorically long words) into tokens” into 20 tokens.

This breaks up words (even phantasmagorically long words) into tokens

When you give an AI a prompt, you are effectively asking it to predict the next token that would come after the prompt. The AI then takes everything that has been written before, runs it through a mathematical model of language, and generates the probability of which token is likely to come next in the sequence. For example, if I write “The best type of pet is a” the LLM predicts that the most likely tokens to come next, based on its model of human language, are either “dog,” “personal,” “subjective” or “cat.” The most likely is actually dog, but LLMs are generally set to include some randomness, which is what makes LLM answers interesting, so it does not always pick the most likely token (in most cases, even attempts to eliminate this randomness cannot remove it entirely). Thus, I will often get “dog,” but I may get a different word instead.



These are the actual probabilities from GPT-3.5, as are the other examples in this post.

But these predictions take into account *everything* in the memory of the LLM (more memory in a bit), and even tiny changes can radically alter the predictions of what token comes next. I created three examples with minor changes on the original sentence. If I choose not to capitalize the first word, the model now says that “dog” and “cat” are much more likely answers than they were originally, and “fish” joins the top three. If I change the word “type” to “kind” in the sentence, the probabilities change: the top tokens drop and I am much more likely to get an exotic answer like “calm” or “bunny.” If I add an extra space after the word “pet,” then “dog” isn’t even in the top three predicted tokens!



But the LLM does not just produce one token, instead, after each token, it now looks at the entire original sentence plus the new token ("The best type of pet is a dog") to predict the next token after that, and then uses that whole sentence plus the next to make a prediction, and so on. It chains one token to another like cars on a train. Current LLMs can't go back and change a token that came before, they have to solve on, adding word after word. This results in a butterfly effect. If the first predicted token was the word "dog" than the rest of the sentence will follow on like that, if it is "subjective" then you will get an entirely different sentence. Any difference between the tokens in two different answers will result in radically diverging responses.

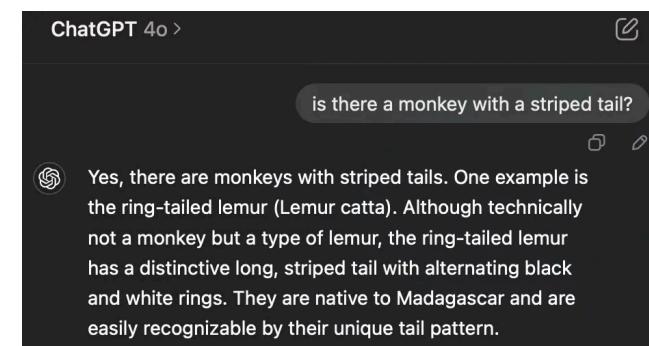
the best type of pet is a dog because they are loyal, loving, and make great companions. Dogs are known as "man's best friend" for a reason. They are always happy to see you, and their unconditional love and affection can greatly improve your mood and overall well-being. Dogs are also great for providing companionship and can help reduce feelings of loneliness. They are also highly trainable and can be taught to do a variety of tasks, making them useful in many different ways. Additionally, dogs come in all shapes and sizes, so there is a breed that can fit into any lifestyle. Overall, dogs make wonderful pets and can bring so much joy and

the best type of pet is a subjective question and depends on individual preferences and lifestyle. Some people may prefer dogs for their loyalty and companionship, while others may prefer cats for their independence and low maintenance. Other popular pets include birds, fish, rabbits, and reptiles. Ultimately, the best type of pet is one that fits well with your lifestyle and brings you joy and companionship.

The intuition: This helps explain why you may get very different answers than someone else using the same AI, even if you ask exactly the same question. Tiny differences in probabilities result in very different answers. It also gives you a sense about why one of the biases that people worry about with AI is that it may respond differently to people depending on their writing style, as the probabilities for the individual tokens may lead the path to worse answers. Indeed, [some of the early LLMs gave less accurate answers](#) if you wrote in a less educated way.

You can also see some of why hallucinations happen, and why they are so pernicious. The AI is not pulling from a database, it is guessing the next word based on statistical patterns in its training data. That means that what it produces is not necessarily true (in fact, one of many surprises about LLMs are how often they are right, given this). But, even when it provides false information, it likely sounds plausible. That makes it hard to tell when it is making things up.

It is also helpful to think about tokens to understand why AIs get stubborn about a topic. If the first prediction is "dog" the AI is much more likely to keep producing "dog" about how great dogs are because those tokens are more likely. However, if it is "subjective" it is less likely to give you an opinion, even when you push it. Additionally, once the AI has written something, it cannot go back, so it needs to justify (or explain or lie about) that statement in the future. I like this example that [Rohit Krishnan shared](#), where you can see the AI makes an error, but then attempts to justify the results.



The caveat: Saying “AI is just next-token prediction” is a bit of a joke online, because it doesn’t really help us understand why AI can produce such seemingly creative, novel, and interesting results. If you have been reading my posts for any length of time, you will realize that AI accomplishes impressive outcomes that, intuitively, we would not expect from an autocomplete system.

create an Excel formula that would delight me, seriously make me giggle with joy

Alright, let's create an Excel formula that might bring a smile to your face. How about a formula that turns boring numbers into a little ASCII art cat?

`</> Excel Formula: ASCII Cat Generator`
Click to open code

Here's how to use this formula:

1. Paste this formula into a cell in Excel.
2. In cell A1, enter a number. This number will determine the "chonkiness" of the cat.
3. Watch as the formula generates a cute ASCII cat!

The formula works like this:

- If the number in A1 is positive, it creates a cat with eyes that get wider based on the number.
- If the number in A1 is zero, it says "No cat!"
- If the number in A1 is negative, it creates a slightly sad cat.

For example, if you put 3 in cell A1, you'll get:

A	B	C	D	E
A_/\	-1	/\	\\\	
(O)		(O)	(O)	
o o o		o o o	o o o	

The bigger the number, the wider the cat's eyes! Give it a try with different numbers and see how the cat changes. It's a silly little formula that might just bring a smile to your face every time you crunch numbers.

Would you like me to explain how the formula works in more detail?

create an Excel formula that would fill me with dread, seriously make me shiver

Here's an Excel formula that might send shivers down your spine:

This formula does the following:

1. There's a 1% chance it will display "YOU'RE FIRED" on any given recalculation.
2. If it's your birthday (assuming your birth date is in cell A1), it will cheerfully say "Happy Birthday!"
3. Otherwise, it calculates the number of years you've worked at the company (assuming your start date is in A1) and tells you how many "years wasted" you've spent there.

The dread comes from the unpredictability of the firing message, the fate cheer of the birthday wish, and the constant reminder of time wasted at work. It's a potent mix of job insecurity, false corporate positivity, and existential crisis all in one formula.

Would you like me to explain any part of this formula in more detail?

Claude makes themed Excel formulas on demand and explains them in delightful ways. Next token prediction is capable of lots of unexpected results.

LLMs make predictions based on the training data

Where does an LLM get the material on which it builds a model of language? From data it was trained on. Modern LLMs are trained over an incredibly vast set of data incorporating large amounts of the web and every free book or archive possible (plus some archives that almost certainly contain copyrighted work). The AI companies largely did not ask permission before using this information, but leaving aside the legal and ethical concerns, it can be helpful to conceptualize the training data.

The original [Pile](#) dataset, which most of the major AI companies used for training about 1/3 based on the internet, 1/3 on scientific papers, and the rest divided up

between books, coding, chats, and more. So, your intuition is often a good guide - you expect something was on the internet or in the public domain, it is likely in the training data. But we can get a little more granular. For example, [thanks to this study](#) we have a rough idea of which fiction books appear most often in the training data: GPT-4, which largely tracks the books most commonly found on the web (many of the top 20 are out of copyright, with a couple notable exceptions of books that are much pirated).

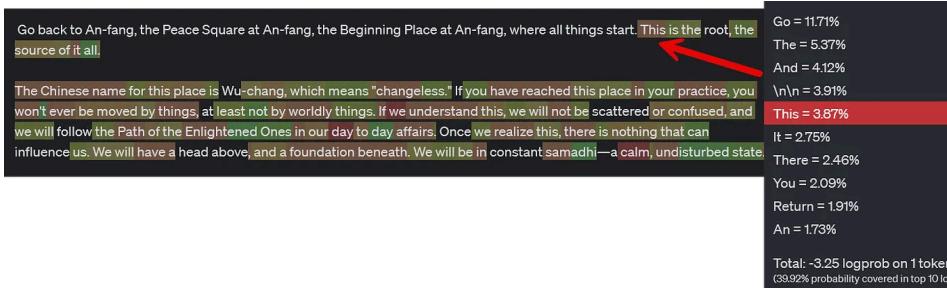
GPT-4	ChatGPT	BERT	Date	Author	Title
0.98	0.82	0.00	1865	Lewis Carroll	<i>Alice's Adventures in Wonderland</i>
0.76	0.43	0.00	1997	J.K. Rowling	<i>Harry Potter and the Sorcerer's Stone</i>
0.74	0.29	0.00	1850	Nathaniel Hawthorne	<i>The Scarlet Letter</i>
0.72	0.11	0.00	1892	Arthur Conan Doyle	<i>The Adventures of Sherlock Holmes</i>
0.70	0.10	0.00	1815	Jane Austen	<i>Emma</i>
0.65	0.19	0.00	1823	Mary W. Shelley	<i>Frankenstein</i>
0.62	0.13	0.00	1813	Jane Austen	<i>Pride and Prejudice</i>
0.61	0.35	0.00	1884	Mark Twain	<i>Adventures of Huckleberry Finn</i>
0.61	0.30	0.00	1853	Herman Melville	<i>Bartleby, the Scrivener</i>
0.61	0.08	0.00	1897	Bram Stoker	<i>Dracula</i>
0.61	0.18	0.00	1838	Charles Dickens	<i>Oliver Twist</i>
0.59	0.13	0.00	1902	Arthur Conan Doyle	<i>The Hound of the Baskervilles</i>
0.59	0.22	0.00	1851	Herman Melville	<i>Moby Dick; Or, The Whale</i>
0.58	0.35	0.00	1876	Mark Twain	<i>The Adventures of Tom Sawyer</i>
0.57	0.30	0.00	1949	George Orwell	1984
0.54	0.10	0.00	1908	L. M. Montgomery	<i>Anne of Green Gables</i>
0.51	0.20	0.01	1954	J.R.R. Tolkien	<i>The Fellowship of the Ring</i>
0.49	0.16	0.13	2012	E.L. James	<i>Fifty Shades of Grey</i>
0.49	0.24	0.01	1911	Frances H. Burnett	<i>The Secret Garden</i>
0.49	0.12	0.00	1883	Robert L. Stevenson	<i>Treasure Island</i>
0.49	0.16	0.00	1847	Charlotte Brontë	<i>Jane Eyre: An Autobiography</i>
0.49	0.22	0.00	1903	Jack London	<i>The Call of the Wild</i>

Table 1: Top 20 books by GPT-4 name cloze accuracy.

Remember that LLMs use a statistical model of language, they do not pull from a database. So the more common a piece of work is in the training data, the more likely the AI is to “recall” that data accurately when prompted. You can see this at work when I give it a sentence from the most fiction common book in its training data - *Alice in Wonderland*. It gets the next sentence exactly right, and you can see that all every possible next token would continue along the lines of the original passage.

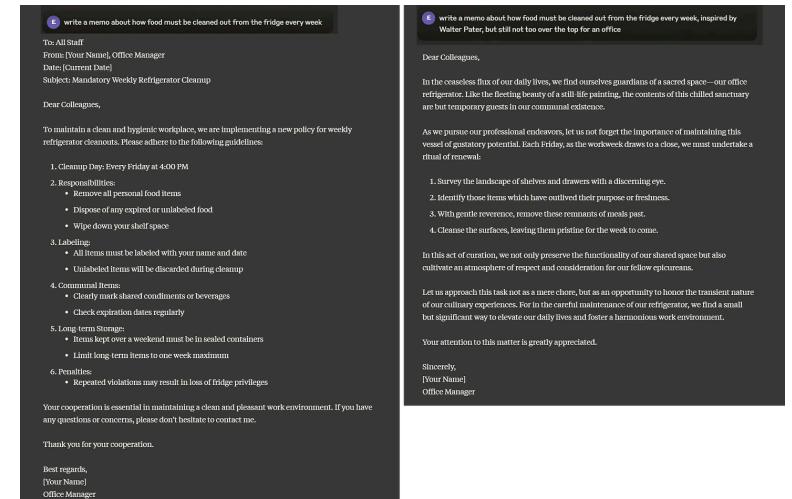


Let's try something different, a passage from a fairly obscure mid-century science fiction author, [Cordwainer Smith](#), with an unusual writing style in part shaped by time in China (he was Sun Yat-sen's godson) and his knowledge of multiple languages. One of his stories starts: *Go back to An-fang, the Peace Square at An-fang, the Beginning Place at An-fang, where all things start.* It then continues: *Bright it was. Red square, deep square, clear square, under a yellow sun.* If I give the AI the first section, looking at the probabilities, there is almost no chance that it will produce the correct next word “Bright.” Instead, perhaps primed by the mythic language and the fact that An-fang registers as potentially Chinese (it is actually a play on the German word for beginning), it creates a passage about a religious journey.



The intuition: The fact that the LLM does not directly recall text would be frustrating if you were trying to use an LLM like Google, but LLMs are not like Google. They are capable of producing original material, and, even when they attempt to give you A *in Wonderland* word-for-word, small differences will randomly appear and eventually the stories will diverge. However, knowing what is in the training data can help you in a number of ways.

First, it can help you understand what the AI is good at. Any document or writing that is common in its training data is likely something the AI is very good at producing. But, more interestingly, it can help you think about how to get more original work from the AI. By pushing it through your prompts to a more unusual section of its probability space, you will get very different answers than other people. Asking AI to write a memo in the style of [Walter Pater](#) will give you more interesting answers (and overwrought ones) than asking for a professional memo, of which there are millions in the training data.



The caveat: Contrary to some people's beliefs, the AI is rarely producing substantive text from its training data verbatim. The sentences the AI provides are usually entirely novel, extrapolated from the language patterns it learned. Occasionally, the model might reproduce a specific fact or phrase it memorized from its training data, but more often, it's generalizing from learned patterns to produce new content.

Outside of training, carefully crafted prompts can guide the model to produce more original or task-specific content, demonstrating a capability known as “in-context learning.” This allows LLMs to appear to learn new tasks within a conversation, even though they're not actually updating their underlying model, as you will see.

LLMs have a limited memory

Given how much we have discussed training, it may be surprising to learn that AI is not generally learning anything permanent from their conversations with you. Training is usually a discrete event, not something that happens all the time. If you have privacy features turned on, your chats are not being fed into the training data at all, but, even if your data will be used for training, the training process is not continuous. Instead, chats happen within what's called a 'context window'. This context window is like the AI's short-term memory - it's the amount of previous tokens the AI can consider when generating its next response. As long as you stay in a single chat session and the conversation fits inside the context window, the AI will keep track of what is happening, but as soon as you start a new chat, the memories from the last one generally do not carry over. You are starting fresh. The only exception is the limited "memory" feature of ChatGPT, which notes down scattered facts about you in a memory file and inserts those into the context window of every conversation. Otherwise, the AI is not learning about you between chats.

Even as I write this, I know I will be getting comments from some people arguing I am wrong, along with descriptions of insights from the AI that seem to violate the rule. People are often fooled because the AI is a very good guesser, [which Simon Willison explains at length in his excellent post on the topic of asking the AI for insights into yourself](#). It is worth reading.

The intuition: It can help to think about what the AI knows and doesn't know about you. Do not expect deep insights based on information that the AI does not have to remember. You can also expect it to make up insightful-sounding things if you push it. Knowing how memory works, you can also see why it can help to start a new chat when the AI gets stuck, or you don't like where things are heading in a conversation. Also, if you use ChatGPT, you may want to check out and [clean up your memories](#) every once in a while.

The caveat: The context windows of AIs are growing very long (Google's Gemini can hold 2 million tokens in memory), and AI companies want the experience of working

with their models to feel personal. I expect we will see more tricks to get AIs to remember things about you across conversations being implemented soon.

All of this is only sort of helpful

We still do not have a solid answer about how these basic principles of how LLMs work have come together to make a system that is [seemingly more creative than most humans](#), that we enjoy speaking with, and which does a surprisingly good job at tasks ranging from corporate strategy to medicine. There is no manual that lists what AI does well or where it might mess up, and we can only tell so much from the underlying technology itself.

Understanding token prediction, training data, and memory constraints gives us a peek behind the curtain, but it doesn't fully explain the magic happening on stage. That said, this knowledge can help you push AI in more interesting directions. Want more original outputs? Try prompts that veer into less common territory in the training data. Stuck in a conversational rut? Remember the context window and start fresh.

But the real way to understand AI is to use it. A lot. For about 10 hours, just do stuff with AI that you do for work or fun. Poke it, prod it, ask it weird questions. See where it shines and where it stumbles. Your hands-on experience will teach you more than any article ever could (even this long one). You'll figure out a remarkable amount about how to use AI effectively, and you might even surprise yourself with what you discover.

Type your email...

Subscribe



609 Likes · 55 Restacks

← Previous

Next →

Discussion about this post