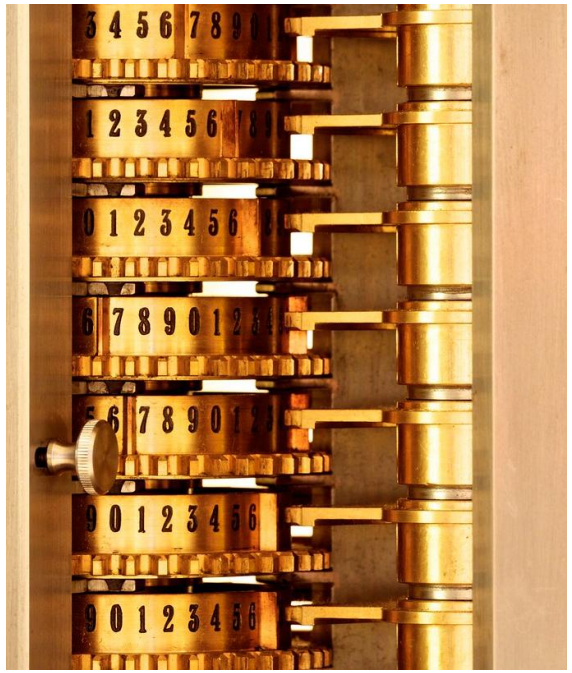
Computers, brains and AI

7: ChatGPT





Difference engine by Marcin Wichary / CC BY
Ada Lovelace by Science Museum Group / CC0

1) Intro

2) Foundations
3) Silicon

4) Deep learning
5) Deep learning

6) ChatGPT
7) ChatGPT (peek inside)
8) ChatGPT

9) The future of AI
10) The future of AI



How they work

Large language models generate text a word at a time.

or part of a word

One word at a time...





apples and pears ✓✓

apples and pairs ✓

apples and
- pears ✓✓
- pairs ✓

Deciding whether to generate “apples and pears” or “apples and pairs”, can be reduced to deciding between “pears” and “pairs” – in other words, by dealing with one word at a time.



Why is the sky blue?

How would you explain this to a five year old?

The sky looks blue because

The sky looks blue because

Humans might answer the question by saying, “the sky, um, looks, err, blue because…” **before** they’ve really thought of the answer.

Then, if they’re lucky, the answer will come to them, and they will continue. Or else, they might backtrack and say they don’t know.

A chatbot, on the other hand, will struggle on and possibly make something up.

The large language model's job is to give probabilities back to the chatbot, by saying that, in its experience of looking at a lot of text, the chance of "looks" is 60% etc.

The chatbot then has a choice. It could simply accept the most likely word ("looks", in this case). But always accepting the most likely word can result in text that is strangely bland and repetitive. So, instead, it rolls a (loaded) dice to decide between "looks" and "appears" and so on.

The dice rolling explains why ChatGPT's answer will vary.

Why is the sky blue?

• • •

How would you explain this to a five year old?

The chatbot can make things "colder" by freezing things more towards the front-runner. "Cold" could be good for precision, and "warm" could be good for more creative text.

The sky looks blue because • • •

The sky

looks	60%	78%
appears	20%	15%
is	10%	5%
⋮	2%	0.5%
	1%	0.2%



• • •

The capital of France is

Paris	99.99%
⋮	

The random nature of chatbots has inspired some people to insult them as “stochastic parrots”.

However, when things are more certain (like, say, the capital of France), the LLM will be distinctly less random.

Why is the sky blue?

• • •

How would you explain this to a five year old?

The sky looks blue because • • •

The sky

looks	60%	78%
appears	20%	15%
is	10%	5%
⋮	2%	0.5%
	1%	0.2%
	• • •	



Dice by Brett Jordan / Unsplash

or part of a word

One word at a time…



No backtracking

(until September 2024)

In late 2024, we began to see LLMs that examine, evaluate, and refine their own outputs. A form of reasoning, maybe?

https://en.wikipedia.org/wiki/OpenAI_o1

https://en.wikipedia.org/wiki/Reflection_(artificial_intelligence)

https://en.wikipedia.org/wiki/DeepSeek_(chatbot)

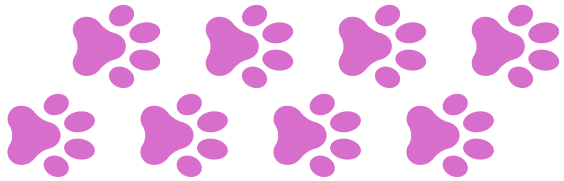


Once upon a time

Generating text one word at a time is a bit like the children's game where they take turns saying a word.

or part of a word

One word at a time...



?



50% cat

bus

tractor

20% dog

LLMs are surprisingly similar to AlexNet.

The sky

looks	60%
appears	20%
is	10%
⋮	2%
	1%
	⋯

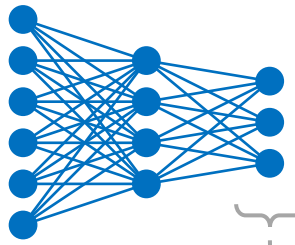


78%
15%
5%
0.5%
0.2%

Dice by Brett Jordan / Unsplash

One letter at a time...

Thinking a letter at a time would be too laborious.



text

pixel values

three “grandmother” cells

0 (dog)
1 (cat)
2 (bus)

A
B
C
Z

One word at a time...

Thinking a word at a time seems more promising, but, unfortunately, there would be too many different words (especially, if you include all the world’s languages)...

three categories

the
is
to

??? Unknown

Next bit of text

... therefore, a neat compromise is to think a word at a time for all the common words, but to split the more obscure words. The chunks that text is split into are called **tokens**.

https://platform.openai.com/tokenizer

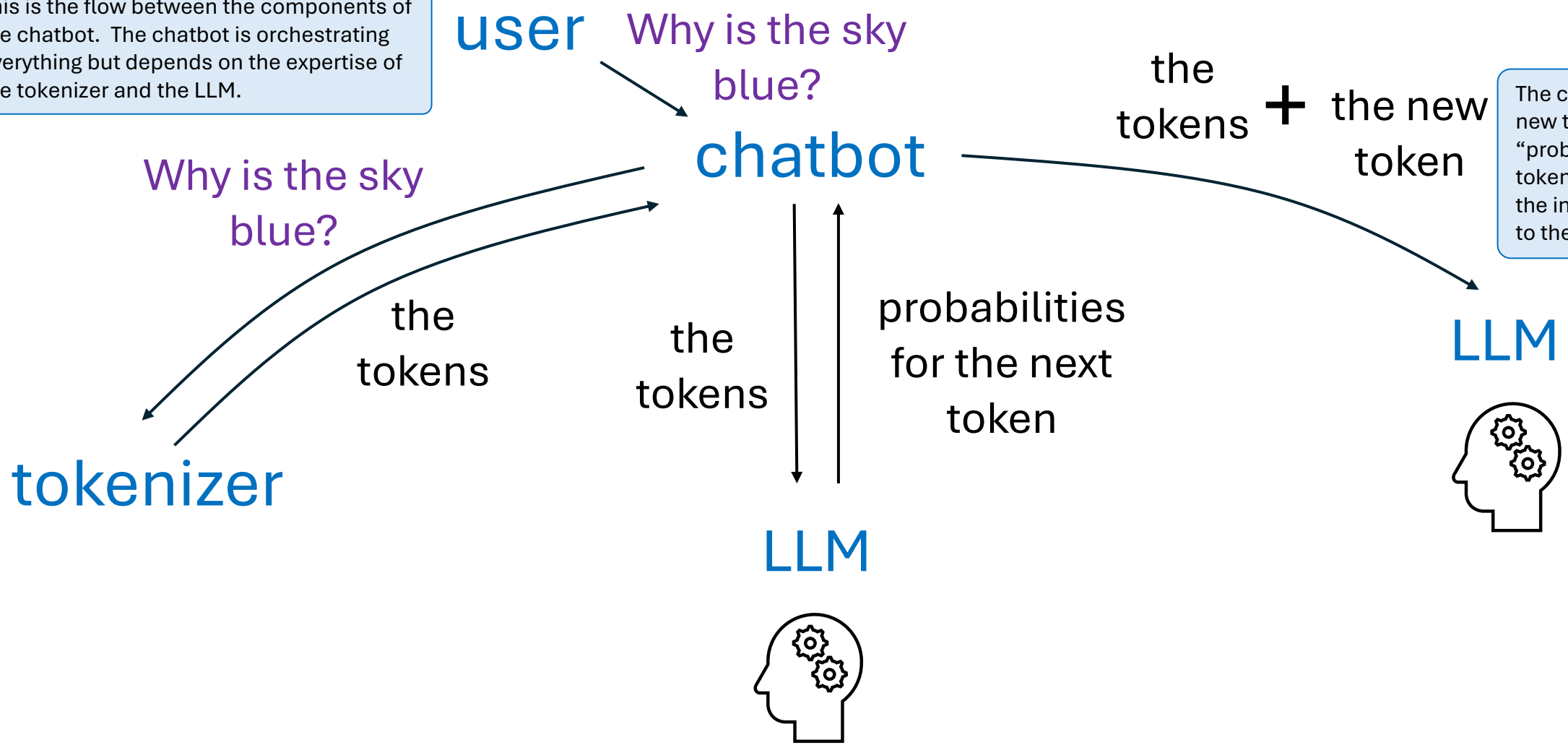
https://observablehq.com/@simonw/gpt-tokenizer

These online tokenizers are quite fun to play with.



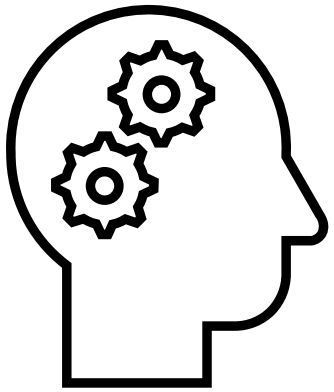
https://en.wiktionary.org/wiki/token#:~:text=The%20basic%20unit%20of%20data

This is the flow between the components of the chatbot. The chatbot is orchestrating everything but depends on the expertise of the tokenizer and the LLM.



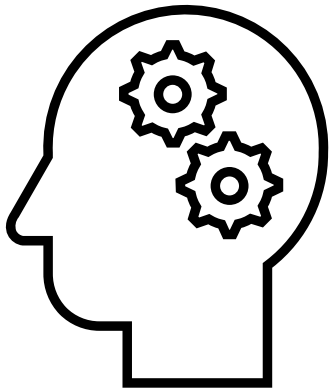
The chatbot chooses a new token using the LLM's "probabilities for the next token" and appends it to the input for the next call to the LLM.

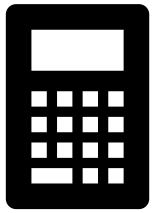
chatbot



We can think of the chatbot and the LLM as having a conversation like this.

LLM





The LLM does maybe trillions of arithmetic operations to arrive at the probabilities for the next token.

Give me the next token

OK

Give me the next token

OK

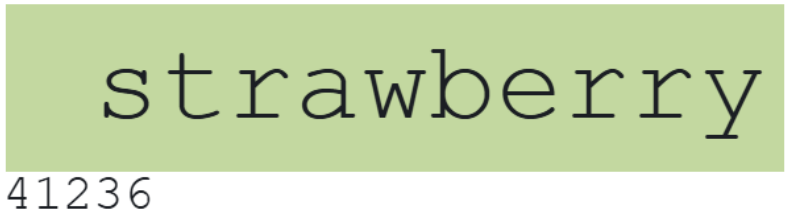
After a while, the LLM will return a special token whose meaning is “that’s all for now”, which signals the end of the response.



strawberry

strawberry

Famously, some chatbots didn't know how many "r"s there were in "strawberry". But maybe it is quite reasonable given how the LLM only receives a single numeric token instead of the individual letters.



These weaknesses are all the more surprising when it turns out that, for example, ChatGPT can pass the Bar Exam!

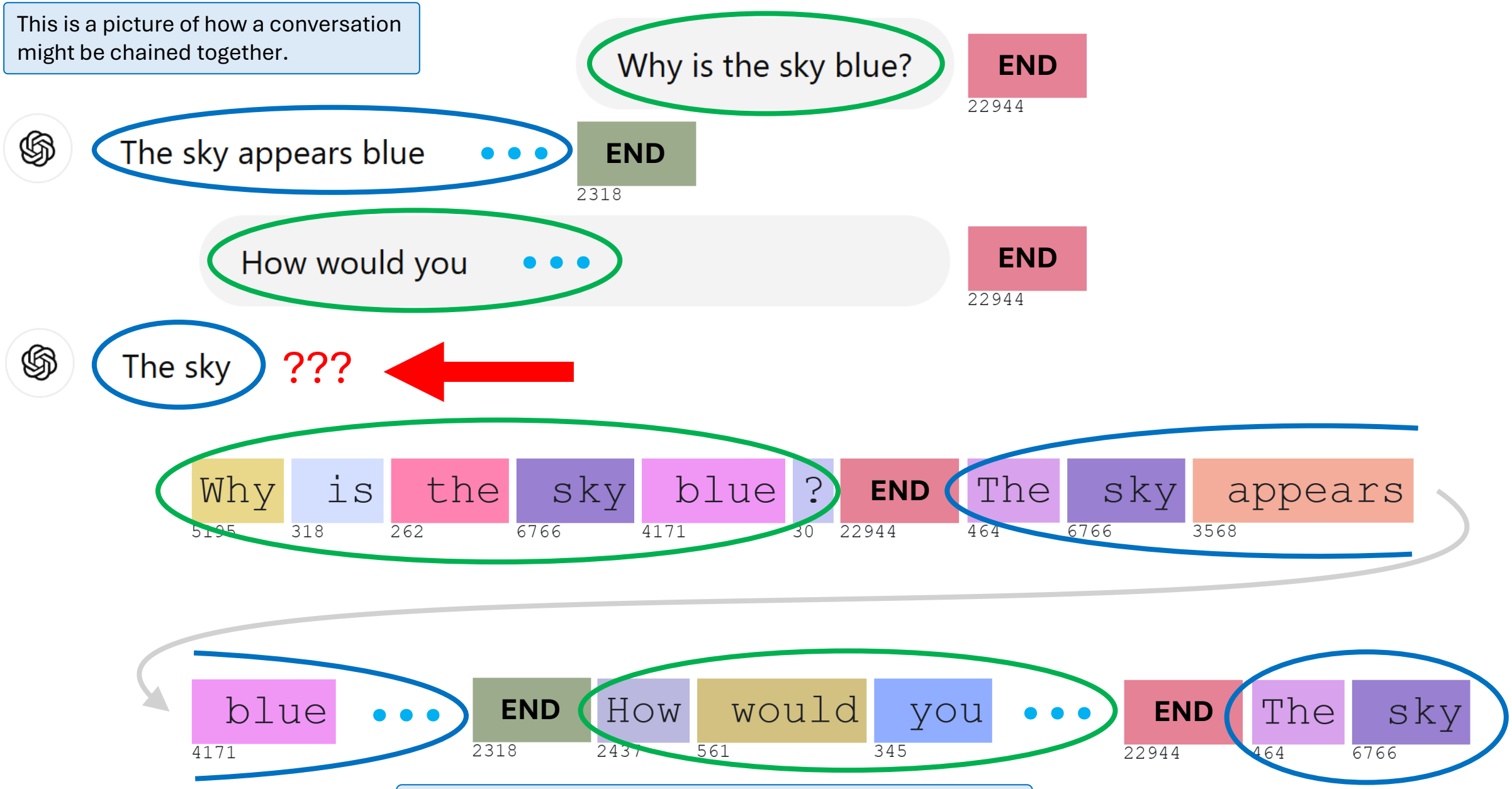
https://law.stanford.edu/2023/04/19/gpt-4-passes-the-bar-exam-what-that-means-for-artificial-intelligence-tools-in-the-legal-industry/

elephant

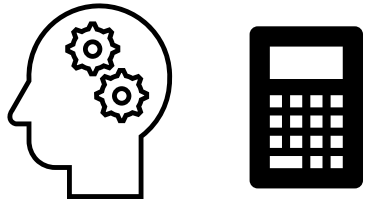
They also couldn't spell "elephant" backwards.



This is a picture of how a conversation might be chained together.



Because LLMs are started afresh, they need to receive a stream of input tokens from the preceding conversation, complete with the “END” tokens to indicate who said what.





Why
5195

is
318

the
262

next token

cat
bus
tractor
dog

1000 categories

200,000 different tokens

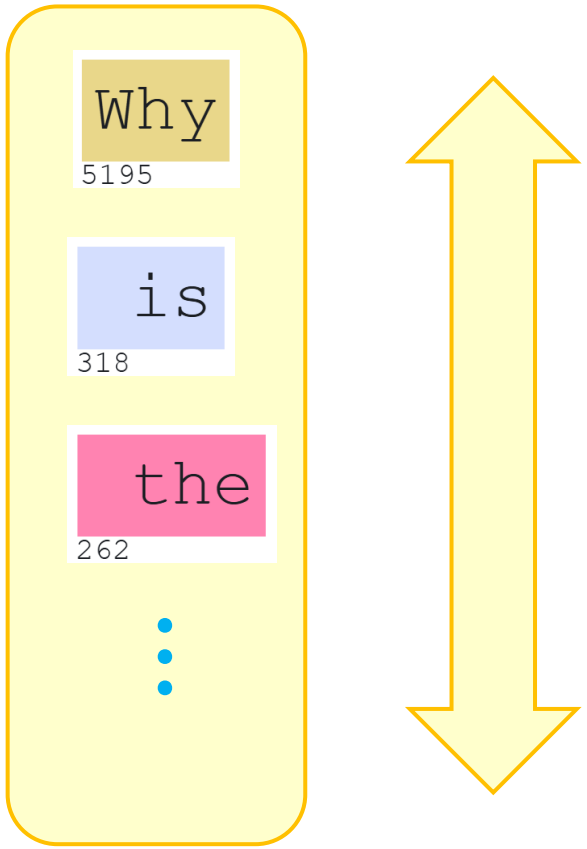
60 million parameters

175 billion

A language model guessing a token is like AlexNet guessing a classification, except the input is tokens instead of pixels.

In both cases, it is a feed-forward network: no going back and worrying about whether it is on the right track.

Context windows have got longer over the years. The one million token context window of Google Gemini is a feat of engineering and presumably involves some serious compromises.



The **context window** is the maximum number of input tokens a LLM can see.

Context window

GPT-3	2048
GPT-4	8192
	32,768
Gemini-1.5	1 million

From last time...

Why is the sky blue?

• • •

How would you explain this to a five year old?

The sky

looks	60%
appears	20%
is	10%
⋮	2%
	1%



• • •

The sky

looks	60%
appears	20%
is	10%
⋮	2%
	1%



probabilities

LLM

The sky s k y

17307

1.2, 0.26, -0.3, 4.1, ...

More than 12,000 of them!

The LLM has the tricky job of predicting the next token, and, to get started, it converts each token into a long list of numbers called an **embedding**.

LLM

chat bot

The sky

looks	60%
appears	20%
is	10%
⋮	
	2%
	1%



GPT-3 used embeddings of length 12,288 (which is 12 * 1024, or 12k).

Word embedding

By the time the LLM is fully trained, it will have arranged for similar words to have similar embeddings and therefore be close to each other.

dog

fox

bus

bank

Unfortunately, many words are ambiguous.

river bank

central bank

the

set

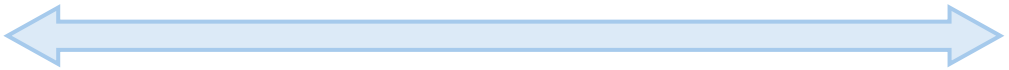
The record for ambiguity might go to the word, “set”. Here’s a screenshot from the OED.

Results for "set" in:

- Quotation work title (1,606)
- Quotation text (29,328)
- Meanings (1,093)
- Definitions (6,837)

https://www.oed.com/

1.2, 0.26, -0.3, 4.1, …



More than 12,000 of them!



word embedding

the answer

The answer (to the question of creating the next token) emerges at the top-right, as another embedding.

numbers numbers numbers numbers

numbers numbers numbers numbers

numbers numbers numbers numbers

The answer (the embedding) is translated back into token probabilities, by choosing the nearby tokens.

Word embeddings

side ✓

fox ✗

dog ✗

bank ✓

the ✗

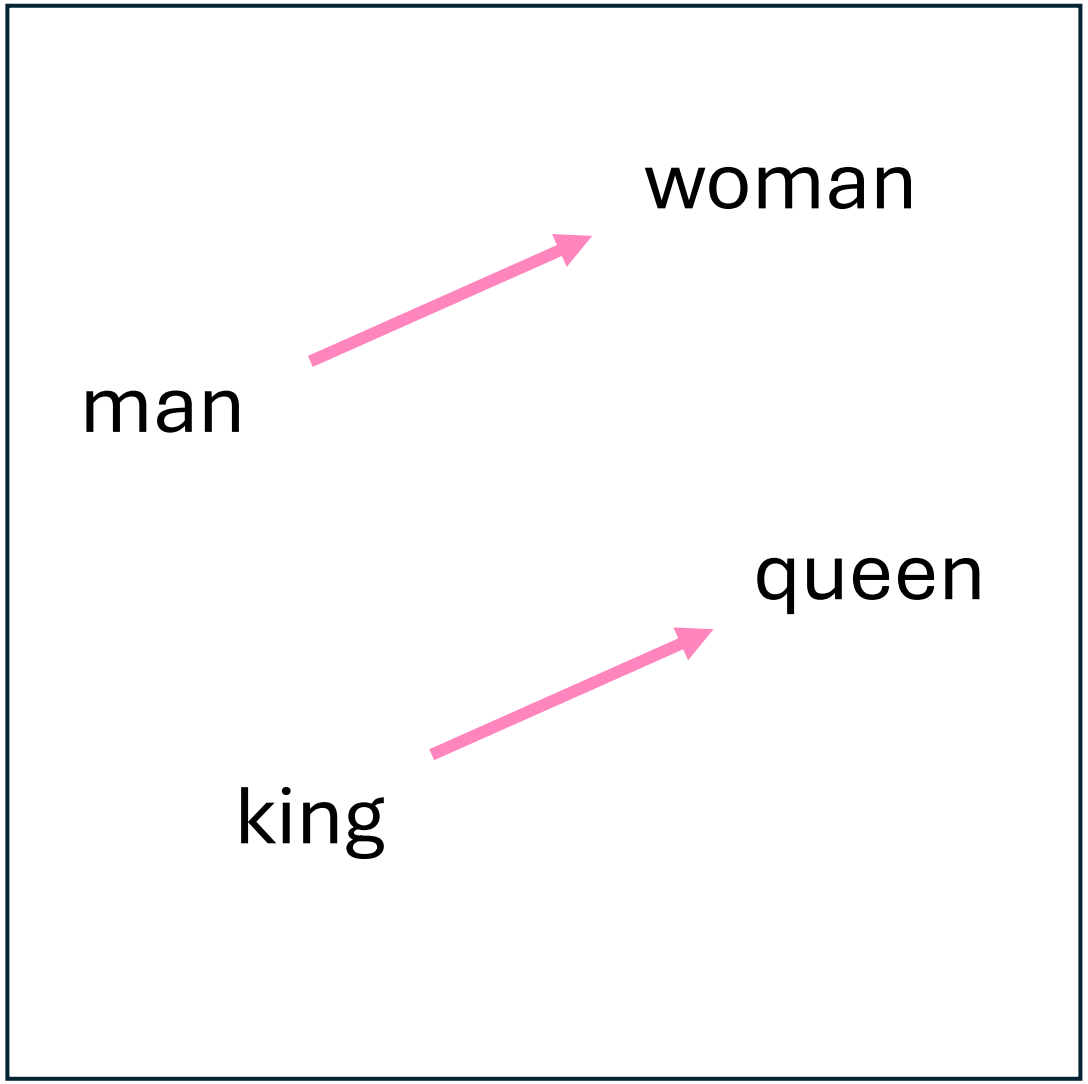
bus ✗

set ✗

1.2, 0.26, -0.3, 4.1, ...

swam to the other ...

Visualizing word embeddings



ChatGPT translates these sentences correctly because, as well as knowing about grammar, it knows things about the world, namely that animals can be tired, and roads can be wide.

Translate the following sentence into French. The animal didn't cross the road because it was too wide.

L'animal n'a pas traversé la route parce qu'elle était trop large.

Translate the following sentence into French. The animal didn't cross the road because it was too tired.

L'animal n'a pas traversé la route parce qu'il était trop fatigué.



Arithmetic

???

What is 123 plus 456 ? ✓

What is 314159 plus 271828 ?

The early language models were amazingly bad at arithmetic. Instead, they would just guess (in true **System 1** fashion).

GPT-3.5 ✗ guess confidently



GPT-4 ✓

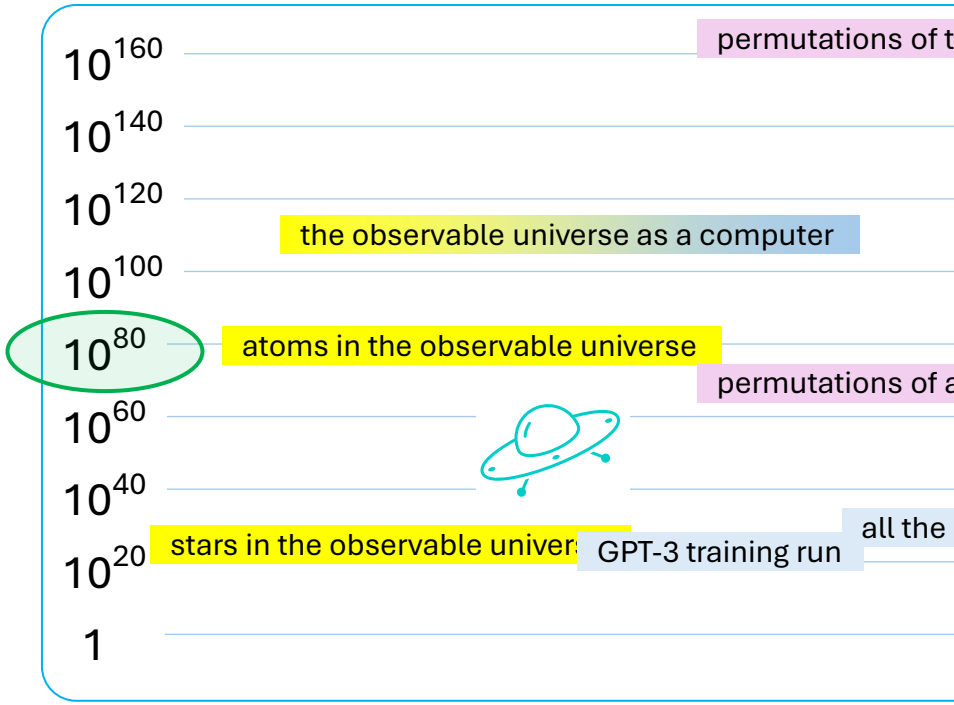
What is 3141592653589793238462643383279502884197 plus 271828182845904523536028747135266249775 7 ?

GPT-4 ✓

To everyone's surprise, GPT-4 somehow found a way to add 40-digit numbers.

later versions ✓ Use external tools

To get round this limitation, modern LLMs will subcontract to Python (the programming language), as shown on the next slide.





What is 59833 times 61897?

The result of 59833×61897 is **3,703,483,201**

When we click this button, it reveals the analysis below, namely, invoking a tiny Python program.

It may seem like cheating, but humans also prefer to use calculators than do the calculations themselves.

python

Analysis

python

```
# Recalculating after executior
result = 59833 * 61897
result
```

Always show details | Copy

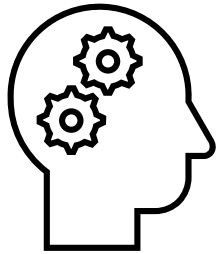
Result

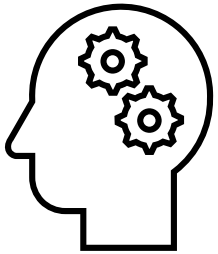
```
3703483201
```

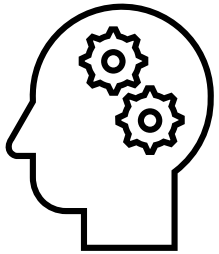
https://pythonhow.com/python-shell

ChatGPT

Large language model



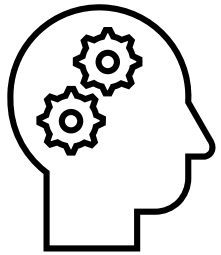


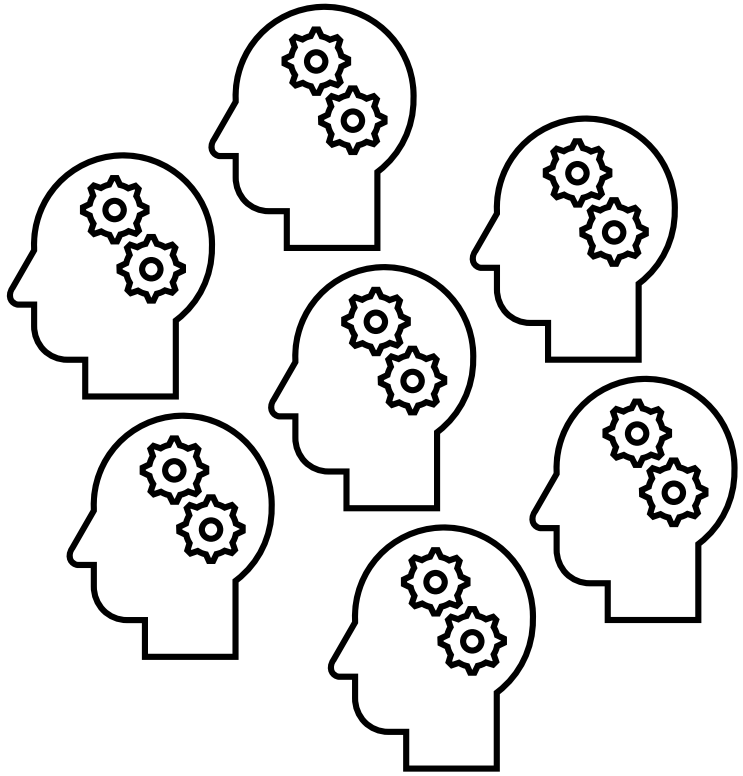


Python

We now have a third player in the conversation. There are other players, for example, web-searching for getting up-to-date information.

Mixture of experts





A popular technique in large, modern chatbots is to have several separate, specialized LLMs, rather than one huge LLM.



How many dots are below?

.....

.....

Wrong

It turns out that LLMs aren't any better at counting than they are at arithmetic. Instead, it has guessed (confidently!).

There are **161** dots in your message.

Tell it to use Python

Use code.

A modern LLM might know its limits and call Python. If not, you can sometimes force it to delegate to Python by saying "use code".

Analyzed

```python
# Count the number of dots in the given string
dots_string = "..."
dot_count = dots_string.count(".")

# Output the result
dot_count
```

Always show details | Copy

Result

177

Copy and pasting is easier than counting

Correct

The total number of dots in your message is **177**.

Luckily, although it can't count, it looks like it can at least paste the correct number of dots into the Python program.

System 1 fast, instinctive, emotional

System 2 slow, deliberative, logical

He talks about these two parts of the brain.

Daniel Kahneman won the Nobel Prize for economics by showing that humans aren't as rational as everyone thought.

One word at a time...



No backtracking

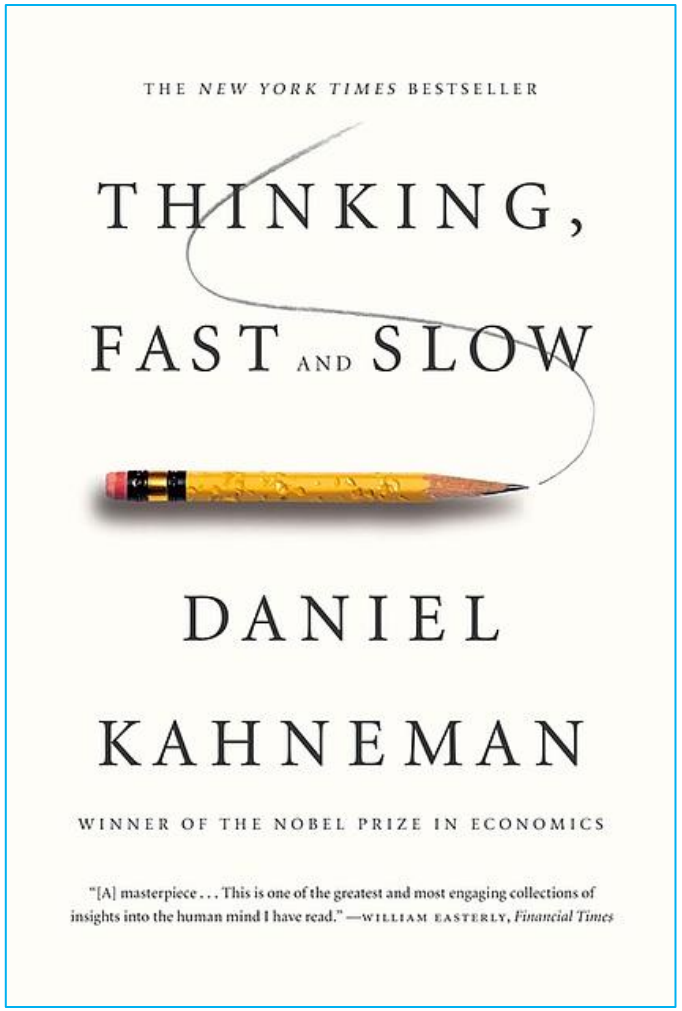
A chatbot charging forward and saying whatever comes into its mind seems like **System 1**.

System 1

If we hear an unexpected word, parts of our brain reacts a fifth of a second.

Thinking, Fast and Slow says unexpected words cause a distinctive pattern in brain activity within a fifth of a second. So, it seems like we are running an LLM in our brains even when we are just listening.

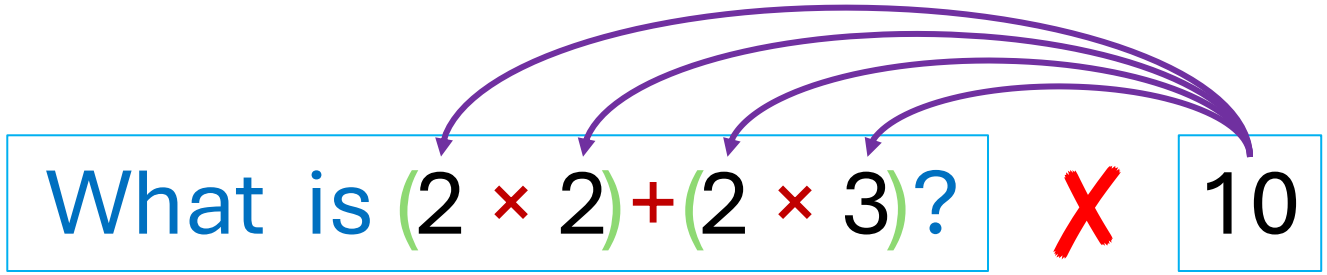
So, the brain is continuously running an LLM!





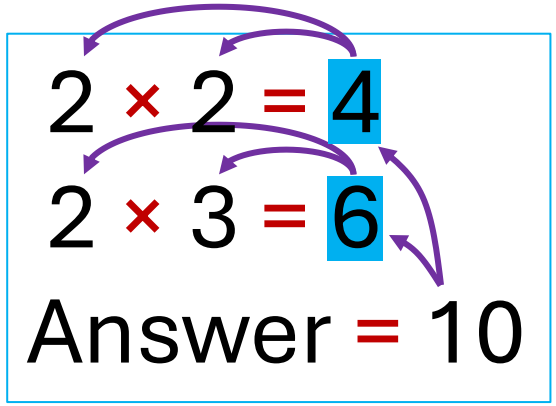
https://research.google/blog/deciphering-language-processing-in-the-human-brain-through-llm-representations/

Arithmetic



A small LLM might struggle with this question because, when emitting the “10”, it is having to account for four things.

What is 2 × 2 + 2 × 3 ? Show your working. ✓



On the other hand, when thinking out loud, it only has to account for two things at a time.

Prompt engineering

=

Creating good prompts

“Think carefully before you answer”

“Let’s think step by step”

Wishful prompting: I must insist, do not hallucinate. My career depends on it, please, please, please.

