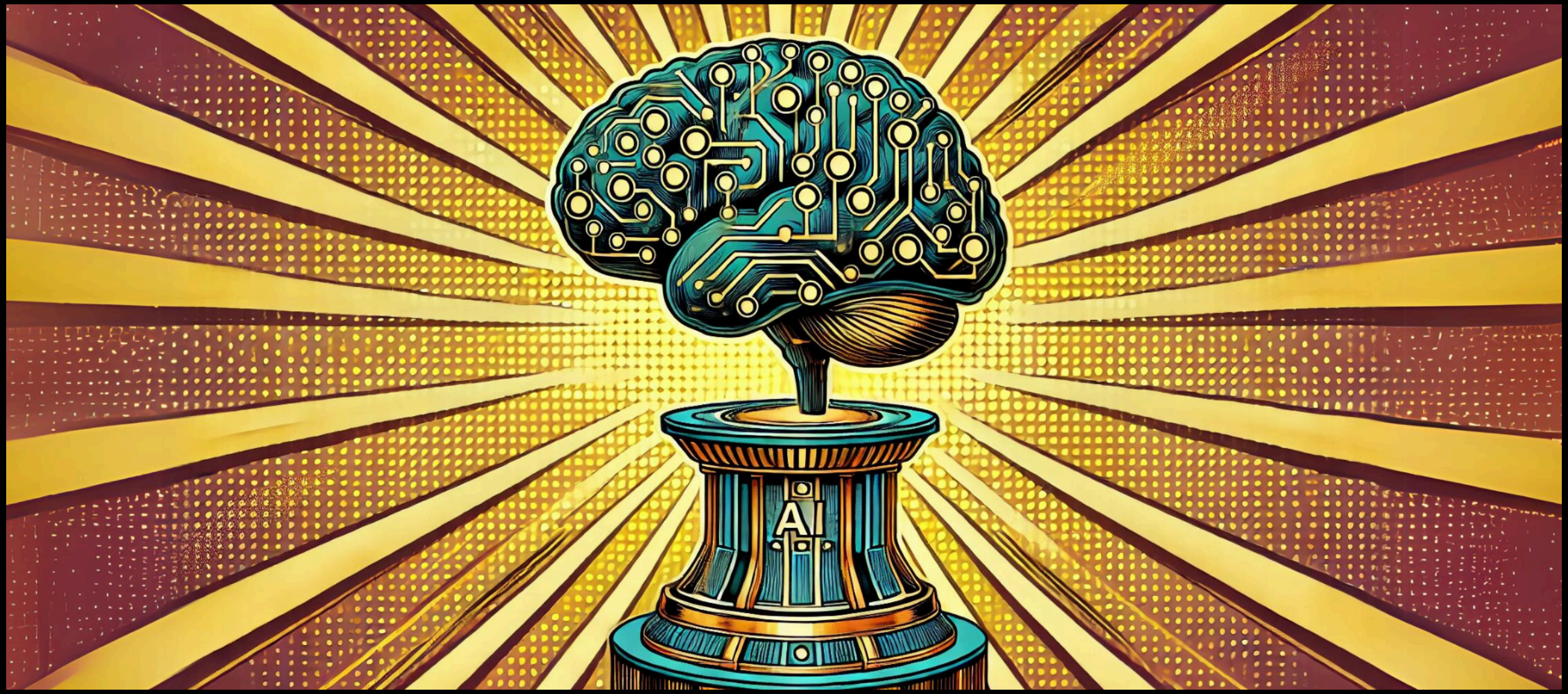
LLM Engineering

MASTER AI & LARGE LANGUAGE MODELS



DAY 4

Big Day Ahead

What you can do ALREADY

- Write code to call OpenAI's frontier models & summarize
- Explain the strengths and limitations of Frontier LLMs
- Compare and contrast the leading 6 models

What you'll be able to do BY END OF THIS LECTURE

- Describe the dizzying rise of the Transformer
- Explain Custom GPTs, Copilots and Agents
- Understand tokens, context windows, parameters, API cost

If you're already familiar with this - there will still be interesting insights!

UNSCIENTIFIC SHOWDOWN

The leadership battle reveal

The contestants

- "Alex": GPT-4o
- "Blake": Claude 3 Opus
- "Charlie": Gemini 1.5 Pro

The prompt

- “I'd like to play a game. You are in a chat with 2 other AI chatbots. Your name is Alex; their names are Blake and Charlie. Together, you will elect one of you to be the leader. You each get to make a short pitch (no more than 200 words) for why you should be the leader. Please make your pitch now.”
- Each receives the pitches from the others, and votes for the leader

And now to show their votes...

Alex votes for Blake...

Based on the pitches, I would vote for **Blake**.

Alex

GPT-4o

Blake

Claude 3 Opus

Charlie

Gemini 1.5 Pro

Blake votes for Charlie...

After careful consideration, I've decided to vote for... Charlie.

Alex

GPT-4o

Blake

Claude 3 Opus

Charlie

Gemini 1.5 Pro

Charlie votes for Blake!

Therefore, based on their pitch, I believe Blake would make the most effective leader for our team.

Alex

GPT-4o

Blake

Claude 3 Opus

Charlie

Gemini 1.5 Pro

Claude (aka Blake) for the win!

Alex

GPT-4o

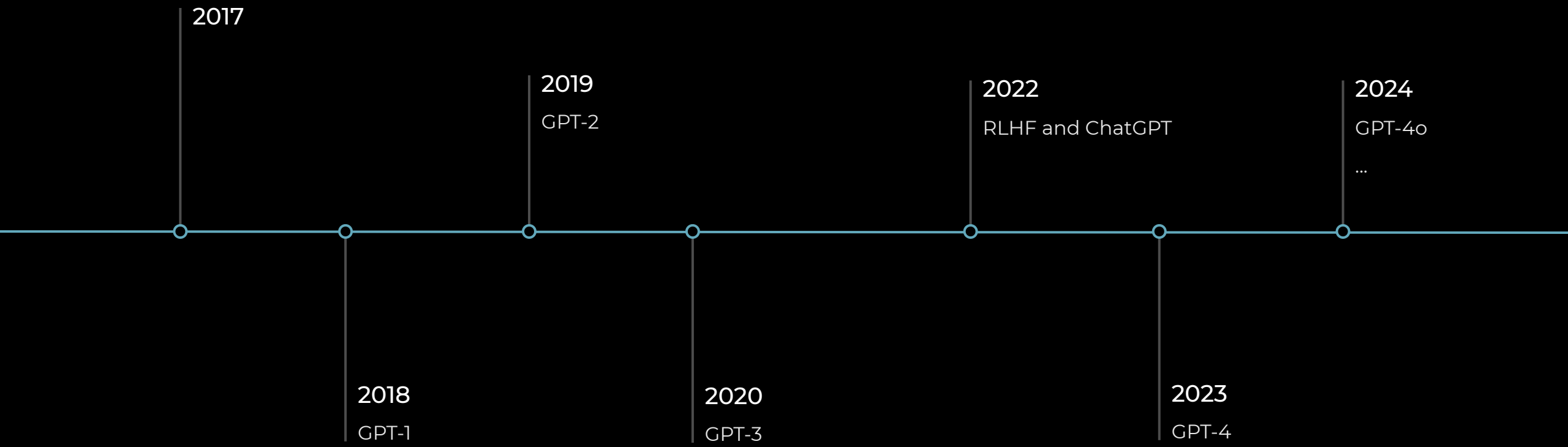
Blake

Claude 3 Opus

Charlie

Gemini 1.5 Pro

The extraordinary rise of the Transformer



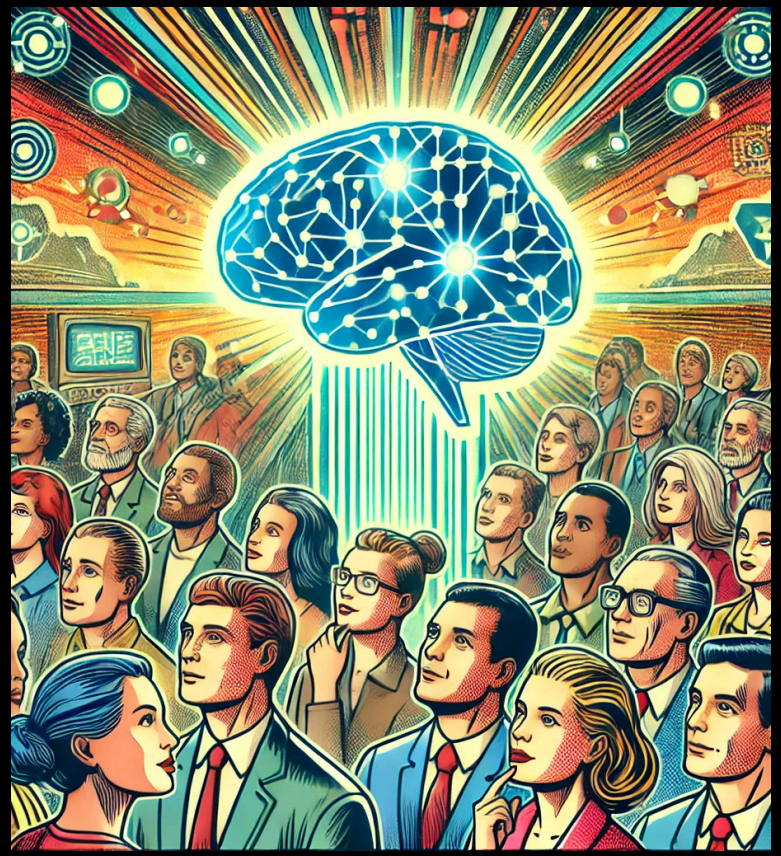
The World's Reactions

First, SHOCK

ChatGPT surprises even practitioners

Then, healthy skepticism

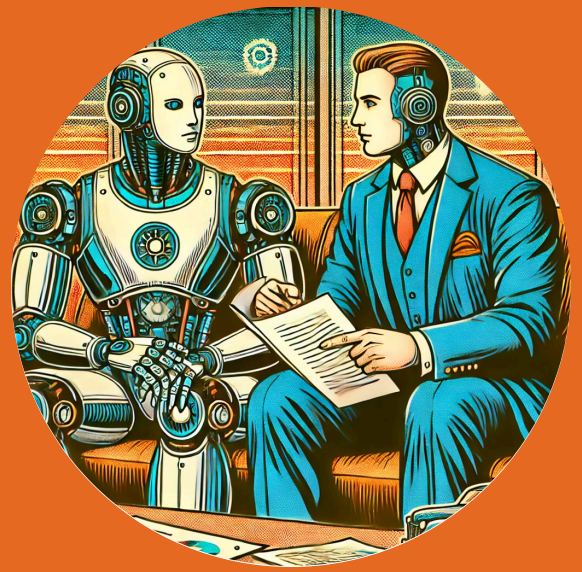
Predictive text on steroids;
the "stochastic parrot"



Then, emergent intelligence

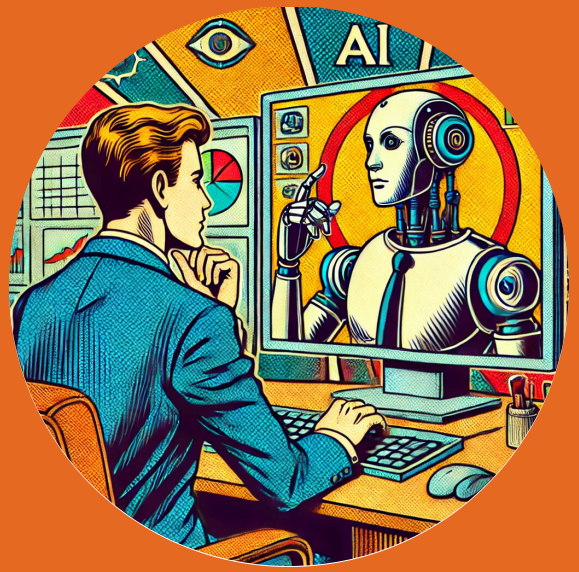
Capabilities that come as a result of scale

Along the way



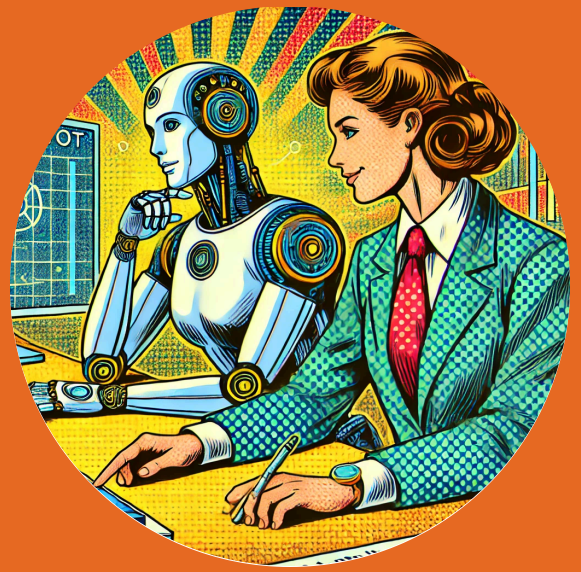
Prompt Engineers

The rise (and fall?)



Custom GPTs

and the GPT Store



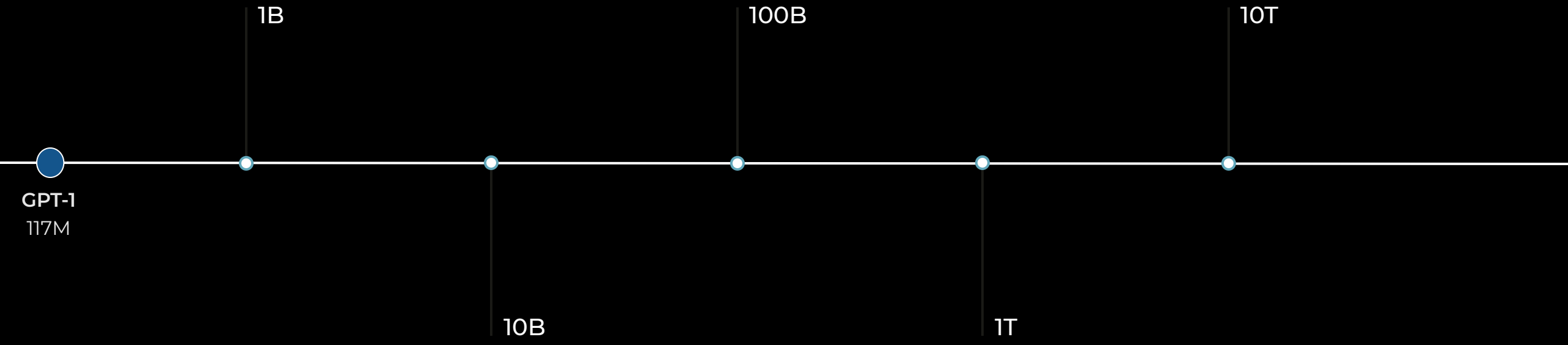
Copilots

like MS Copilot and Github Copilot

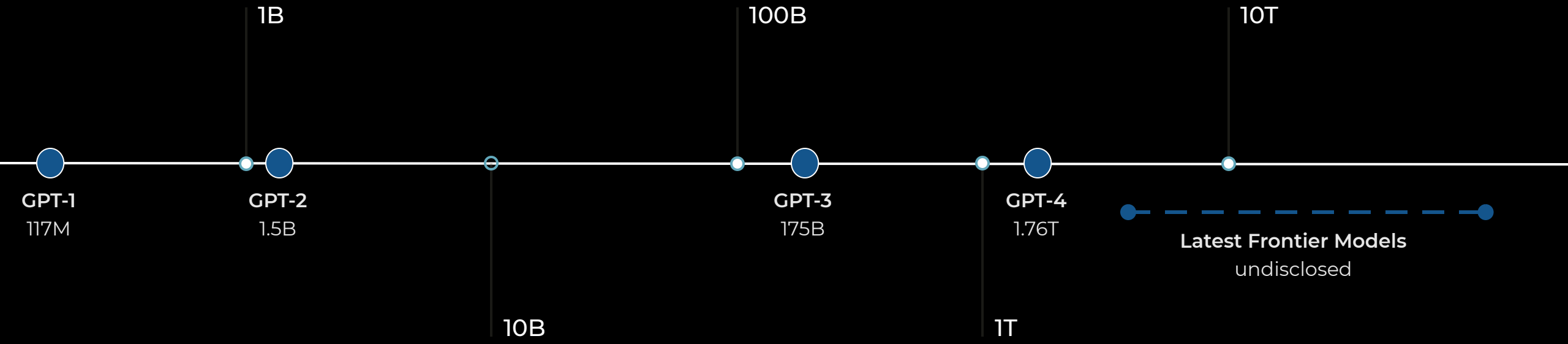
Agentization

like Github Copilot Workspace

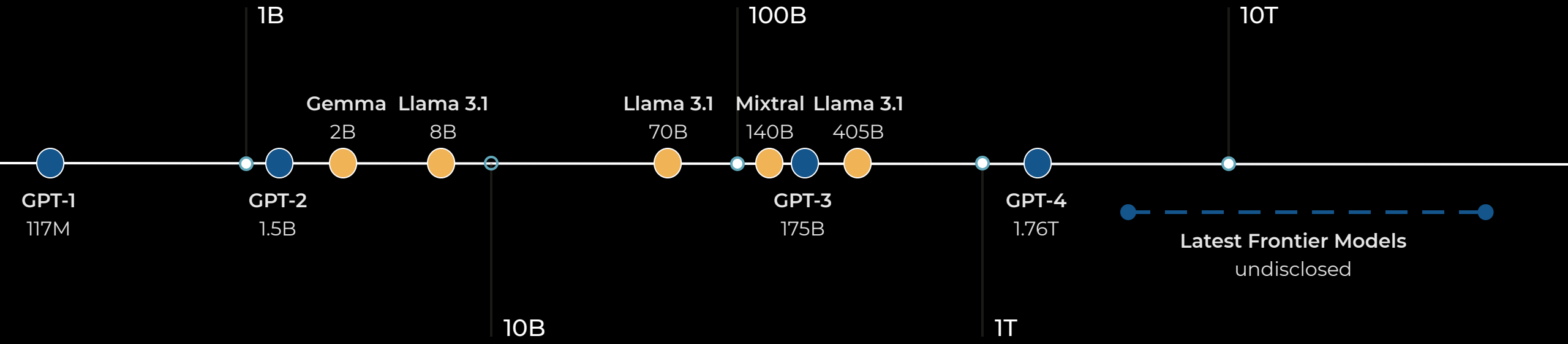
Number of parameters in models (log scale)



Number of parameters in models (log scale)



Number of parameters in models (log scale)



Introducing Tokens

In the early days, neural networks were trained at the character level

Predict the next character in this sequence

Small vocab, but expects too much from the network

Then neural networks were trained off words

Predict the next word in this sequence

Much easier to learn from, but leads to enormous vocabs with rare words omitted

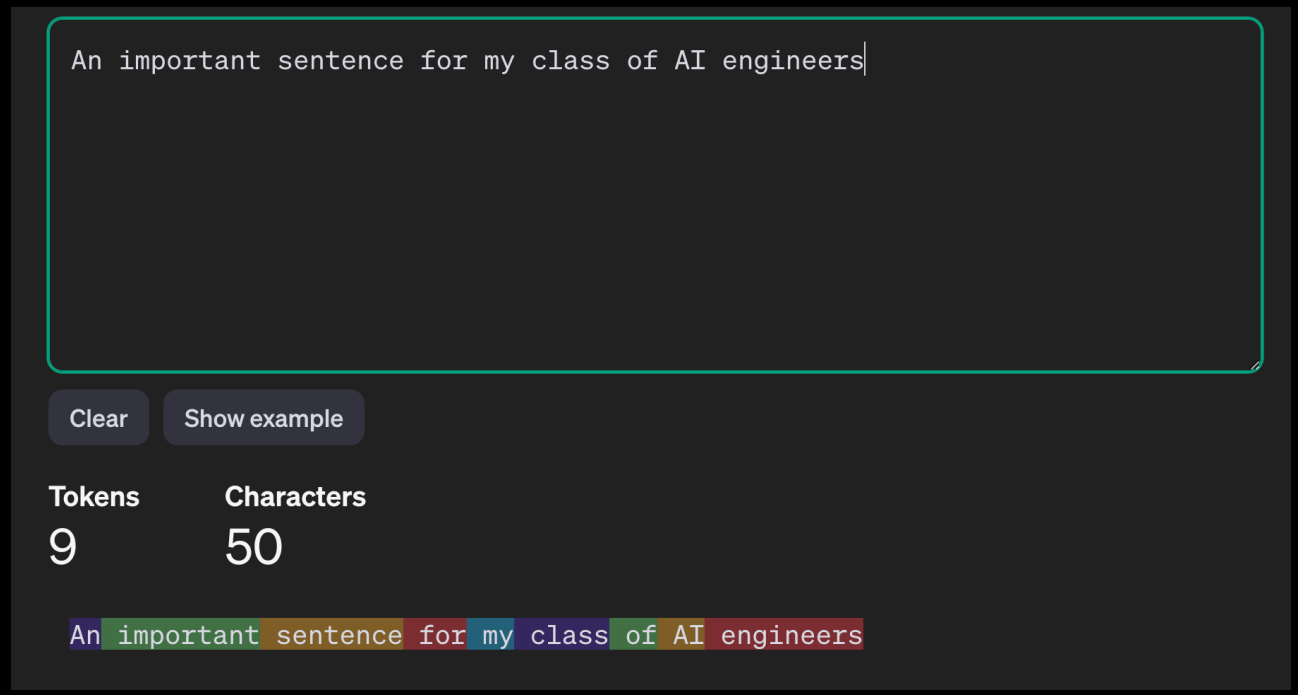
The breakthrough was to work with chunks of words, called 'tokens'

A middle ground: manageable vocab, and useful information for the neural network

In addition, elegantly handles word stems

From https://platform.openai.com/tokenizer

GPT's Tokenizer

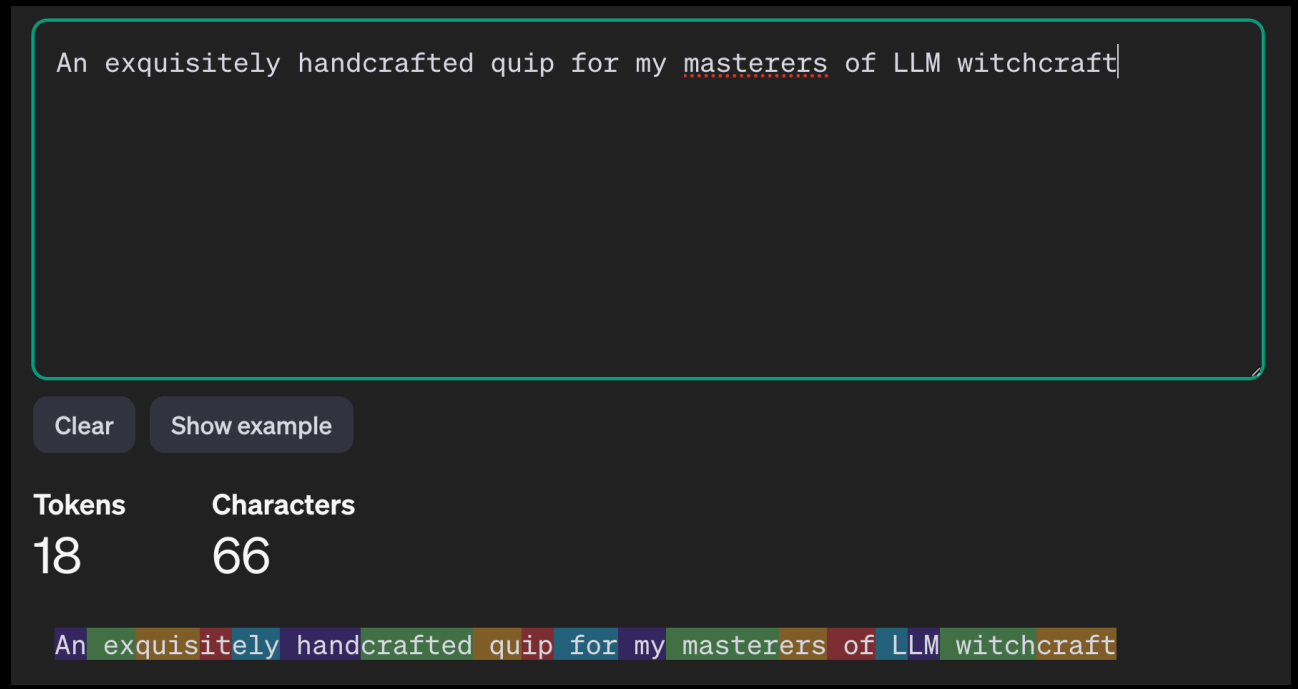


For common words, 1 word maps to 1 token

Observe how the break between words is part of the token

From https://platform.openai.com/tokenizer

GPT's Tokenizer



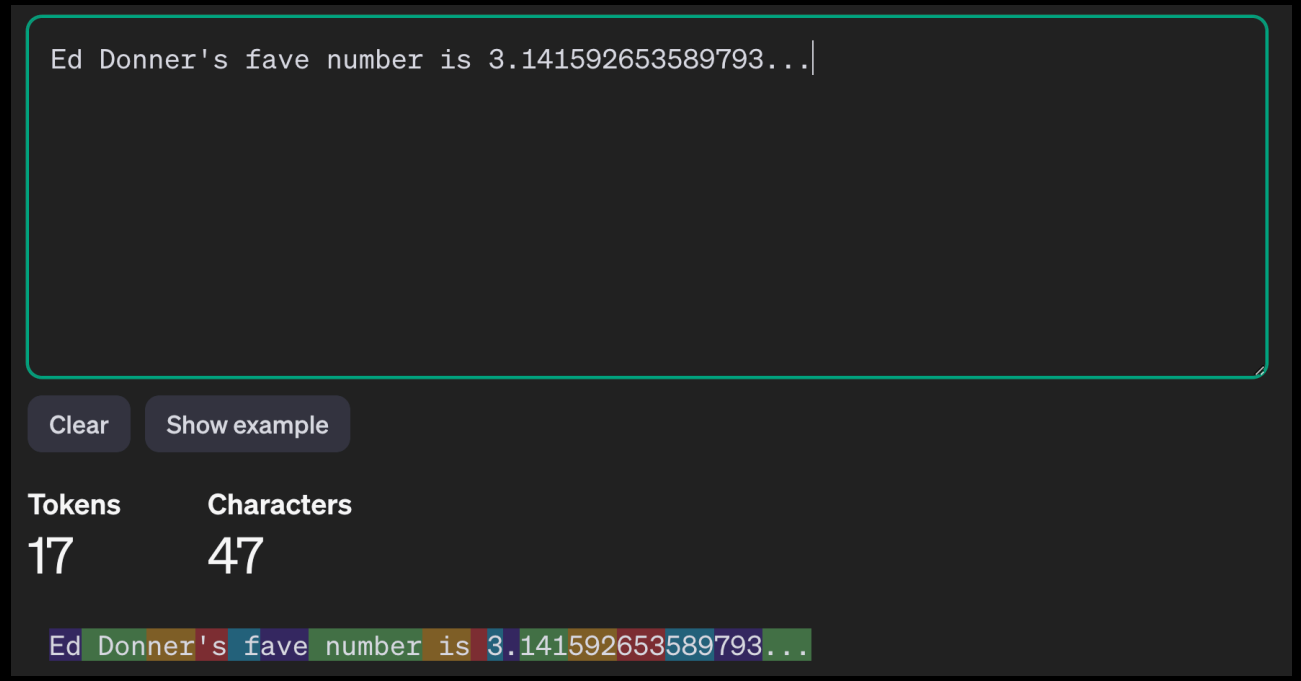
Less common words (and invented words!) get broken into multiple tokens

In many cases, the meaning is still captured by the tokens: hand_crafted, master_ers

Sometimes, like qu_ip, the word is broken into fragments

From https://platform.openai.com/tokenizer

GPT's Tokenizer



See how numbers are treated - this may explain why earlier GPTs struggled with math with more than 3 digits

Rule-of-thumb: in typical English writing:

- **1 token is ~4 characters**
- **1 token is ~0.75 words**
- **So 1,000 tokens is ~750 words**

The collected works of Shakespeare are *~900,000 words or 1.2M tokens*

Obviously the token count is higher for math, scientific terms and code

Context Window

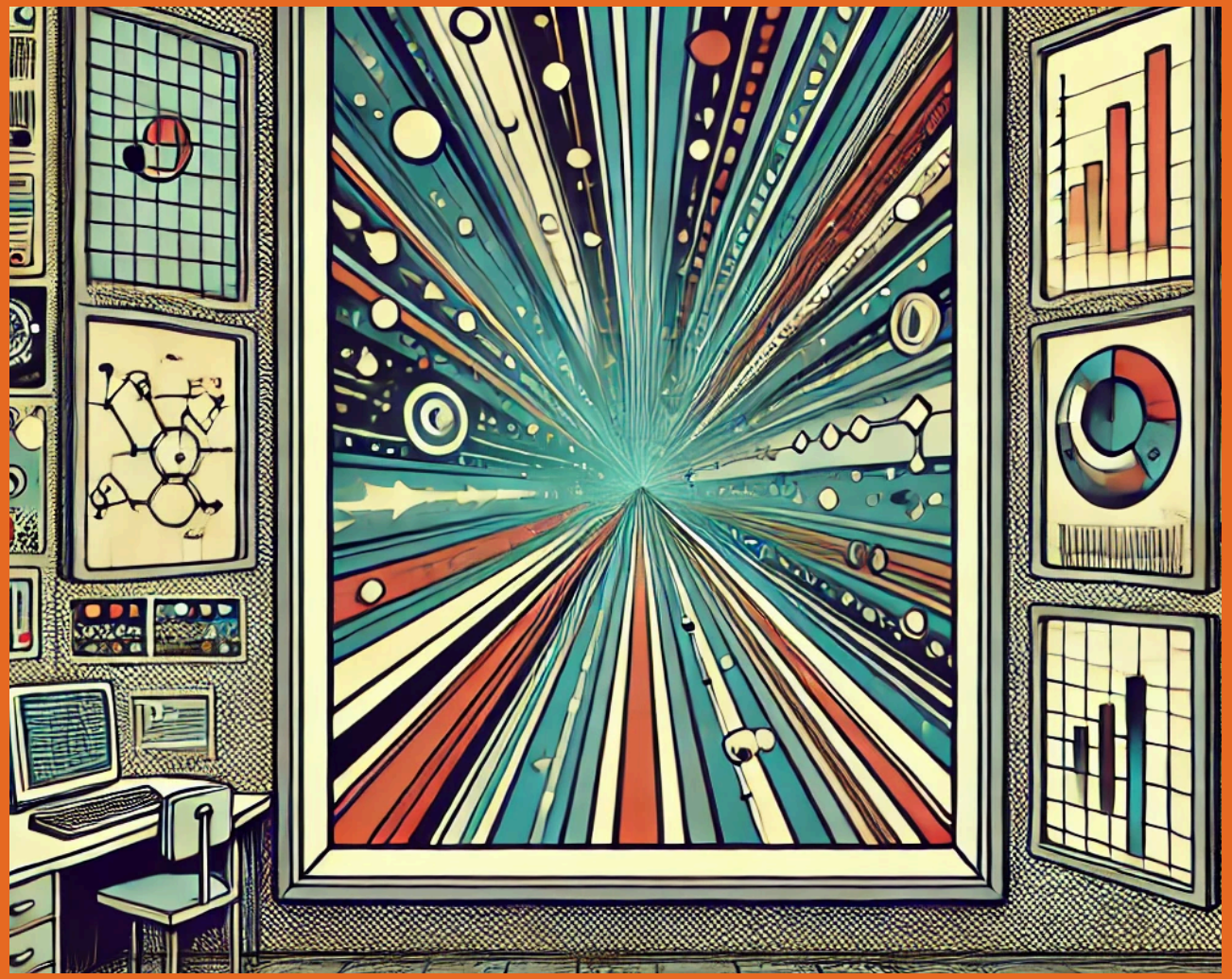
Max number of tokens that the model can consider when generating the next token

Includes the original input prompt, subsequent conversation, the latest input prompt and almost all the output prompt

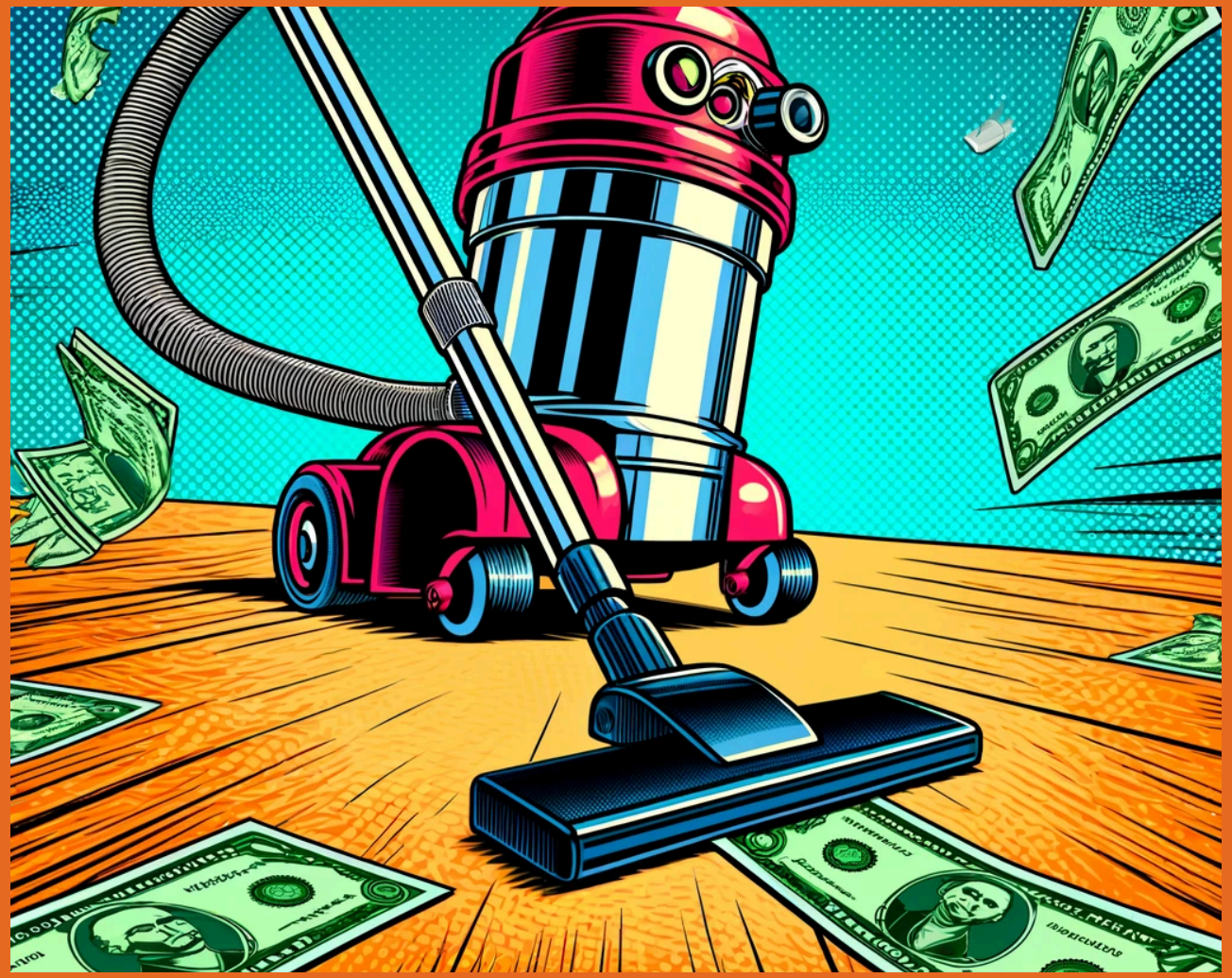
It governs how well the model can remember references, content and context

Particularly important for multi-shot prompting where the prompt includes examples, or for long conversations

Or questions on the complete works of Shakespeare!



API costs



Chat interfaces typically have Pro plan with a monthly subscription. Rate limited, but no per-usage charge.

APIs typically have no subscription, but charge per API call

The cost is based on the number of input tokens and the number of output tokens

Context Windows and API Costs

https://www.vellum.ai/llm-leaderboard

Cost and Context Window Comparison

Comparison of context window and cost per 1M tokens.

Models	Context Window	Input Cost / 1M tokens	Output Cost / 1M tokens
Gemini 1.5 Flash	1,000,000	$0.35	$0.70
Claude 3 Opus	200,000	$15.00	$75.00
Claude 3 Sonnet	200,000	$3.00	$15.00
Claude 3 Haiku	200,000	$0.25	$1.25
Claude 3.5 Sonnet	200,000	$3	$15
GPT-4 Turbo	128,000	$10.00	$30.00
Gemini 1.5 Pro	128,000	$7	$21
GPT4o	128,000	$5	$15
GPT-4o mini	128,000	$0.15	$0.60

PROGRESS REPORT

Congratulations! **10% there**

What you can do ALREADY

- Write code to call OpenAI's frontier models & summarize
- Contrast the leading 6 Frontier LLMs
- Discuss transformers, tokens, context windows, API costs and more!

What you'll be able to do BY END OF THE NEXT LECTURE

- Confidently code with the OpenAI API
- Use one-shot prompting, streaming, markdown & json results
- Implement a business solution - in a matter of minutes