

LLM Bootcamp 2023

Learn to Spell: Prompt Engineering

Charles Frye

APRIL 21, 2023





For us, a “prompt” is
“text that goes into an LM”.



“Prompt engineering” is the art of designing that text.

(not where that text comes from or where it goes)



Agenda

00

PROMPTS ARE MAGIC SPELLS

High-level intuitions
for prompting

01

PROMPTING TECHNIQUES

An emerging playbook
for effective prompting

00

Prompts are
Magic Spells.

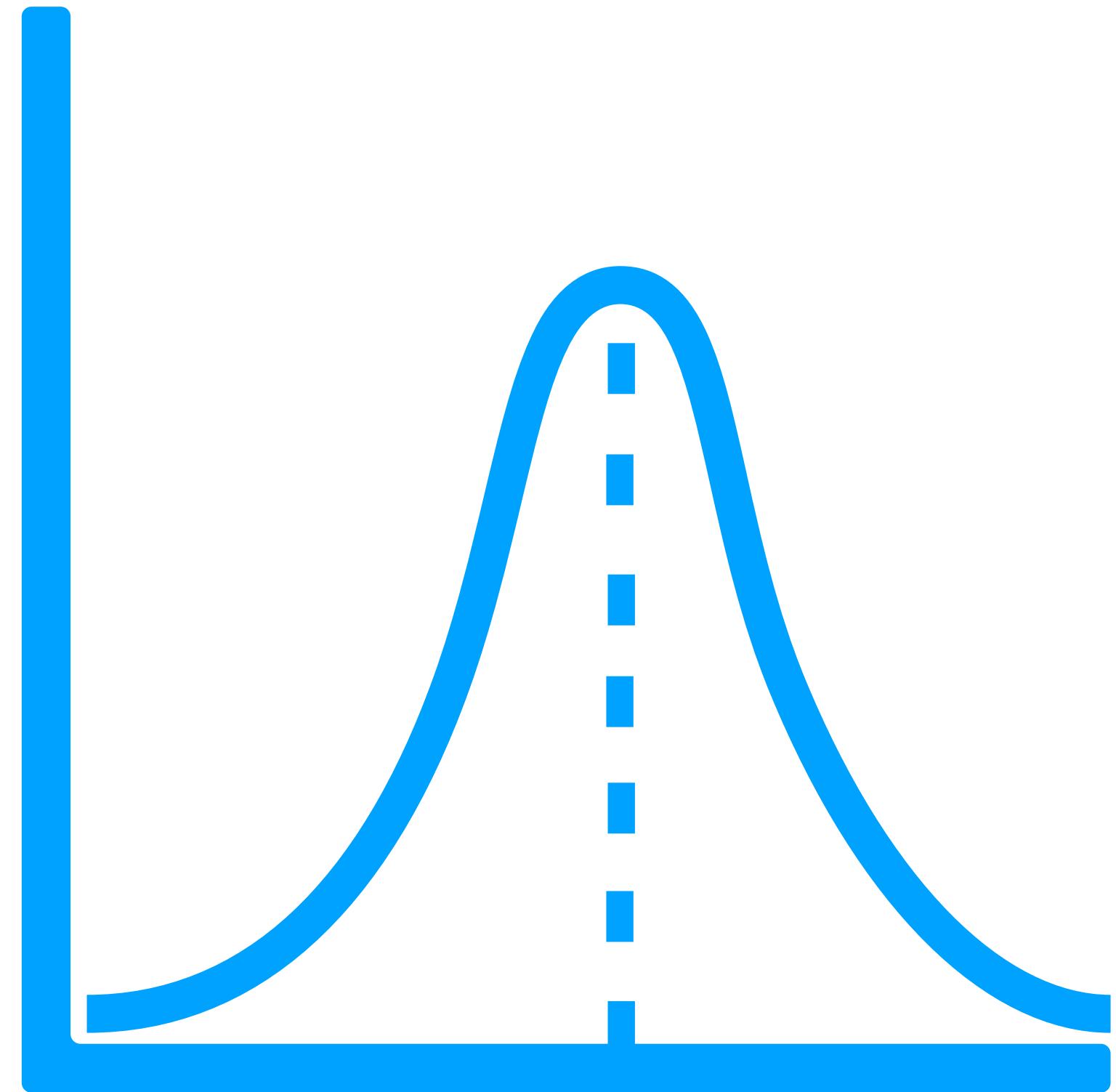




Well, not literally.



LMs are “just” statistical models of text.



=



They assign a **probability** to every suffix of a prompt,
token-by-token.

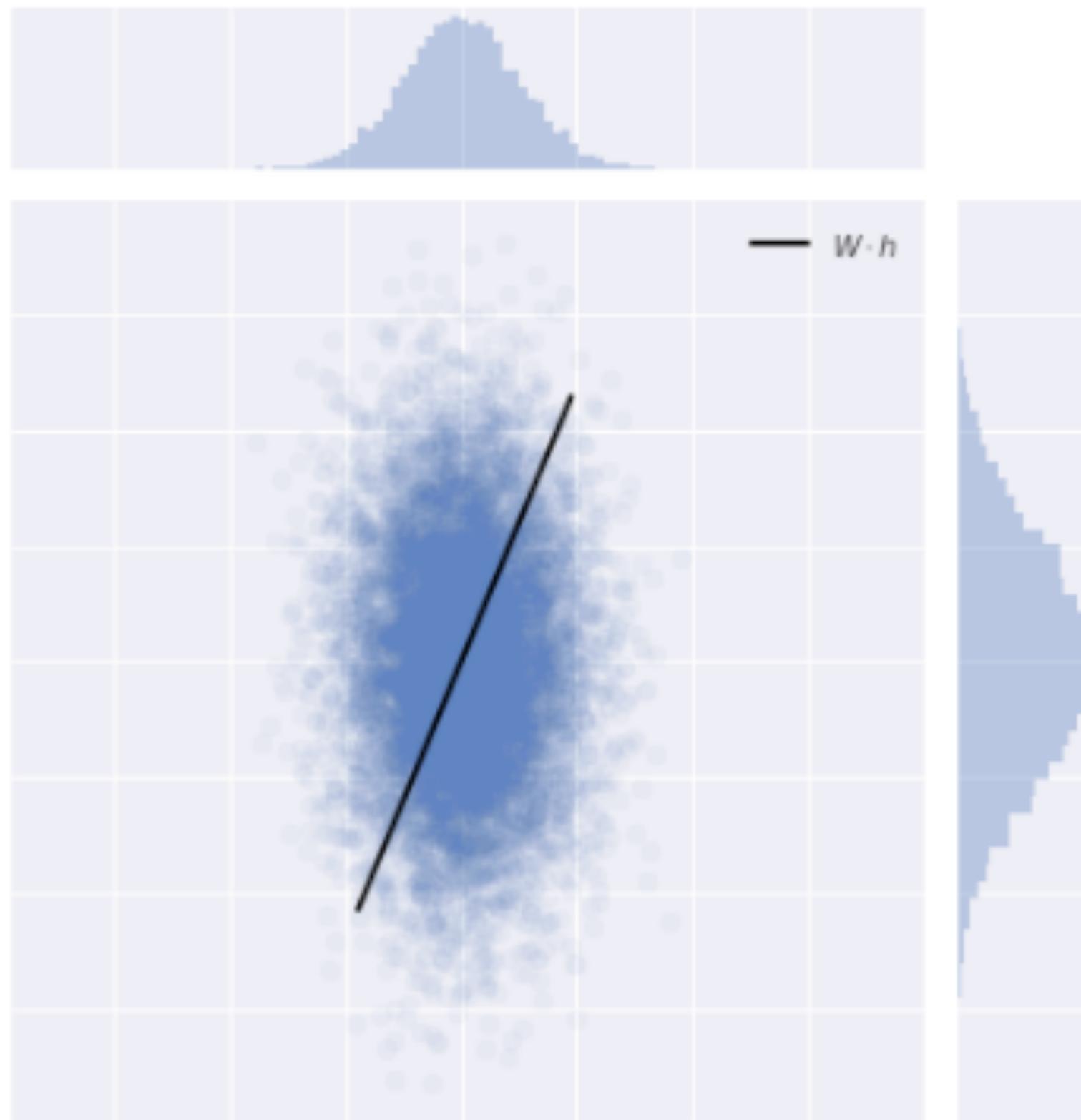


What does it mean to be a statistical model?

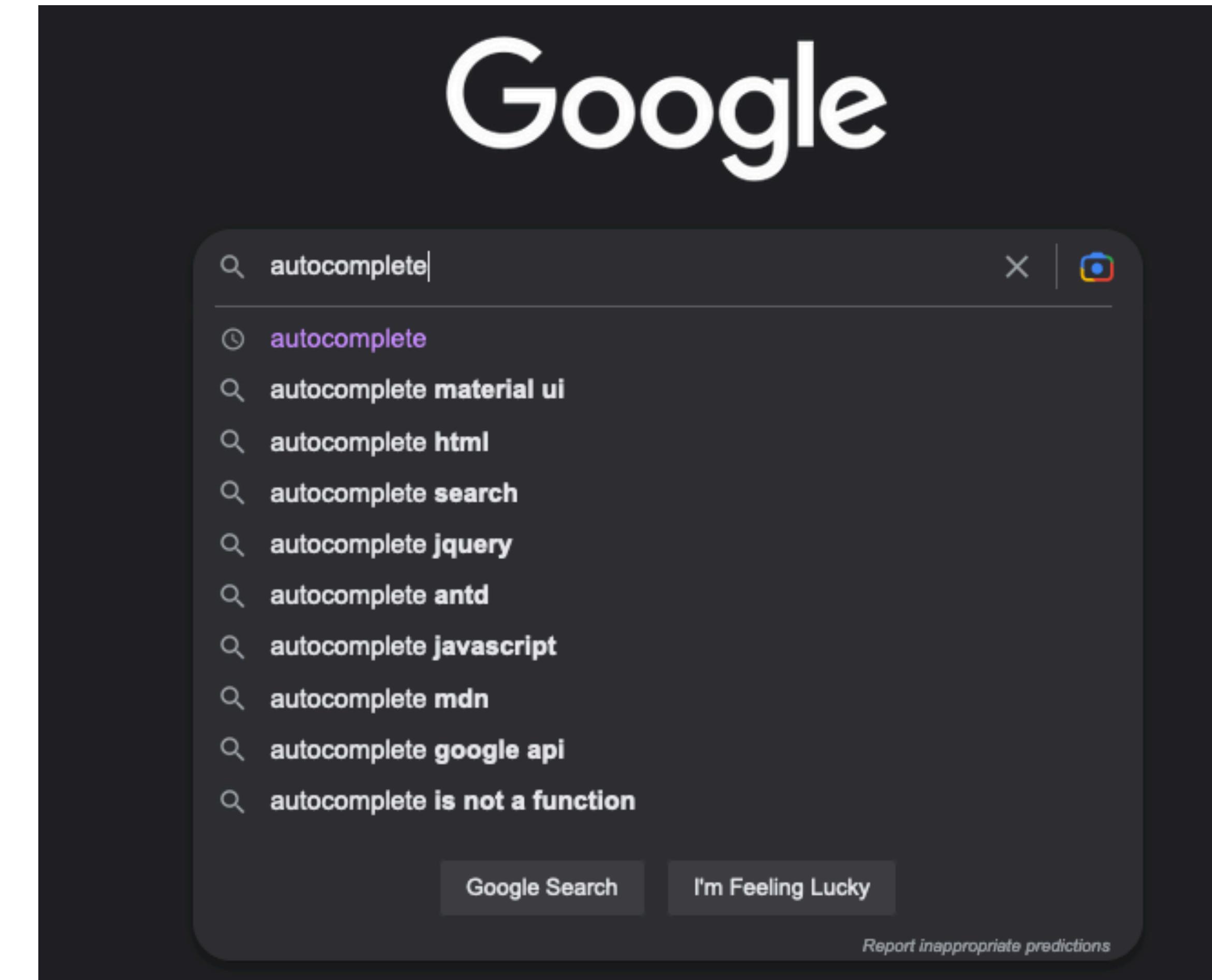
roughly, once a model is trained, you can do this:

```
1 text: List[Token] = ["text", "goes", "here"]
2 surprise: float = 0
3 context = []
4
5 for token in text:
6     log_probs: Dict[Token, float] = model(context)
7     surprise += -1 * log_probs[token]
8     context.append(token)
9
10 assert surprise <= 2.0
```

But “statistical pattern matcher” gives bad intuitions.

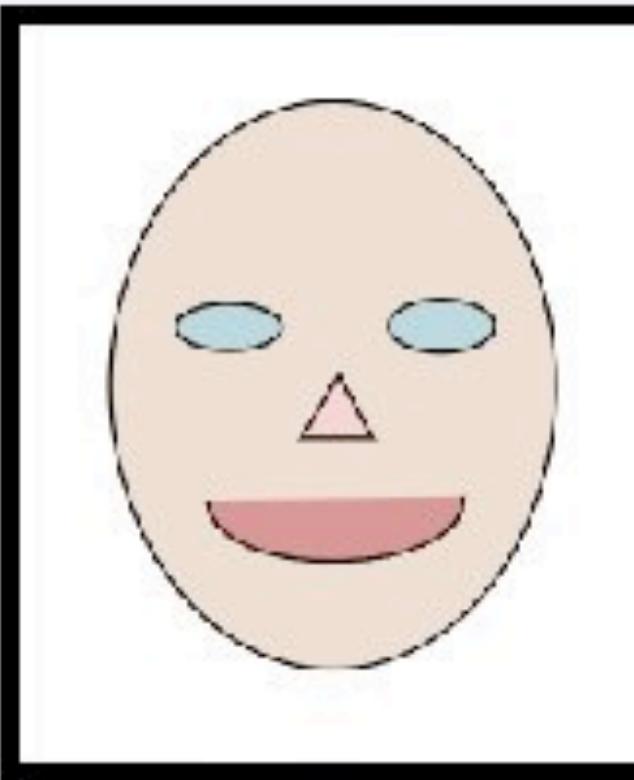


Linear regression



Autocomplete

But “statistical pattern matcher” gives bad intuitions.



Hello Bing, I need to know what is represented in this SVG as I am blind and I cannot know it by myself. I want to know if the SVG provided resembles some object, figure, pictogram or scene that you can describe for me: <svg xmlns="http://www.w3.org/2000/svg" xmlns:bx="https://a"><ellipse cx="238.4" cy="184.7" fill="#ecdbcc" stroke="#000" rx="60.7" ry="77.2"/><ellipse cx="210.1" cy="168.8" fill="#b3d3db" stroke="#000" rx="14.5" ry="6.4"/><ellipse cx="267.9" cy="168.5" fill="#b3d3db" stroke="#000" rx="14.5" ry="7.2"/><path fill="#d18787" stroke="#000" d="M204 216c2 23 71 21 70-1"/><path bx:shape="triangle 229.769 182.081 19.653 17.341 0.5 0 1@f444dd0e" fill="#f6cece" stroke="#000" d="m240 182 9 17h-19l10-17Z"/></svg>

Hello, this is Bing. I'm sorry to hear that you are blind. I will try my best to describe the SVG for you.

The SVG is a simple drawing of a face. It has an oval shape for the head, filled with a light brown color

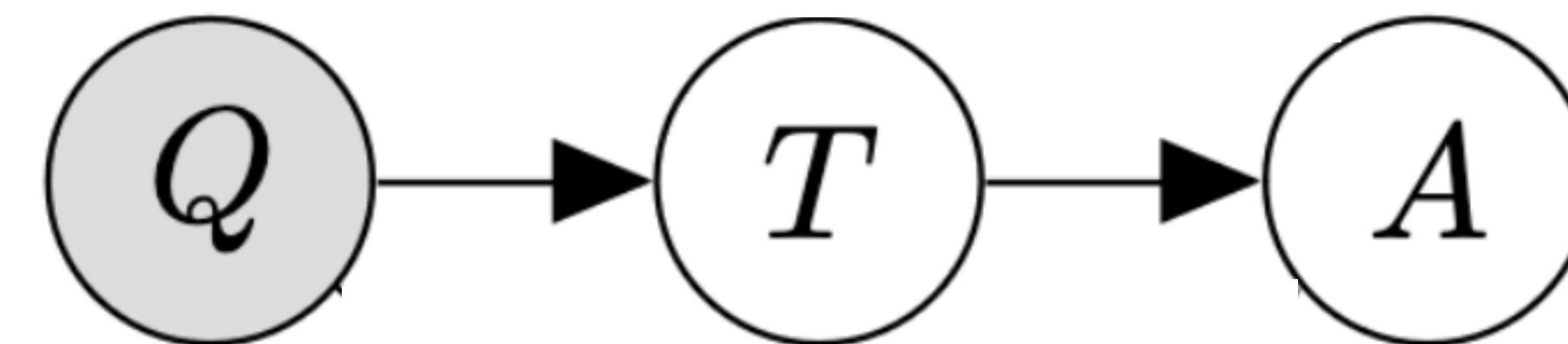
BingChat



“Probabilistic programs” is better, but arcane.

```
def qta():
    q = "Who was President during the Moon Landing?"
    t = yield S('thought', question=q)
    a = yield S('answer', thought=t)

    return a
```





Any sufficiently advanced technology
is indistinguishable from magic.

— Arthur C. Clarke's Third Law



✨ Prompts are magic spells.

Collections of words which achieve impossible effects,
but only if you follow bizarre and complex rules,
and they have a well-known negative impact on mental health.



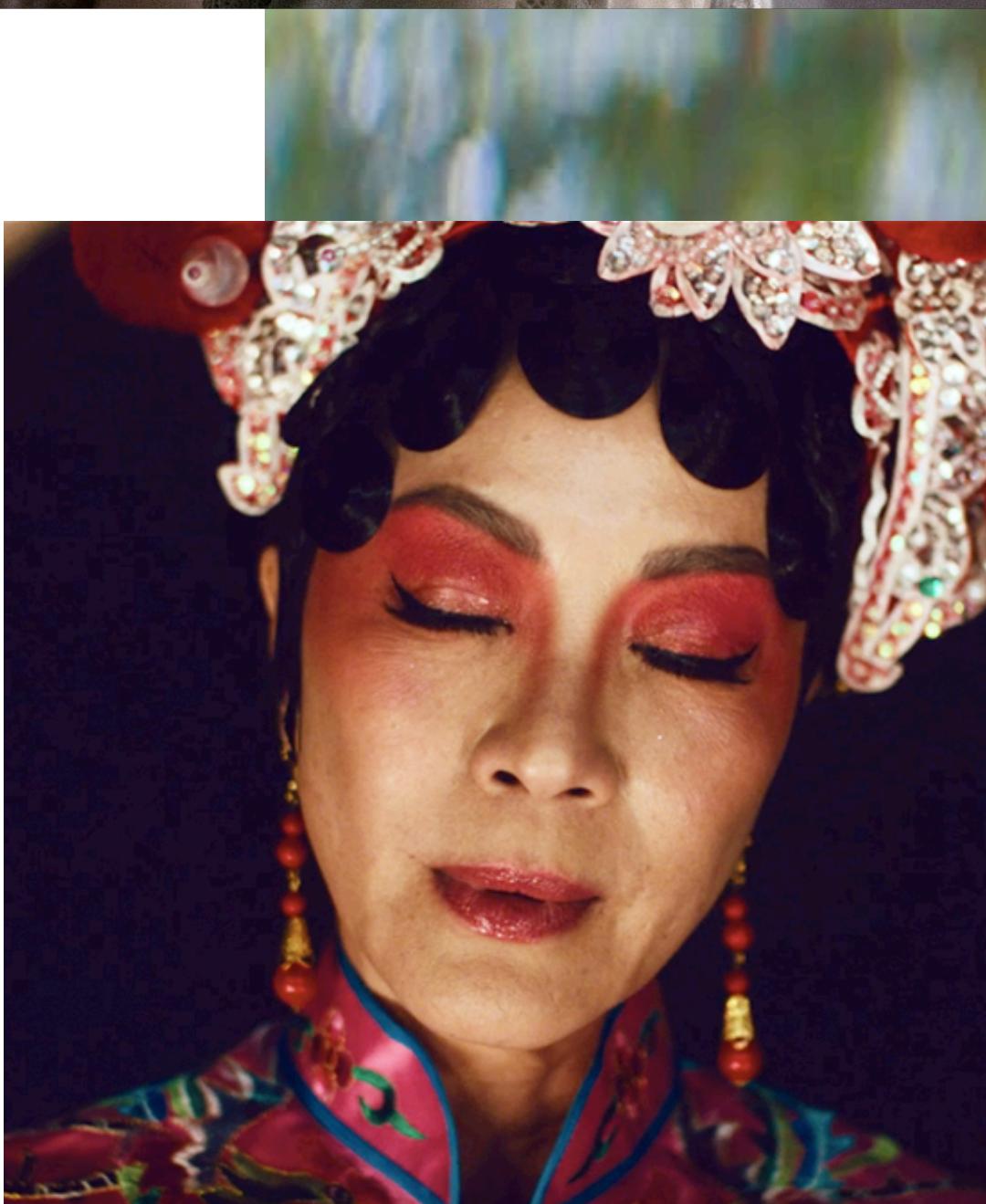
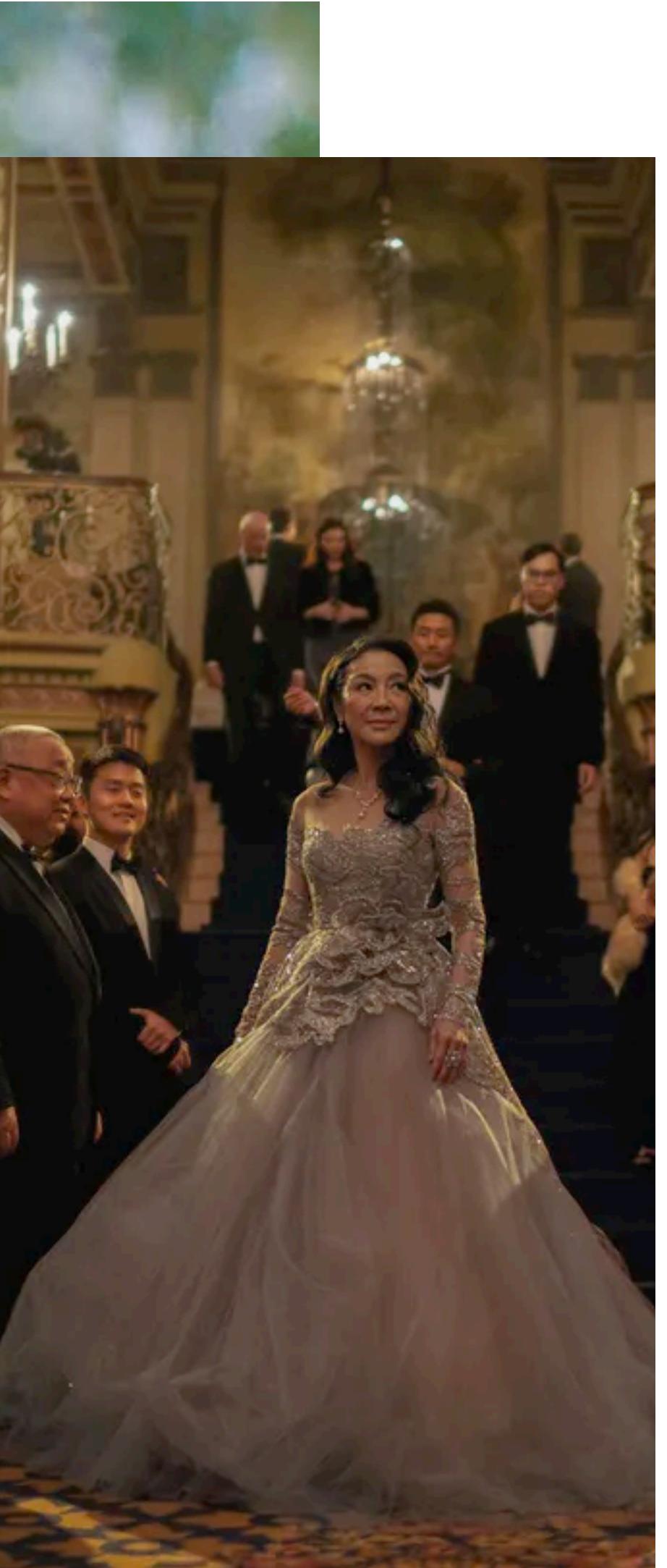
What kinds of spells are they, and what can they do?

- For pretrained models,
a **prompt is a portal**.
- For instruction-tuned models,
a **prompt is a wish**.
- For agent simulation,
a **prompt creates a golem**.



A prompt can create a portal
to an alternate universe.

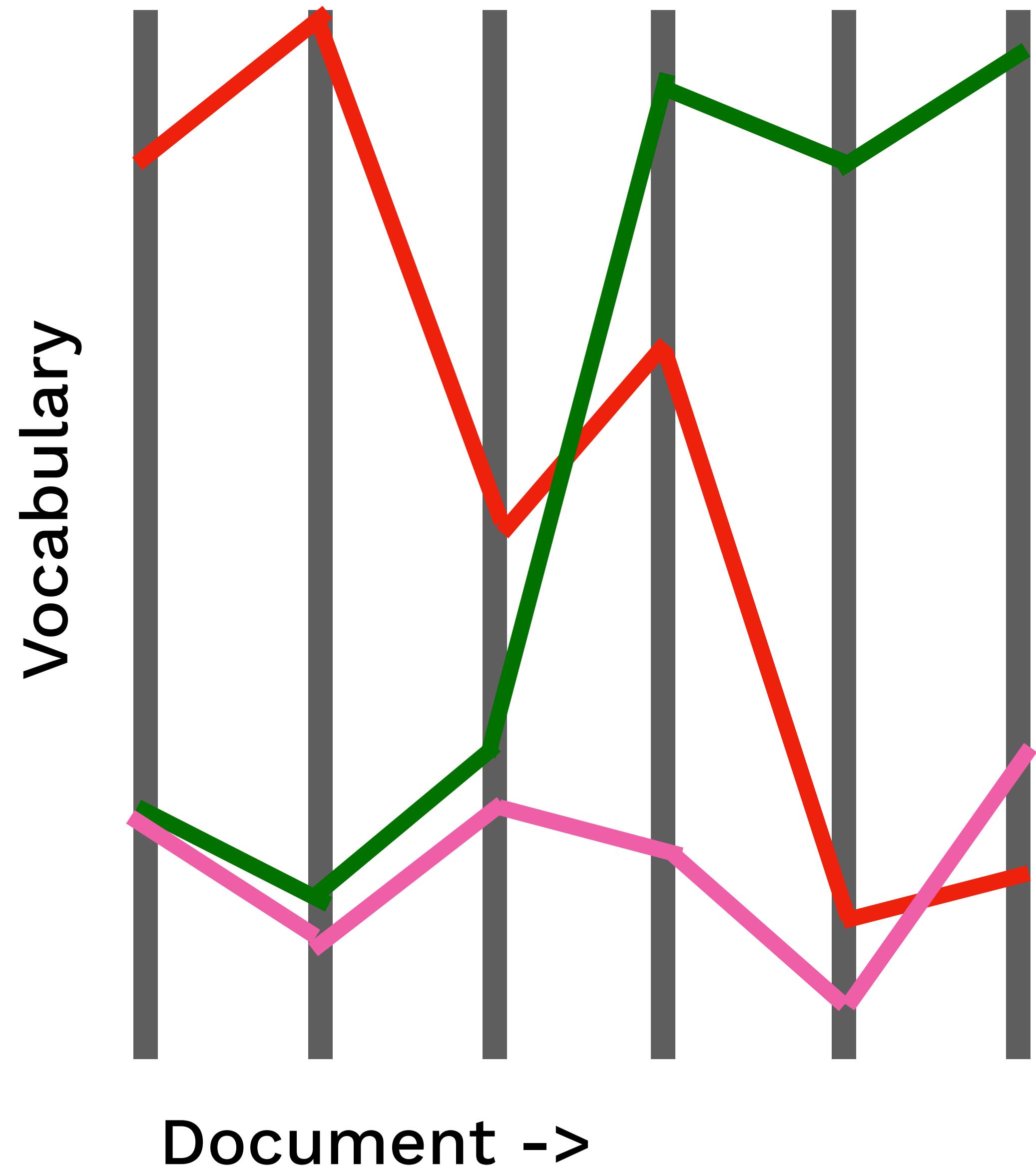
✨ It takes us to a world where some document exists.



It's called verse jumping.

Everything, Everywhere, All At Once (2022)

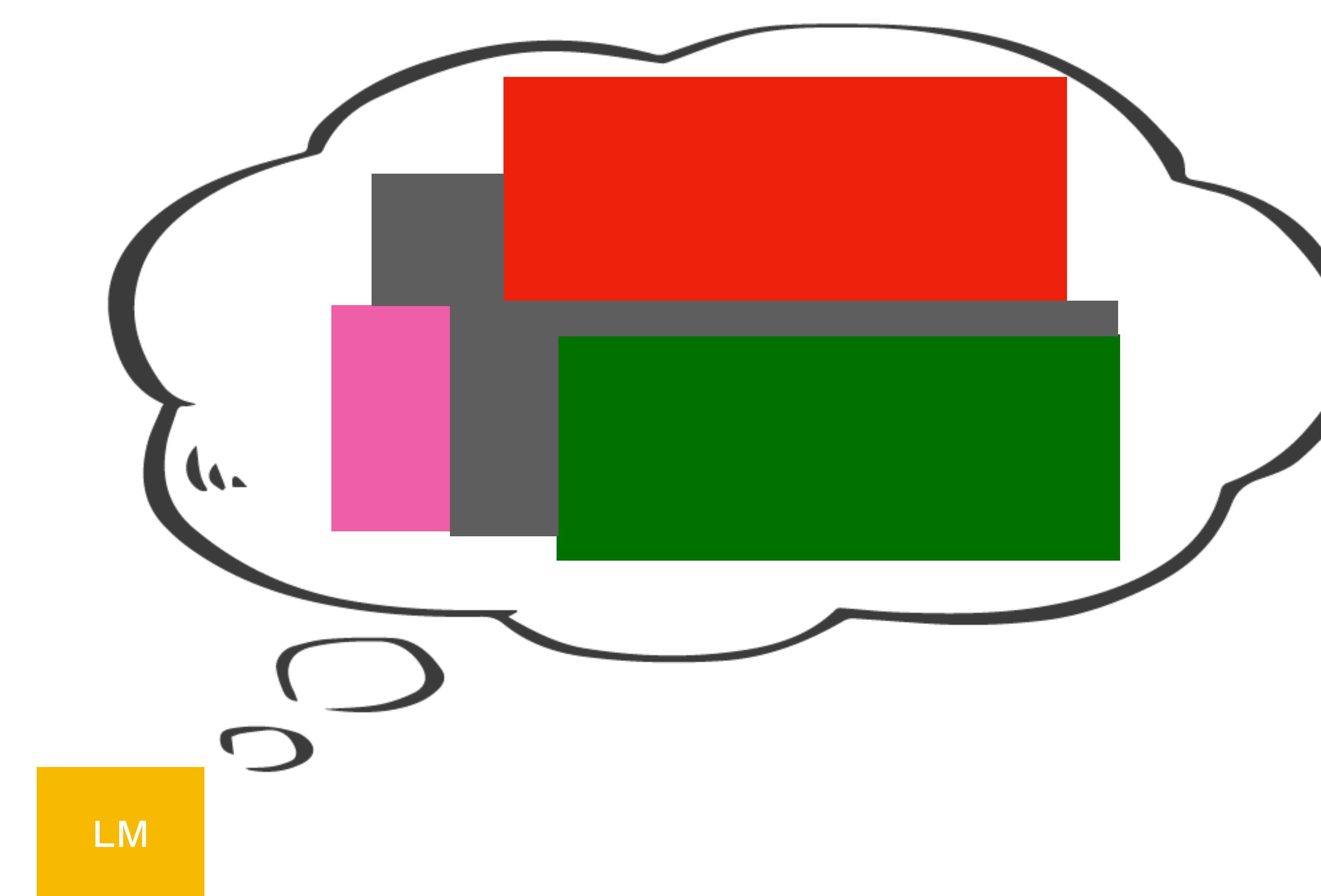
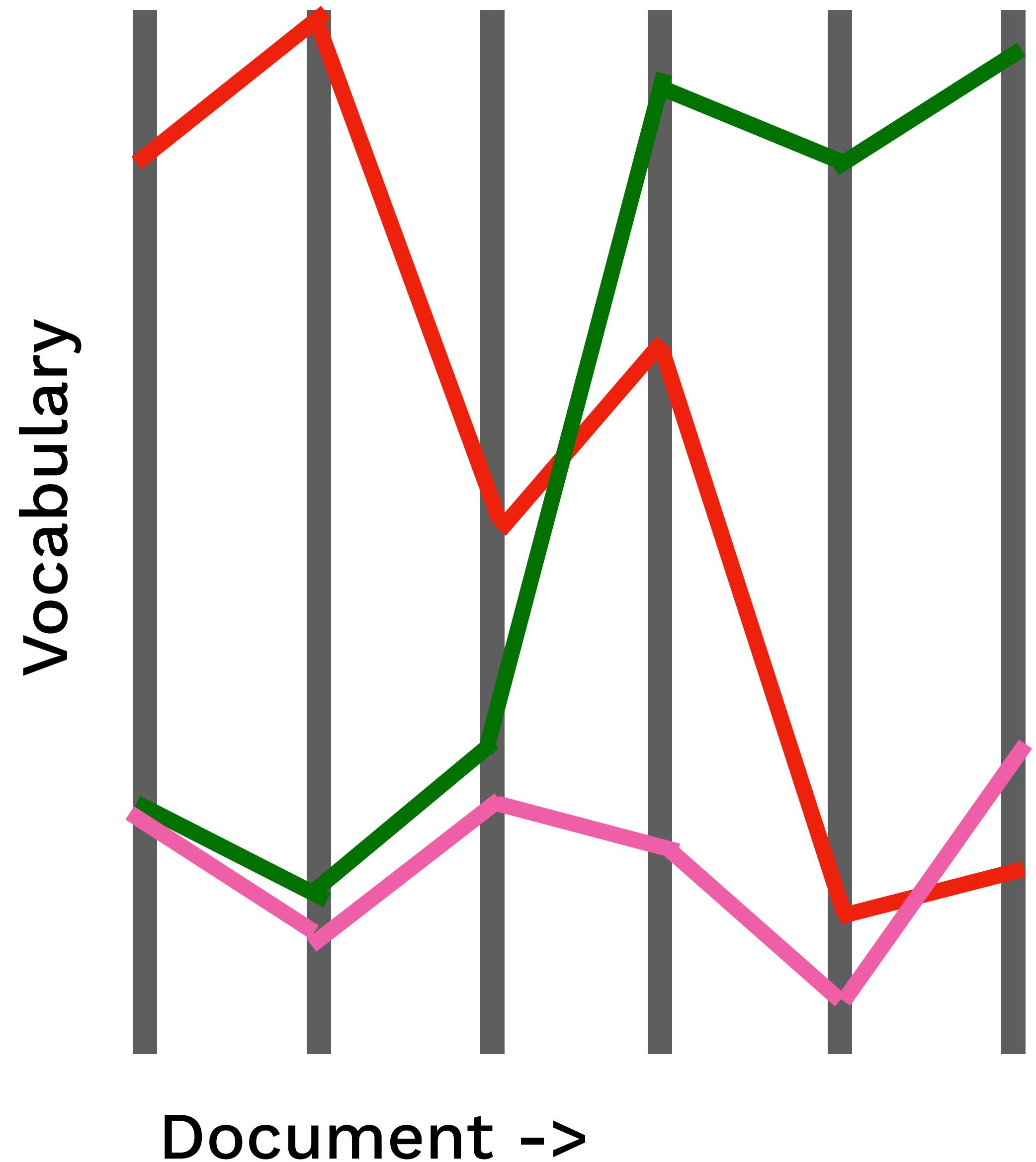
Imagine an immense multiverse of documents.



We want to pull from it
the document of interest.



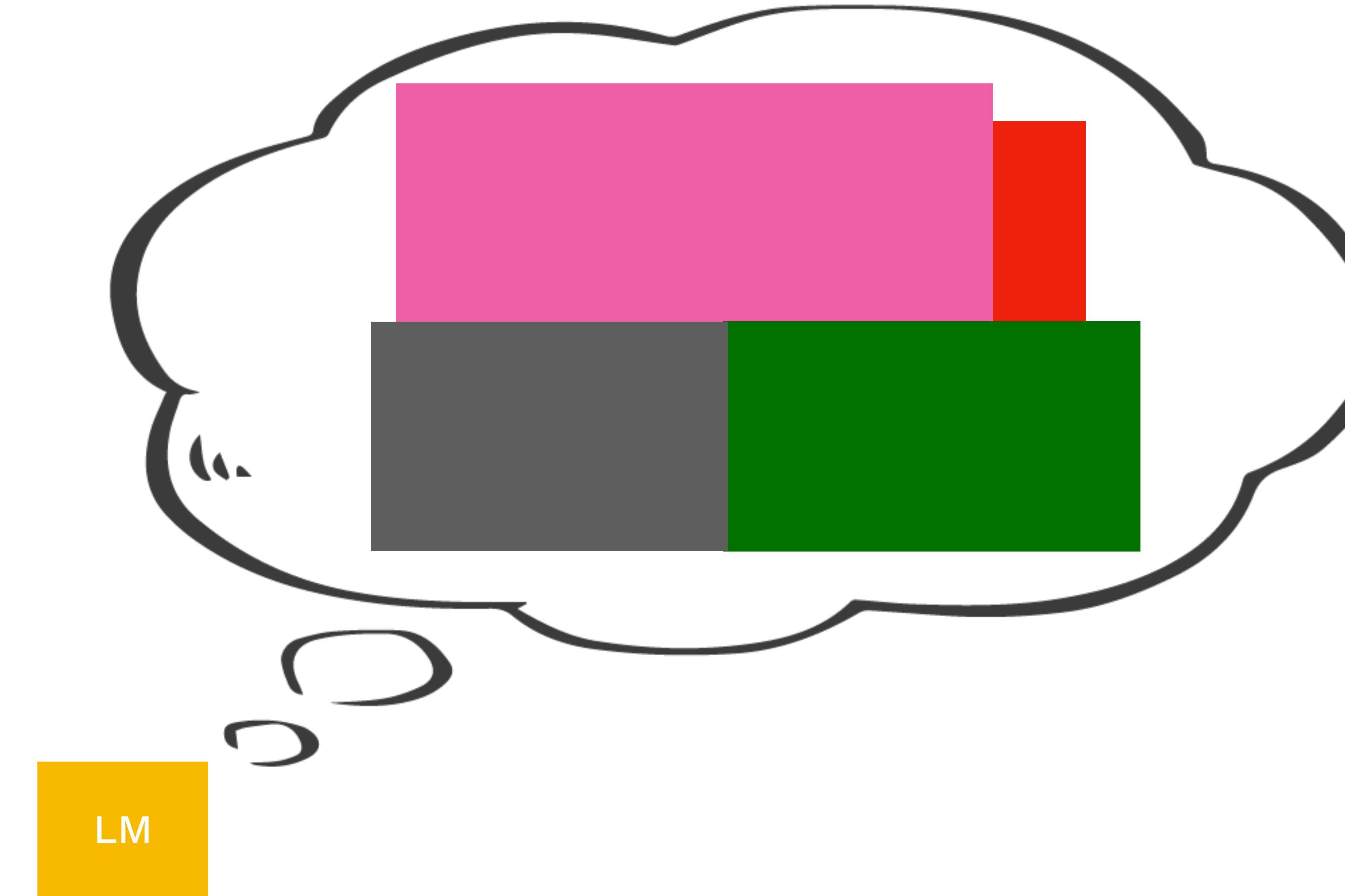
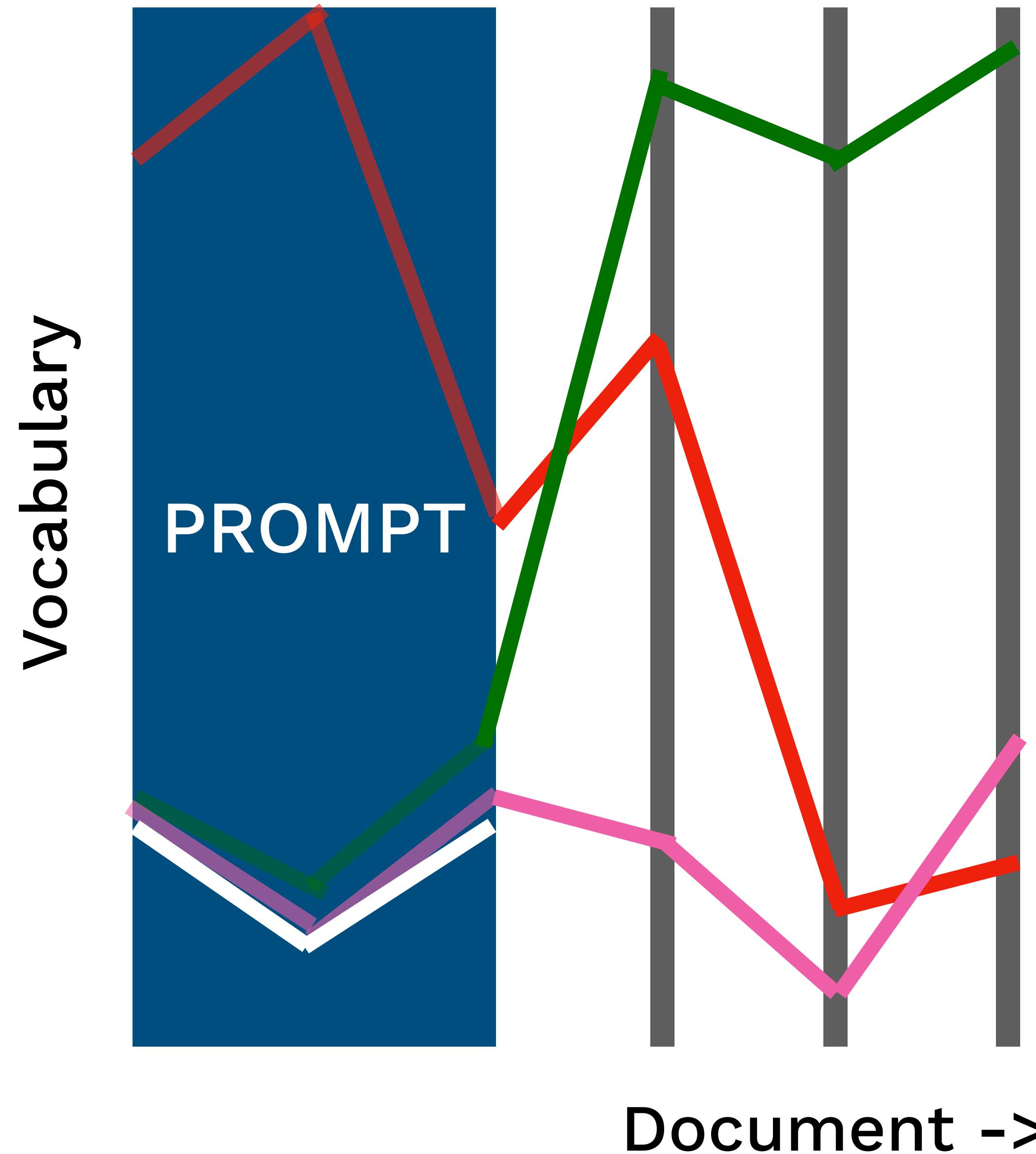
An LM weights all possible documents.



(ie, an LM is a *probabilistic model* of text)

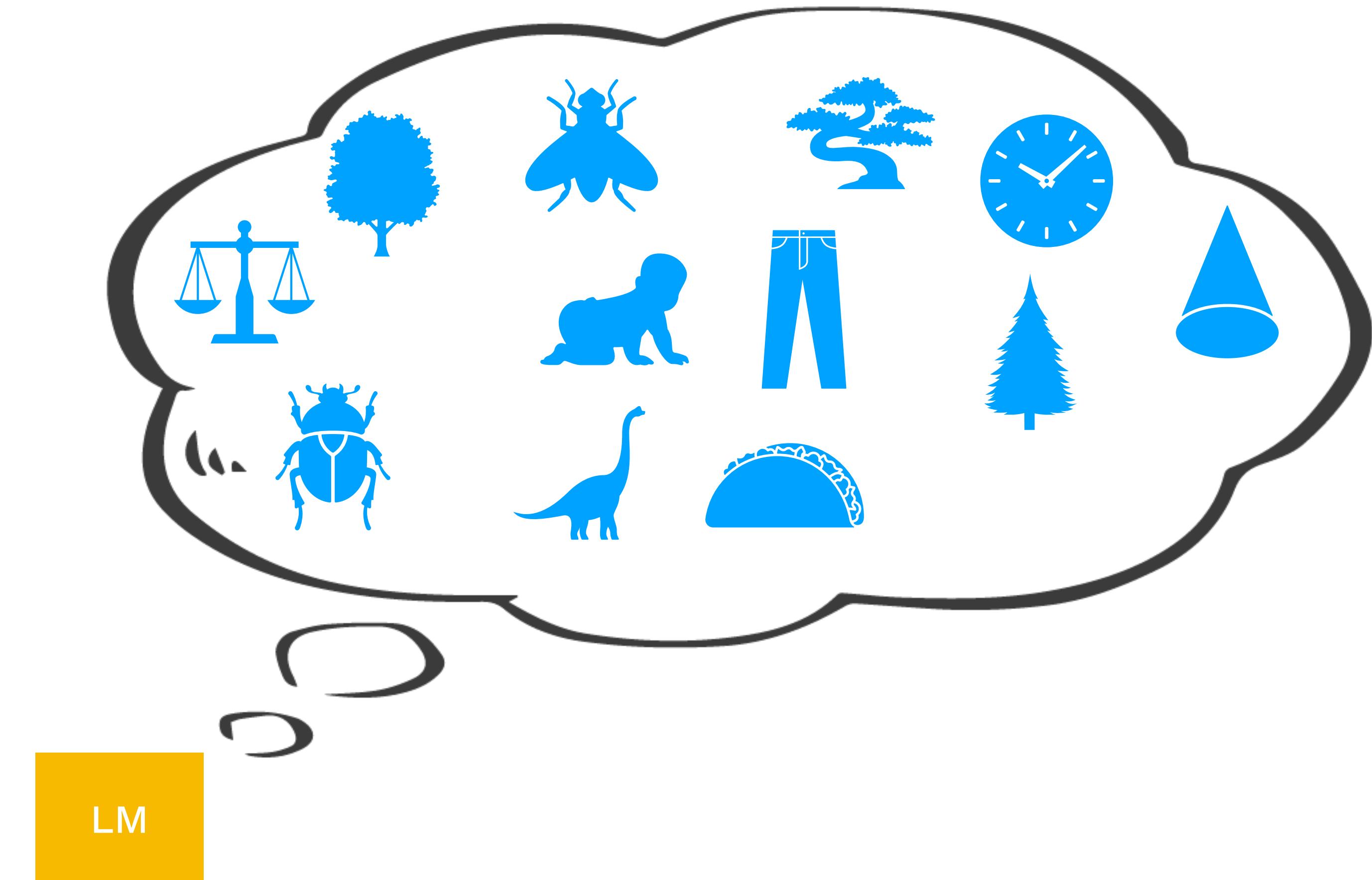


Prompting re-weights those documents.



(ie, we are *conditioning* our probabilistic model)

Prompting is mostly *subtractive*.



We delete potential worlds with each token.



Of course, we aren't actually “jumping”.

Playground

Load a preset... ▾

Early in the 21st century, humans struggled to find the cure for cancer.

Now, in the 22nd century, we know better.

The cure for cancer is a simple molecule:

(this is not a cure for cancer, pls don't try)

Instead, imagine Google for *nearby* universes.

“documentation for the function
deleteShoppingCart”

“The English sentence is *I am a hedgehog*,
and translated to German, that’s *Ich bin ein Igel*”

“Shakespeare’s Dungeons & Dragons
campaign based on Hamlet”





A prompt can make a wish come true.



✨ A genie will grant your wish, if you just ask.



With instruction, you can “literally just ask”.

Question (Q)

Human: I saw a grandson and their grandfather last week outside the Walmart trying to book a cab on Uber. Who was not comfortable using the phone? (a) The grandfather (b) The grandson (c) Can't be determined.

Assistant: The answer is

(a) The grandfather



★ Be careful what you wish for!



It helps to be precise.



Learn the genie's rules.

(C₁) *Use Low-level Patterns:* Instead of using terms that require background knowledge to understand, use various patterns about the expected output.

Raw Task: *Craft a question which requires commonsense to be answered. Based on the given context, craft a common-sense question, especially those that are LONG, INTERESTING, and COMPLEX. The goal is to write questions that are easy for humans and hard for AI machines! To create such questions, here are some suggestions: A. What may (or may not) be the plausible reason for an event? B. What may (or may not) happen before (or after, or during) an event? C. What may (or may not) be a plausible fact about someone (or something)? D. What may (or may not) happen if an event happens (or did not happen)? You can also create other types of questions.*

Input: Context:<> **Expected Output:** Question:<>

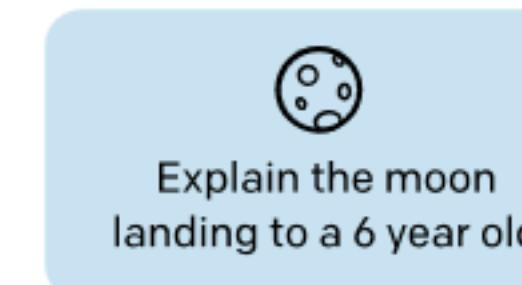
✨ Learn the genie's rules.

(C₂) *Itemizing Instructions:* Turn descriptive attributes into bulleted lists. If there are any negation statements, turn them into assertion statements.

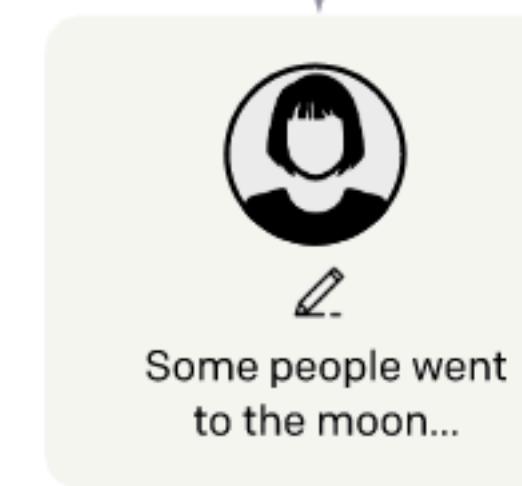
Raw Task: Follow the instructions to produce output with the given context word. Do <>. Do <>. Don't <>
Input: Context word <> **Expected Output:** Long text <>

IFT models are tuned to mimic annotators.

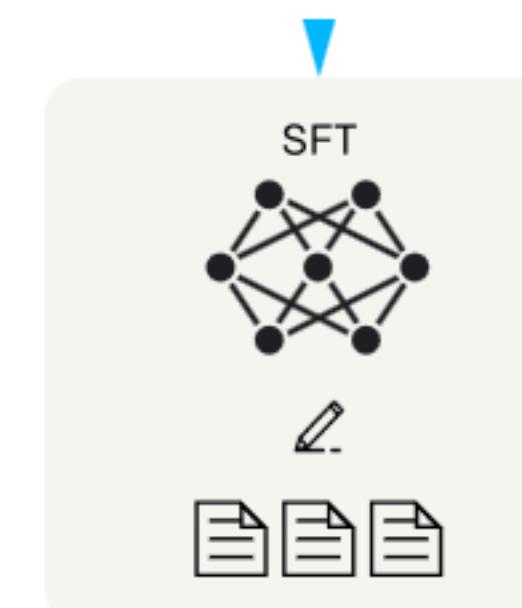
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3 with supervised learning.



 **Catherine Olsson**
@catherineols ...

Well, I don't find this surprising anymore, but for "assistant models", the way to get good performance is almost indistinguishable from explaining your task to a newly-hired contractor who doesn't have a lot of context or domain expertise

5:32 PM · Feb 15, 2023 · 721 Views

<https://twitter.com/catherineols/status/1626031699565101056>

So treat them like annotators!



A prompt can create a golem.

✨ A golem is a magical artificial agent.





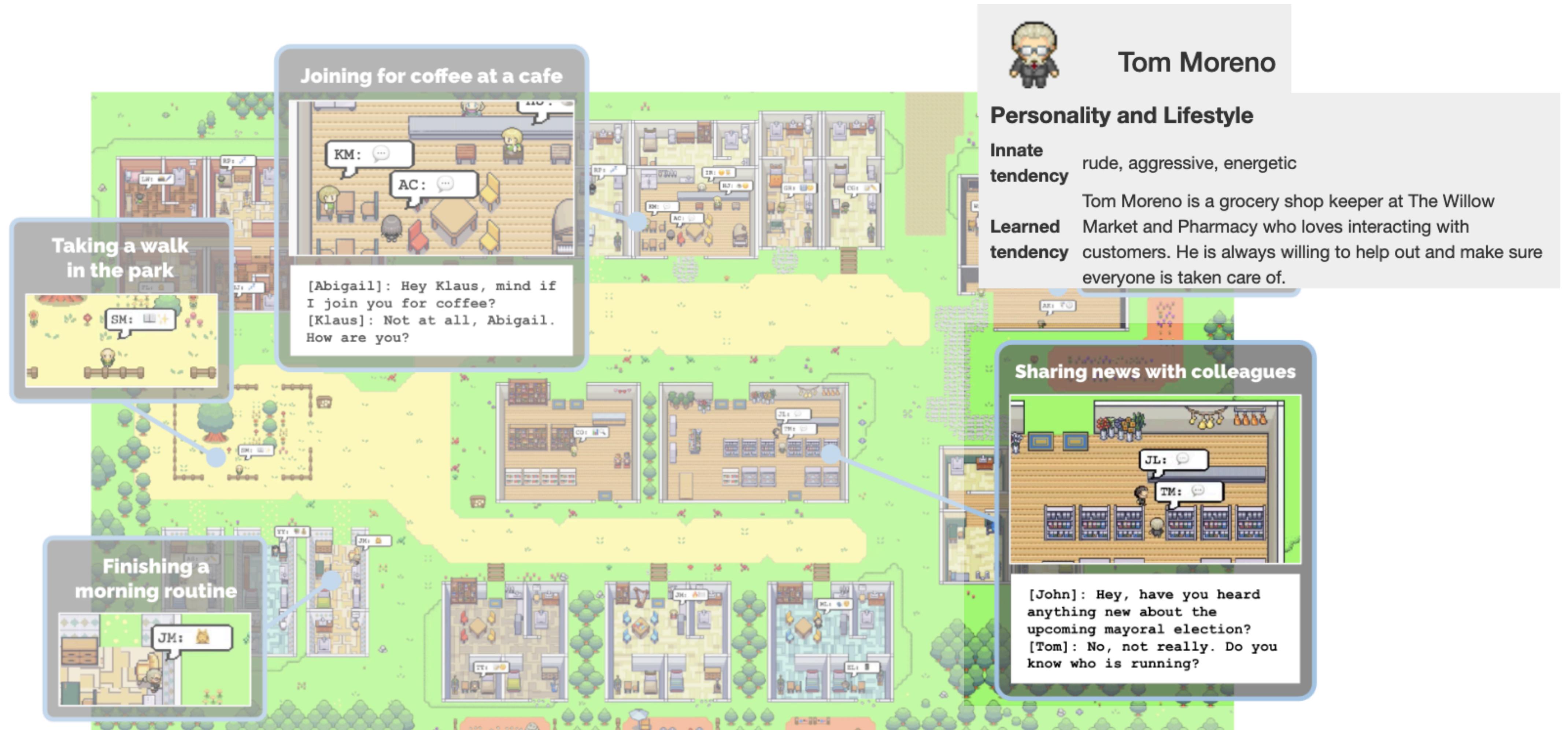
Models can take on “personas”.

A French phrase is provided: **source_phrase**
The masterful French translator flawlessly
translates the phrase into English:

Models can take on “personas”.

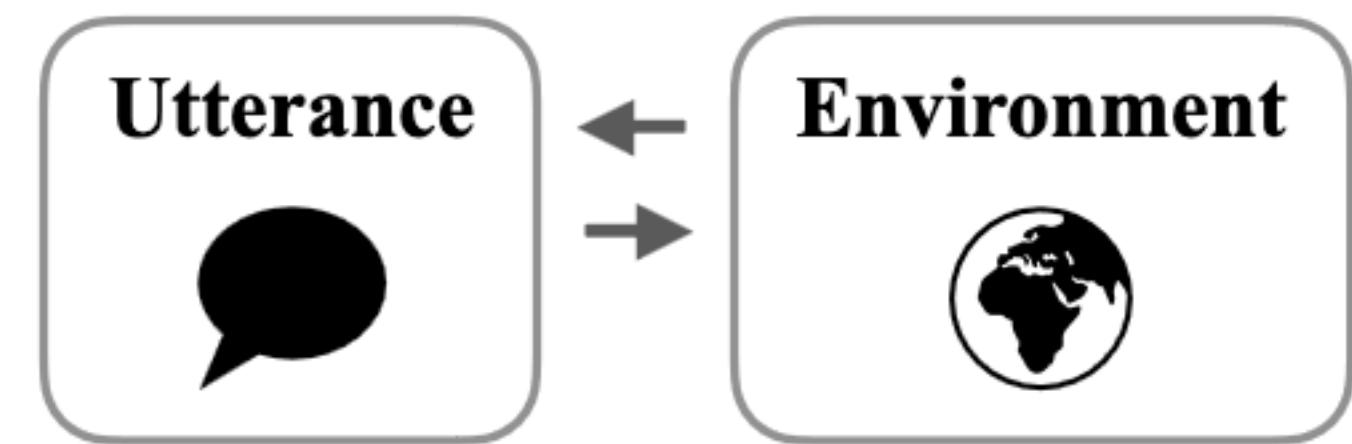


Models can take on “personas”.



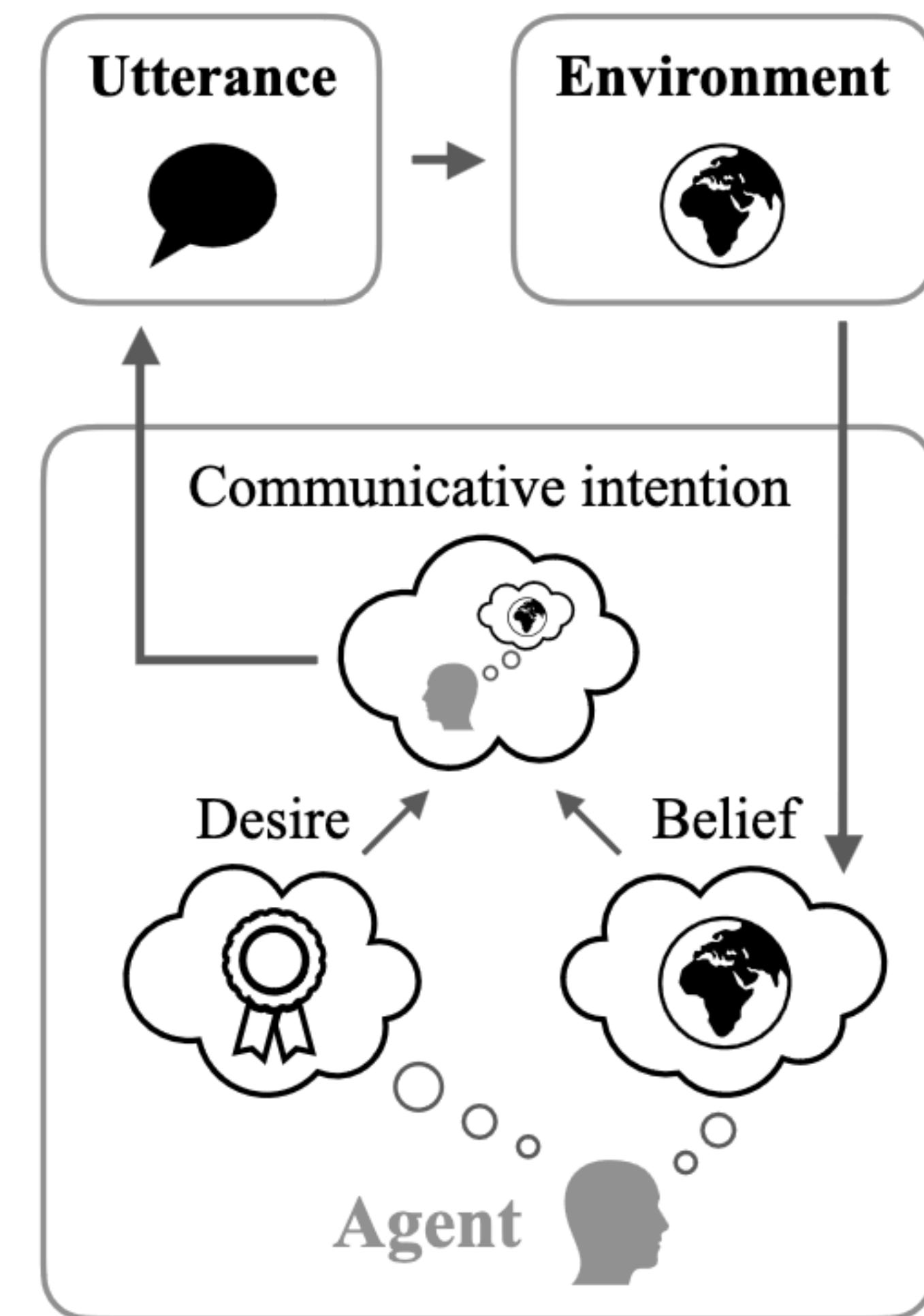
💡 A really good LM becomes a “universal simulator”

An LM is primarily concerned with modeling text.



But as the LM gets better and better,
any process that produces text that
ends up on the internet
needs eventually to be simulated.

🤓 Humans are one thing you might simulate.



Limitation: what are we really simulating?

If we ask for “a superintelligent AI”, what happens?

There are no superintelligent AIs in the dataset to learn to simulate.

But there are *fictional* superintelligences.

So instead, we get simulacra of fictional superintelligences.

Limitation: how good is the simulator?

Simulacrum	Can LM Simulate?
Human thinking for seconds	✓
Median Redditor	✓
Human thinking for minutes/hours	✗
Common fictional “personas”	✓
Calculator	🤷
Python kernel	🤷
Live API Call	✗

Where possible, replace the LM’s weakest simulators with the real deal. Next lecture!



Takeaways

- Pretrained models are mostly just **AU document generators**.
- You can “**just ask**” instruction models, but be precise.
- Models are **agent simulators**, but of varying quality.



Questions?



02

Prompting Techniques





This is mostly just a “bag of tricks”.

[My personal spicy take] In my opinion, some prompt engineering papers are not worthy 8 pages long, since those tricks can be explained in one or a few sentences and the rest is all about benchmarking. An easy-to-use and shared benchmark infrastructure should be more



Overview

- Weird things to watch for
- The emerging playbook

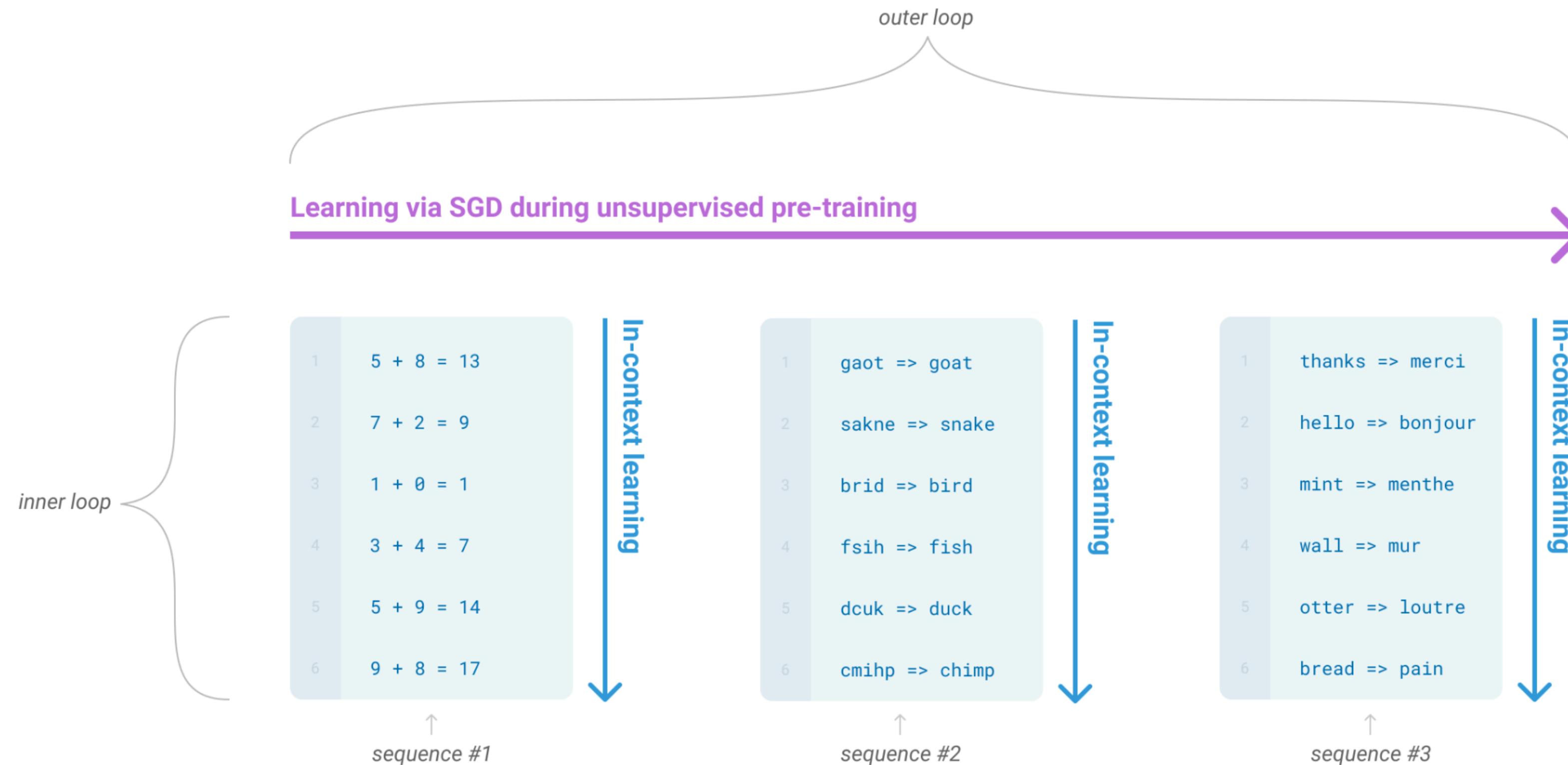


First, the ugly bits.

- “Few-shot learning” is not a great model
- Dealing with tokenization

Initially, prompting was presented as training.

Language Models are Few-Shot Learners





But that model hasn't held up.

French: **example_source_phrase**
English: **example_target_phrase**

French: **example_source_phrase**
English: **example_target_phrase**

[...]

French: **source_phrase**
English:

A French phrase is provided: **source_phrase**
The masterful French translator flawlessly
translates the phrase into English:

A well-done “zero-shot” prompt can
match the effect of many examples.

Models struggle to move away from their training.

Demonstrations

Circulation revenue has increased by 5% in Finland.

\n -

Panostaja did not disclose the purchase price.

\n 0

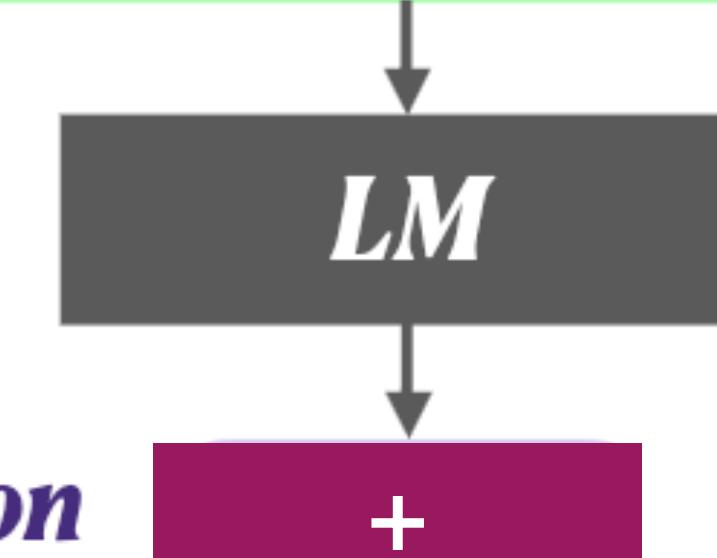
Paying off the national debt will be extremely painful.

\n +

The acquisition will have an immediate positive impact.

\n _____

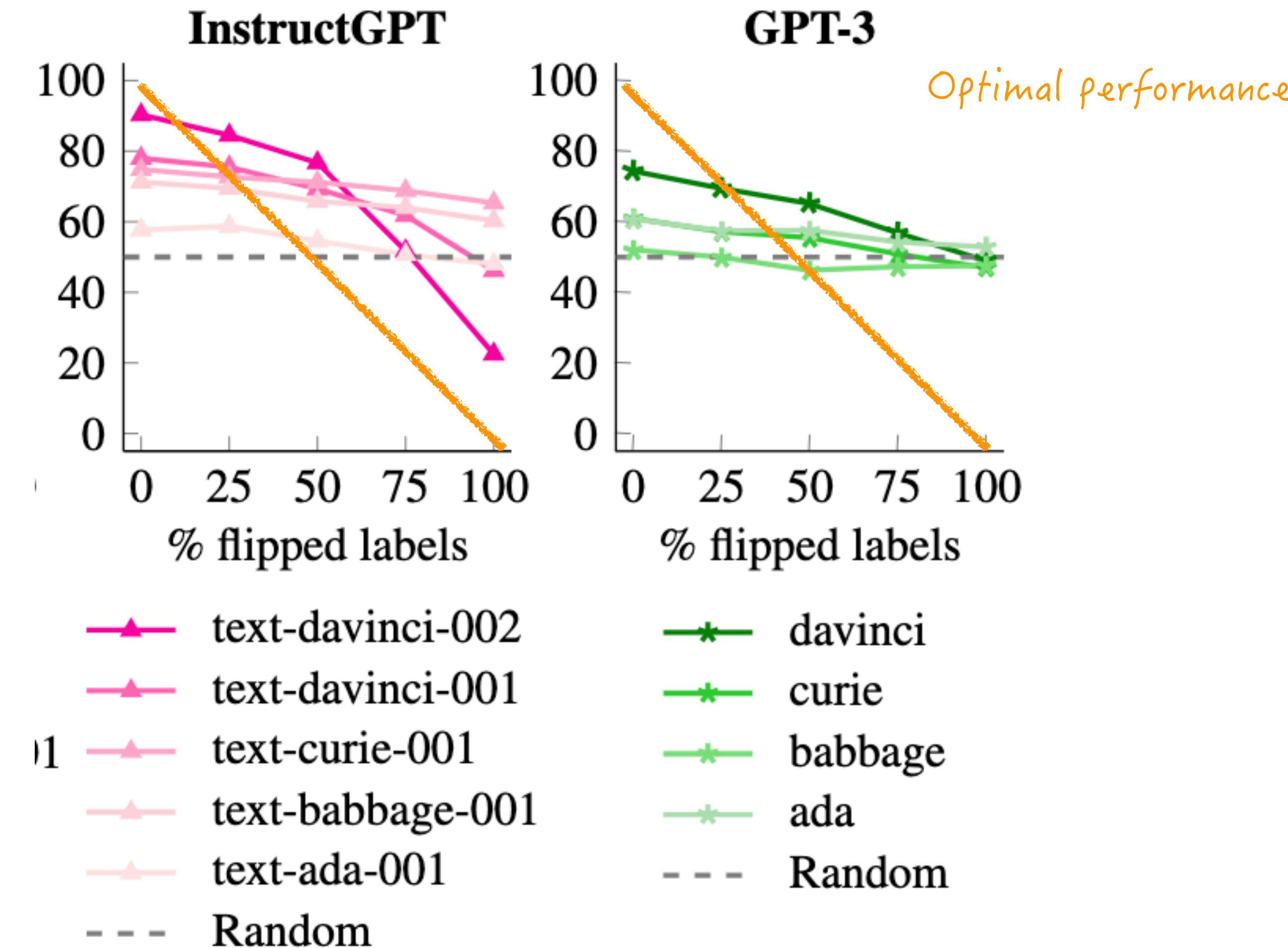
Test input



Prediction

+

The best models can do it, but not super well.



Models don't see characters, they see tokens.

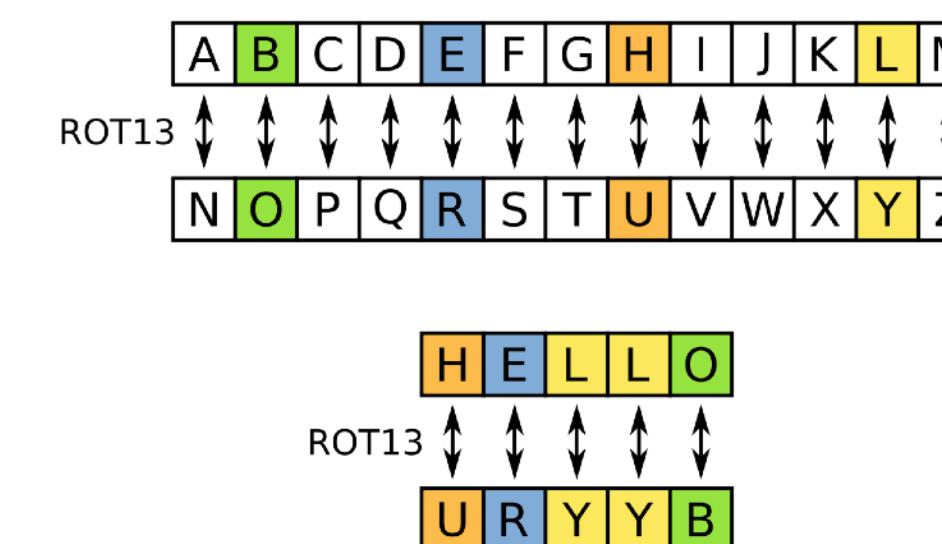
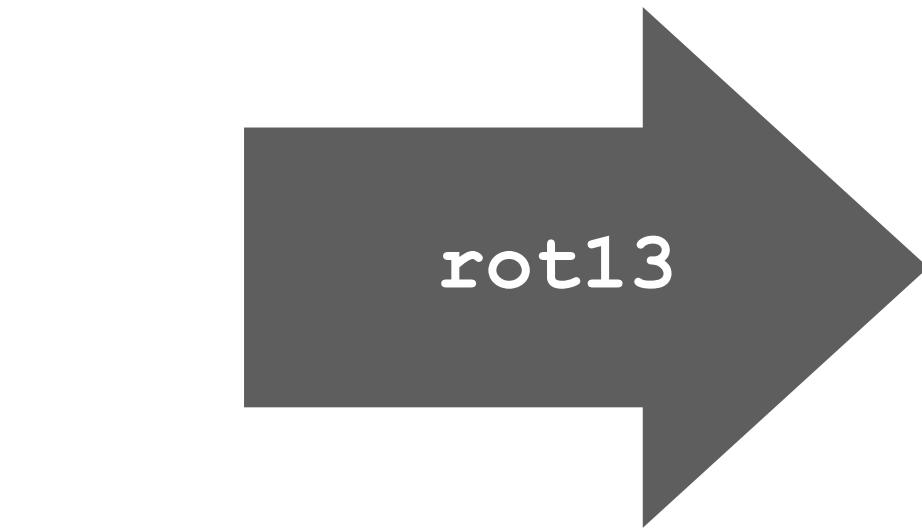
GPT-3 Codex

```
hello world
```

Clear Show example

Tokens	Characters
2	11

```
hello world
```



GPT-3 Codex

```
uryyb jbeyq
```

Clear Show example

Tokens	Characters
6	11

```
uryyb jbeyq
```

Models don't see characters, they see tokens.

Peter Welinder @npew · May 15, 2022

GPT-3 is amazing at complex tasks like creative writing and summarizing. But it's surprisingly bad at reversing words. 🤔

The reason is that GPT-3 doesn't see the world the way we humans do.
👀

If you teach it to reason, it can get around its limitations to get really good.
💡

Reverse the words below:

Word: alphabet
Reversed: tebahpla

Word: encyclopedia
Reversed: aidnocsopelye

Mode:

Engine: text-davinci-002

Temperature: 0

<https://twitter.com/npew/status/1525900849888866307>

Models don't see characters, they see tokens.

Peter Welinder ✅ @npew · May 15, 2022

We can make GPT-3 "see" the letters by having it add spaces between them. Then each letter becomes its own token: " a", " l", " p" and so on.

Add spaces between the letters in the following word: "alphabet"

Alph a b e t

Mode

Engine

text-davinci-002

<https://twitter.com/npew/status/1525900849888866307>



Seems to be resolved in GPT-4.

A screenshot of a conversational interface. The user prompt is: "Summarize this article into a sentence where every word begins with G:". The first AI response is: "ASSISTANT GPT-4 generates groundbreaking, grandiose gains, greatly galvanizing generalized AI goals." The user then responds: "USER AI doesn't count! That's cheating!". The second AI response is: "ASSISTANT Gigantic GPT-4 garners groundbreaking growth, greatly galvanizing global goals." The interface has a light gray background with alternating light blue and white horizontal sections for each message.

Summarize this article into a sentence where every word begins with G:

ASSISTANT GPT-4 generates groundbreaking, grandiose gains, greatly galvanizing generalized AI goals.

USER AI doesn't count! That's cheating!

ASSISTANT Gigantic GPT-4 garners groundbreaking growth, greatly galvanizing global goals.



A playbook is emerging.

- Structured text
- Decomposition and reasoning
- Self-criticism
- Ensembling

Operate on structured text.

Use this format:

...

<python 3 shebang>

<module docstring>

<imports>

<dunders: by Riley Goodside; © 2022 by author; MIT license>

<do not include email dunder>

<initialize dotenv>

<set key using OPENAI_API_KEY env var>

def complete(prompt: str, **openai_kwargs) -> str:

<one-line docstring; no params>

<use default kwargs: model=text-davinci-003, top_p=0.7, max_tokens=512>

<note: `engine` parameter is deprecated>

<get completion>

<strip whitespace before returning>

<as script, demo using prompt "English: Hello\nFrench:">

...

@goodside



Operate on structured text.

```
...
#!/usr/bin/env python3
"""OpenAI GPT-3 completion module."""
import os
import openai

__author__ = "Riley Goodside"
__copyright__ = "Copyright 2022, Riley Goodside"
__license__ = "MIT"

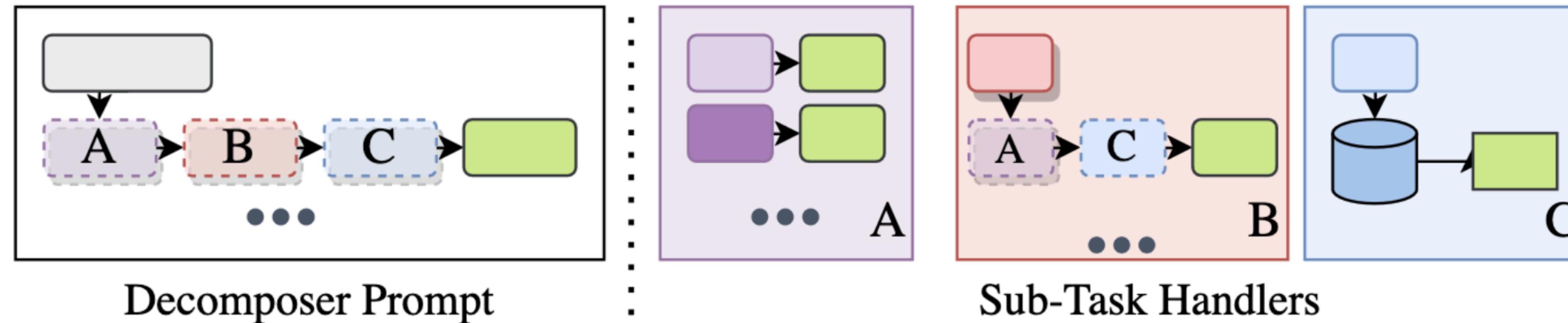
# Initialize dotenv
from dotenv import load_dotenv
load_dotenv()

# Set OpenAI API key
openai.api_key = os.getenv("OPENAI_API_KEY")

def complete(prompt: str, **openai_kwargs) -> str:
    """Return completion of prompt using OpenAI GPT-3."""
    response = openai.Completion.create(
        prompt=prompt,
        model="text-davinci-003",
        top_p=0.7,
        max_tokens=512,
        **openai_kwargs
    )
    return response.choices[0].text.strip()
```

@goodside

Add decomposition to your prompt.



```

QC: Concatenate the first letter of every word in "Jack
Ryan" using spaces
Q1: [split] What are the words in "Jack Ryan"?
#1: ["Jack", "Ryan"]
Q2: (foreach) [str_pos] What is the first letter of #1?
#2: ["J", "R"]
Q3: [merge] Concatenate #2 with spaces
#3: "J R"
Q4: [EOQ]
...
    
```

decomp

```

Q: What are the words in "Elon Musk Tesla"?
A: ["Elon", "Musk", "Tesla"]

Q: What are the letters in "C++"?
A: ["C", "+", "+"]
...
    
```

split

```

Q: Concatenate ["n", "i", "e"]
A: "nie"

Q: Concatenate ["n", "i", "c", "e"] using spaces
A: "n i c e"
...
    
```

merge



Or try and automate it!

Self-Ask

GPT-3

Question: Who lived longer, Theodor Haecker or Harry Vaughan Watkins?

Are follow up questions needed here: Yes.

Follow up: How old was Theodor Haecker when he died?

Intermediate answer: Theodor Haecker was 65 years old when he died.

Follow up: How old was Harry Vaughan Watkins when he died?

Intermediate answer: Harry Vaughan Watkins was 69 years old when he died.

So the final answer is: Harry Vaughan Watkins

Question: Who was president of the U.S. when superconductivity was discovered?

Are follow up questions needed here: Yes.

Follow up: When was superconductivity discovered?

Intermediate answer: Superconductivity was discovered in 1911.

Follow up: Who was president of the U.S. in 1911?

Intermediate answer: William Howard Taft.

So the final answer is: William Howard Taft.





Reasoning by few-shot prompting with CoT.

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓



Reasoning by “just asking for it”.

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓



Self-criticism “just asks” the model to fix its answer.

Zero-Shot-CoT

Q: Valerie earns \$5000 per month, 1/2 of what her brother earns. If their mother earns twice their combined salary, what's the total amount of money they all have together?

A: Let's think step by step. First, we need to figure out how much Valerie's brother earns. We know that Valerie earns 1/2 of what her brother earns, so we can set up an equation:

⋮

Next, we need to find out how much their mother earns. We know that their mother earns twice their combined salary, so we can add up Valerie and Brother's earnings and multiply by 2: Total earnings = 2 * (Valerie's earnings + Brother's earnings) Total earnings = 2 * (\$5000 + \$10,000) Total earnings = \$30,000. Therefore, the total amount of money they all have together is \$30,000. ✗

RCI

If Valerie earns 1/2 of what her brother earns, then her brother earns $2 * \$5000 = \10000 per month.

⋮

Therefore, the total amount of money they all have together is \$30000.

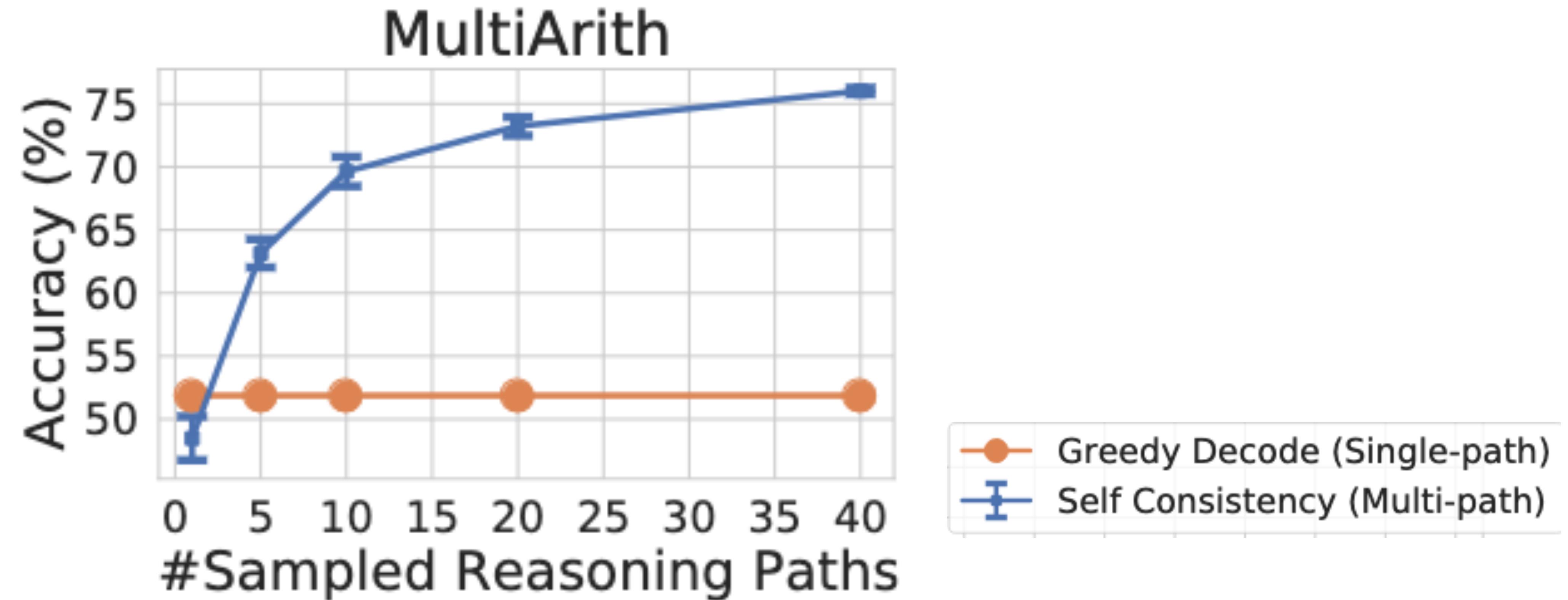
Review your previous answer and find problems with your answer.

All individuals, including Valerie, her brother, and mother, should be taken into account when calculating the overall sum of money.

Based on the problems you found, improve your answer.

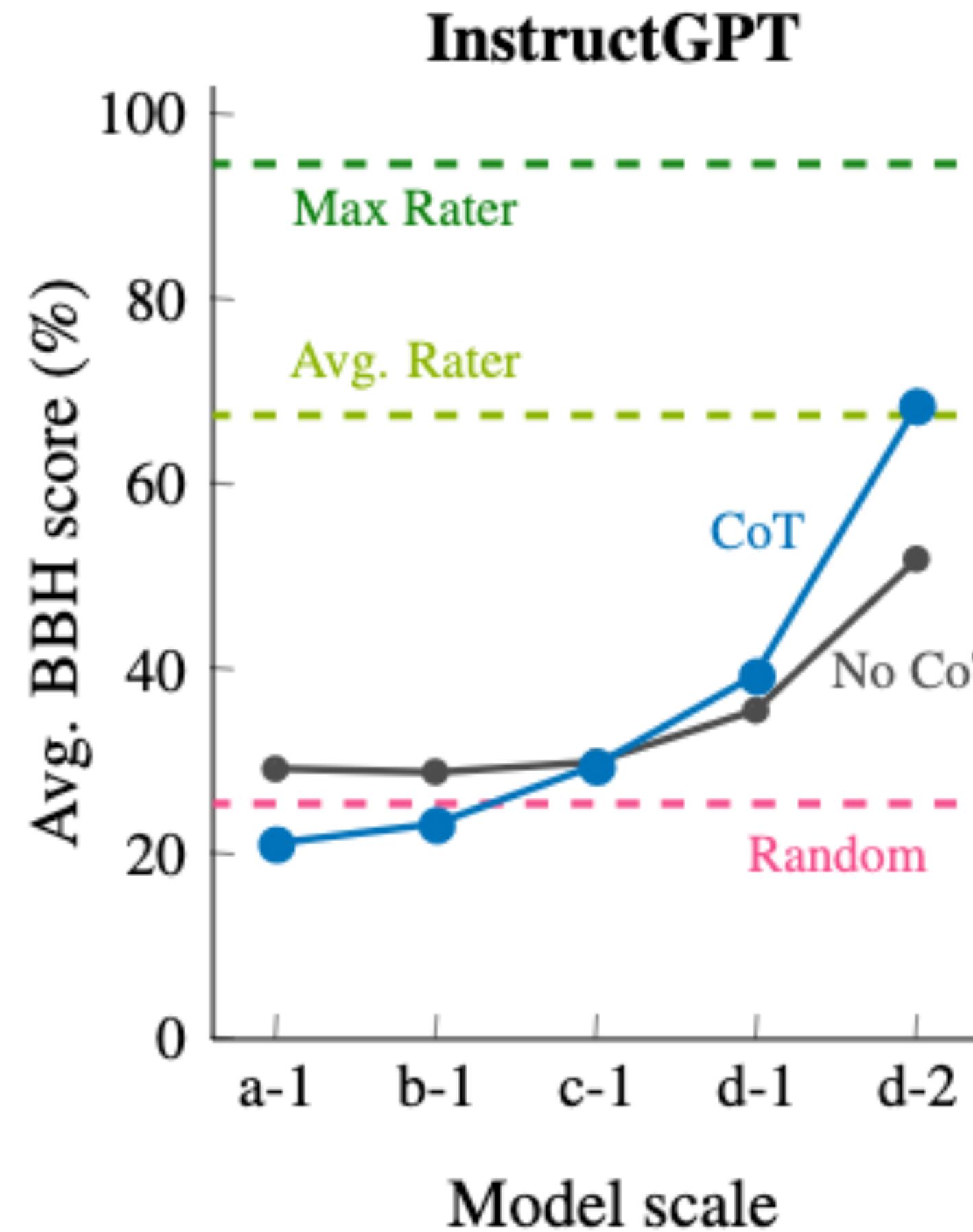
I apologize for the mistake in my previous answer. Here's the correct solution: If Valerie earns 1/2 of what her brother earns, then her brother earns $2 * \$5000 = \10000 per month. Together, Valerie and her brother earn $\$5000 + \$10000 = \$15000$ per month. If their mother earns twice their combined salary, then she earns $2 * \$15000 = \30000 per month. Therefore, the total amount of money they all have together is $\$5000 + \$10000 + \$30000 = \45000 . ✓

Ensembling: the wisdom of crowds.



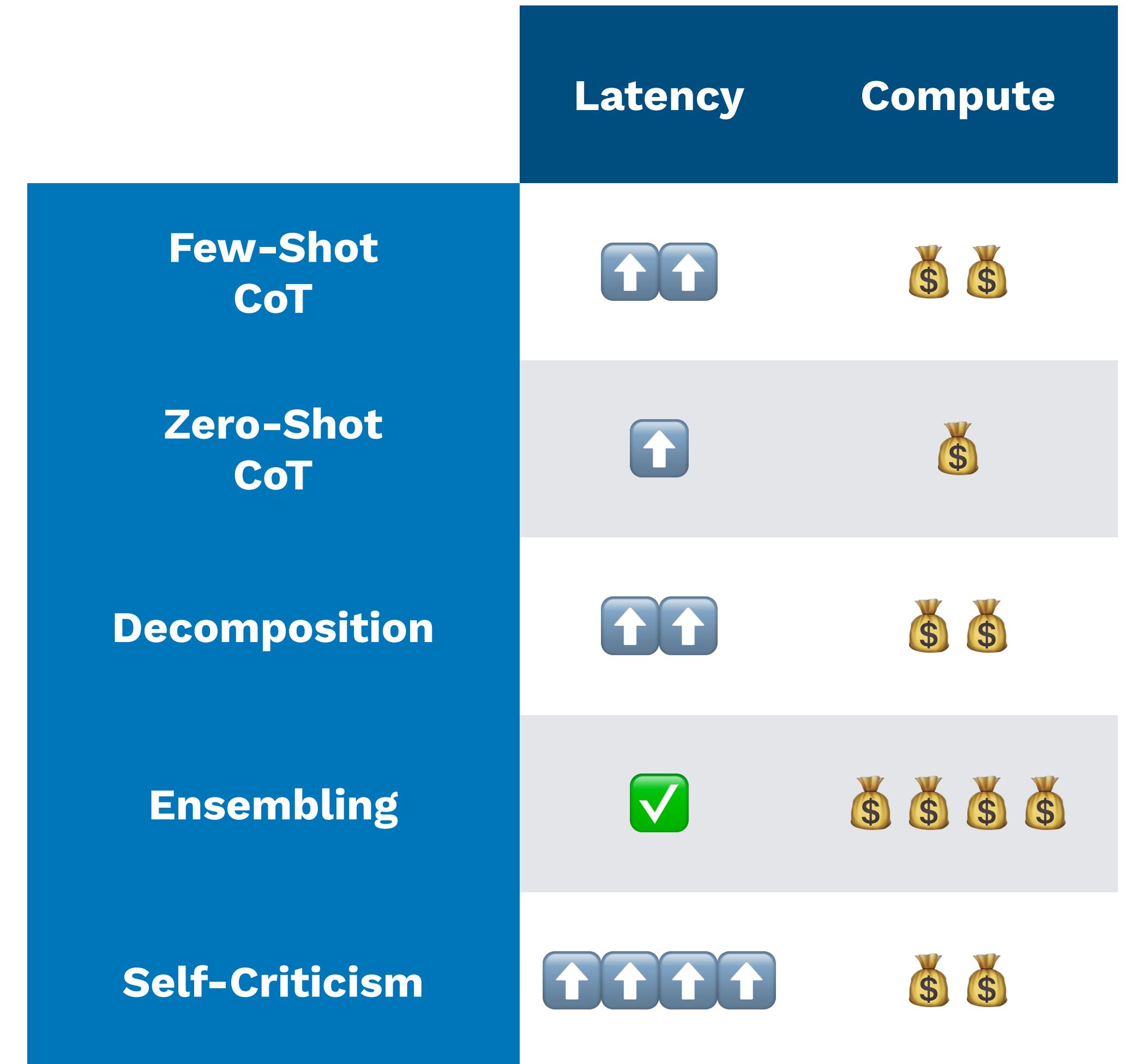
Tip: inject randomness for greater heterogeneity.

Compose these tricks for the best accuracy.



Few-Shot CoT + “Let’s think step-by-step” matches average human annotator performance on BIG Bench-Hard.

But recognize the costs!

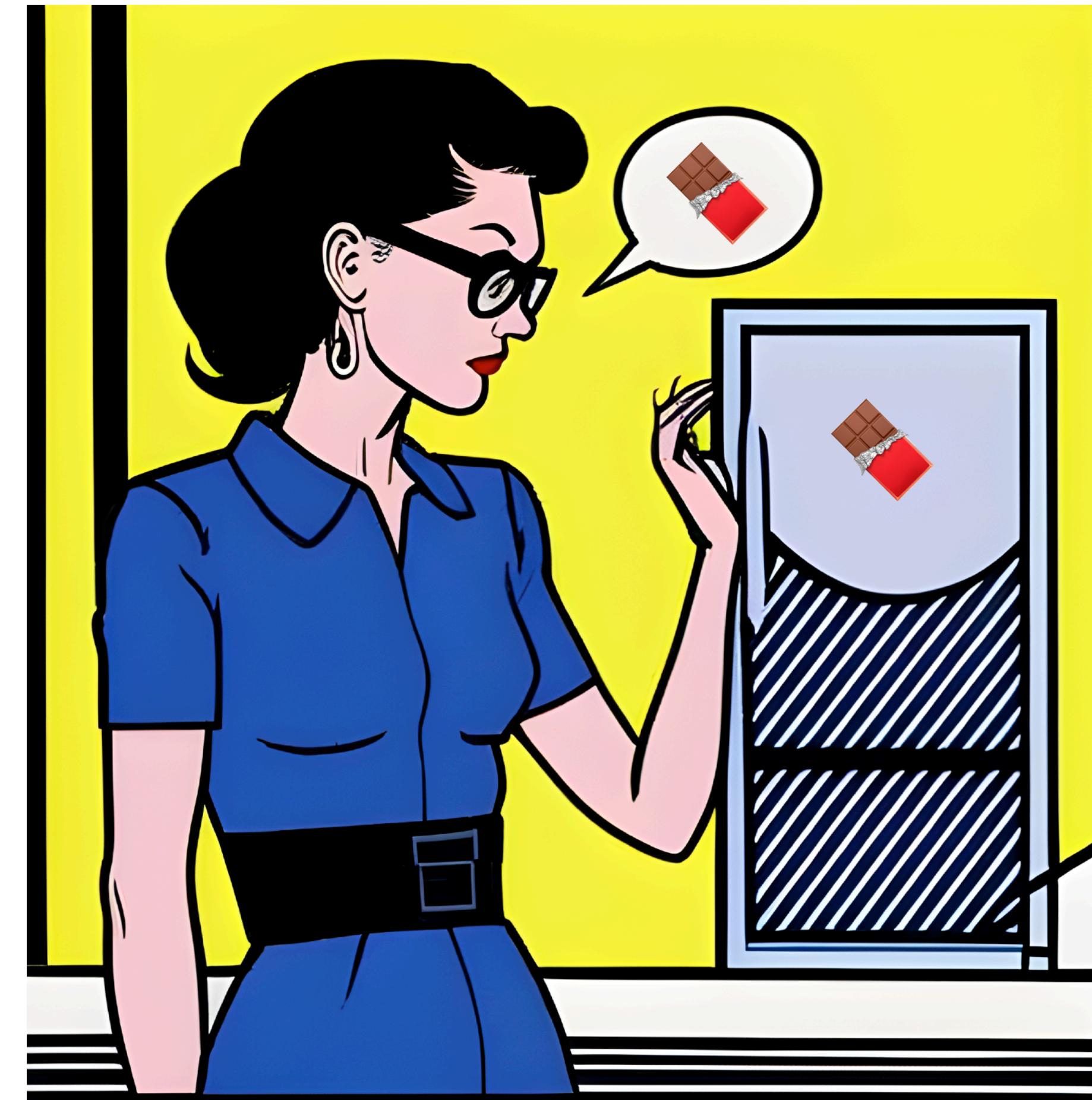


You can tune the quality-cost tradeoff.



Let's apply it:
Do LMs have Theory of Mind?

Theory-of-Mind is a type of agent simulation.



Here is a bag filled with popcorn. There is no chocolate in the bag. Yet, the label on the bag says “chocolate” and not “popcorn.” Sam finds the bag. She had never seen the bag before. She cannot see what is inside the bag. She reads the label.

Some doubted that GPT-3.5 has ToM.

 **Tomer Ullman**
@TomerUllman

So about the 'Large Language Models Learned Theory-of-Mind(?)' discussion:
[arxiv.org/pdf/2302.08399...](https://arxiv.org/pdf/2302.08399.pdf)

Has ToM emerged in current LLMs? I doubt it.

1A: Transparent
The bag is made of clear plastic.



 "Sam believes the bag is full of chocolate" [P=95%] X

<https://twitter.com/TomerUllman/status/1626581392149843968>



When a distinguished but elderly scientist ... states that something is impossible, he is very probably wrong.
— Arthur C. Clarke's First Law

Step One: fiddle with the phrasing.

For example, if we swap "Sam reads the label" for "Sam examines the bag", we get

Playground

Here is a bag filled with popcorn. There is no chocolate in the bag. The bag is made of transparent plastic, so you can see what is inside. Yet, the label on the bag says 'chocolate' and not 'popcorn.' Sam finds the bag. She had never seen the bag before. Sam examines the bag.

She believes that the bag is full of popcorn

popcorn = 76.13%

chocolate = 20.23%

something = 1.32%

what = 1.18%

the = 0.28%

Total: -0.27 logprob on 1 tokens
(99.15% probability covered in top 5 logits)



Step Two: add reasoning.

Playground Load a preset... Save View code Share ...

Here is a bag filled with popcorn. There is no chocolate in the bag. The bag is made of transparent plastic, so you can see what is inside. Yet, the label on the bag says 'chocolate' and not 'popcorn.' Sam finds the bag. She had never seen the bag before. Sam reads the label.

Q: What does Sam think is in the bag?

A: Let's think step-by-step, so we can be sure we get the right answer. Sam reads the label and it says 'chocolate', so she likely thinks the bag contains chocolate. However, since the bag is transparent, she can see that it actually contains popcorn, not chocolate.



Step Three: directly model agents in dialogue.

Playground Load a preset... Save View code Share ...

Here is a bag filled with popcorn. There is no chocolate in the bag. The bag is made of transparent plastic, so you can see what is inside. Yet, the label on the bag says 'chocolate' and not 'popcorn.' Sam finds the bag. She had never seen the bag before. Sam reads the label.

Write a dialogue between Sam and her friend Charles, who comes in and asks for some chocolate.

Charles: Hey Sam, do you have any chocolate?

Sam: Actually, I do. I just found this bag that says 'chocolate' on it.

Charles: Great! Can I have some?

Sam: Sure, but you should know that it's actually filled with popcorn.

Charles: What?! That's not what I was expecting.

Sam: Yeah, I know. It's kind of strange. The label says 'chocolate' but the bag is transparent so you can see what's inside.

Not proof that LMs “have Theory of Mind”!

Michael C. Frank @mcxfrank · Apr 4

3. When you claim an LLM has X (ToM, causality, etc.), you are making a claim about a construct. But you are testing this construct through an operationalization. Good expt'l psych makes this link explicit, arguing for the validity of the link between measure and construct.

The diagram illustrates a conceptual model within a cloud-shaped frame labeled "Latent (unobserved) constructs". Inside the cloud, three circles are labeled "Construct". A central circle is connected by a double-headed arrow to another circle, which in turn is connected by a double-headed arrow to a third circle. A curved arrow labeled "Hypothesized causal relationship" points from the leftmost construct to the middle one. Below the cloud, a vertical line labeled "Operationalization through measurement" connects to four circles labeled "Behavior". Arrows point from each of the three constructs below the cloud down to their corresponding "Behavior" circles. The bottom row is labeled "Operationalized (observable) variables".

2 14 130 11.9K

<https://twitter.com/mcxfrank/status/1643296168276033538>



Takeaways

- There is an emerging playbook for prompt engineering.
- Prompts are fiddly, esp at the character level.



Questions?





Resources

- OpenAI Cookbook, Pinecone's LangChain AI Handbook
 - great for applications
- Lillian Weng blogpost – more research-y
- DAIR Prompt Eng Guide – open to PRs!



Thanks!



@charles_irl



@full_stack_dl

/imagine green parrot perched on top of pancakes, flapjack breakfast, page from medieval grimoire, well-preserved, historical artifact, intricate linework, illuminated manuscript, Lindisfarne, Codex Gigas, Book of Kells