

Working with LLMs

Manoel Horta Ribeiro
manoel@cs.princeton.edu



Quizz Time!





MENU

APPETIZER

Language Models

MAIN COURSE

Using Language Models

DESSERT

Miscellanea



MENU

APPETIZER

Language Models

MAIN COURSE

Using Language Models

DESSERT

Miscellanea

What is a Language Model?

- Given a sequence of tokens, predict the probability of the next token:

$$P(x_{t+1} \mid x_1, x_2, \dots, x_t)$$

- Create text by sampling, appending, and repeating
 - When to stop? Create a token that ends the text!

The cat sat on the ____

$$P(\text{mat} \mid \text{The cat sat on the}) = 0.62$$

$$P(\text{floor} \mid \text{The cat sat on the}) = 0.21$$

...

$$P(\text{moon} \mid \text{The cat sat on the}) = 0.01$$

...

$$P(<\text{EOS}> \mid \text{The cat sat on the}) = 0.0005$$

How do we choose the next token?

- **Greedy decoding:** always choose the most likely token.

$$x_{t+1} = \arg \max_x P(x \mid x_{1:t})$$

- **Sampling:** $P_i(T) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$

- Low T : more concentrated
- High T : flatter, more varied

The cat sat on the ____

$$P(\text{mat} \mid \text{The cat sat on the}) = 0.62$$

$$P(\text{floor} \mid \text{The cat sat on the}) = 0.21$$

...

$$P(\text{moon} \mid \text{The cat sat on the}) = 0.01$$

...

$$P(<\text{EOS}> \mid \text{The cat sat on the}) = 0.0005$$

Simple language models

- A simple language model:

$$P(w_t \mid w_{t-1}) = \frac{\#(w_{t-1}, w_t)}{\#(w_{t-1})}$$

- Only consider the next word!

- $P(\text{sat} \mid \text{The cat}) = P(\text{sat} \mid \text{cat}) =$?

- $P(\text{mat} \mid \text{The cat sat on the}) = P(\text{sat} \mid \text{mat}) =$?

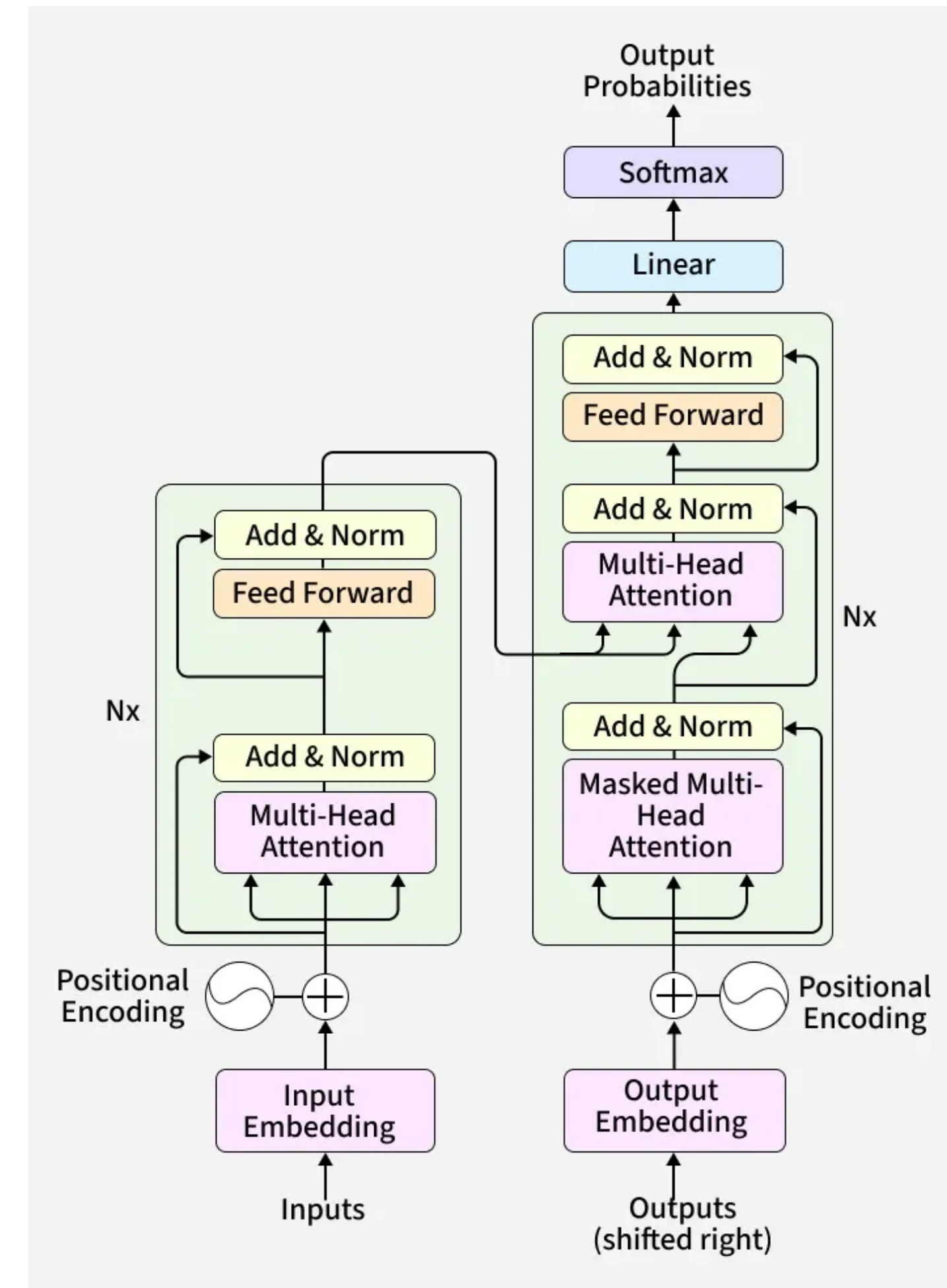
the cat sat on the mat
the cat sat on the floor
the cat slept on the couch
the dog sat on the floor

Modern Day (Large) Language Models

- Same goal! Predict the next token!

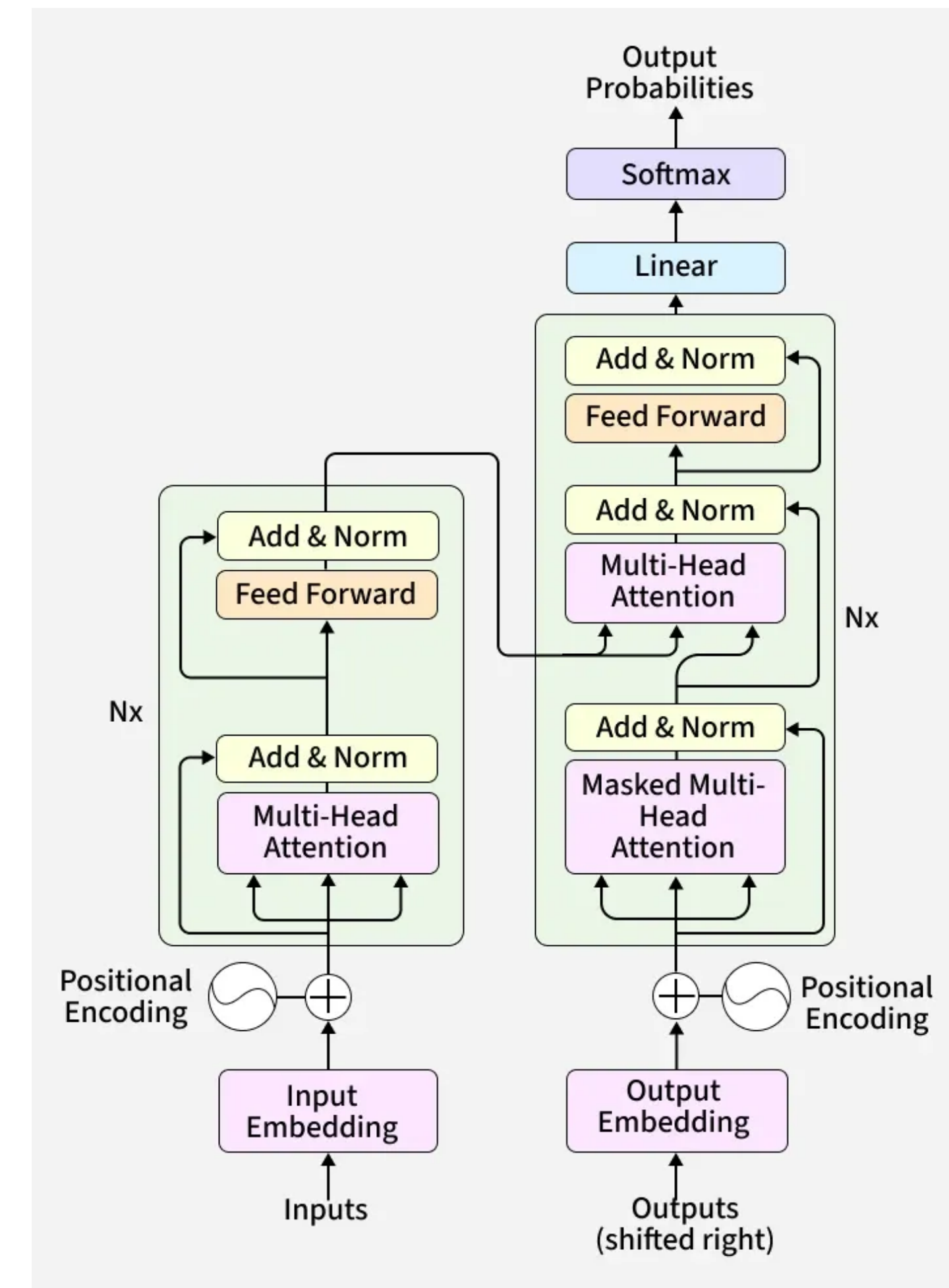
$$P(x_{t+1} \mid x_1, x_2, \dots, x_t)$$

- But instead of simple counts...
 - A neural network learns how to predict the next token...
 - Architecture: Transformers
 - Billions of parameters, enormous training datasets

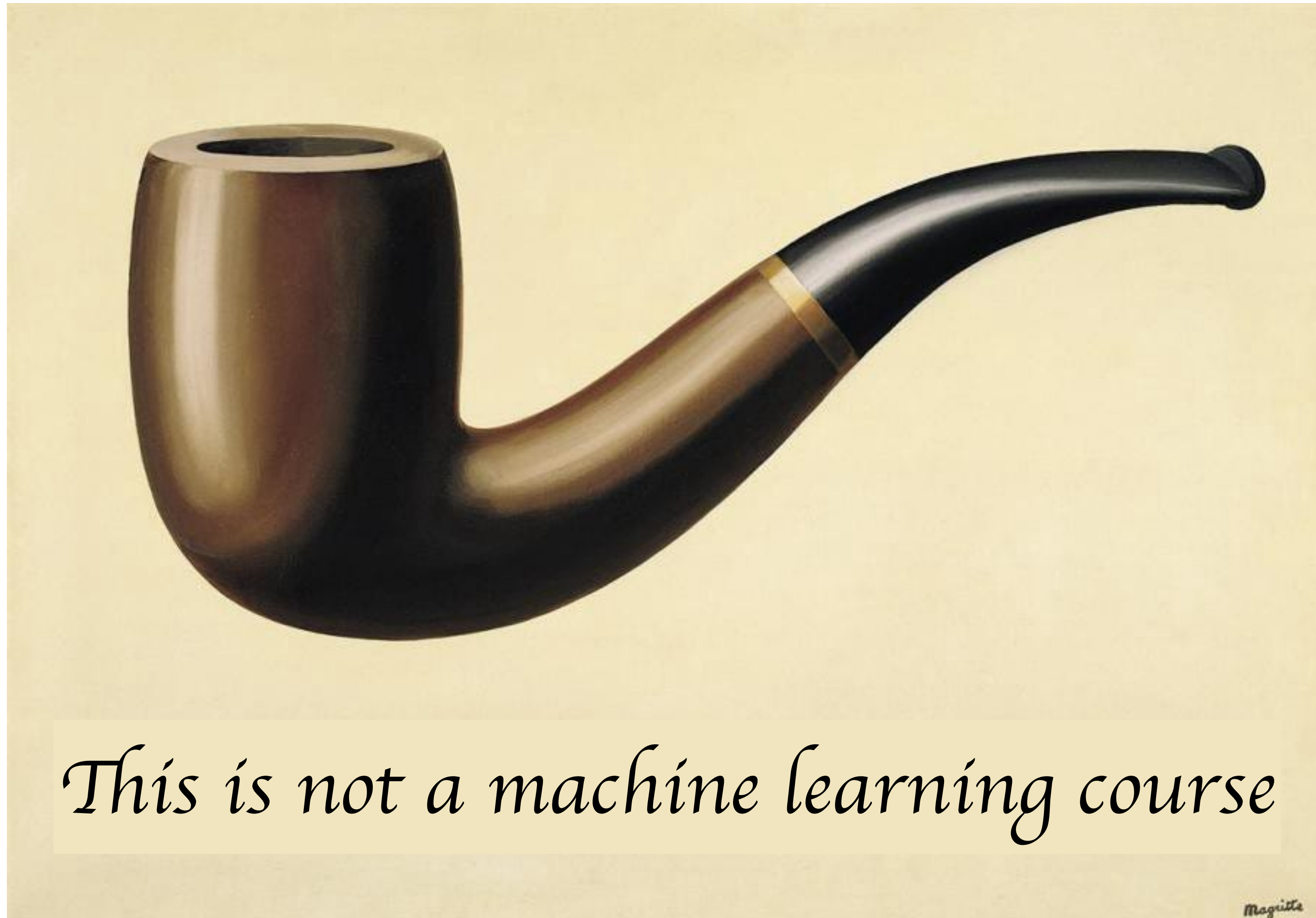


Three things to know about attention

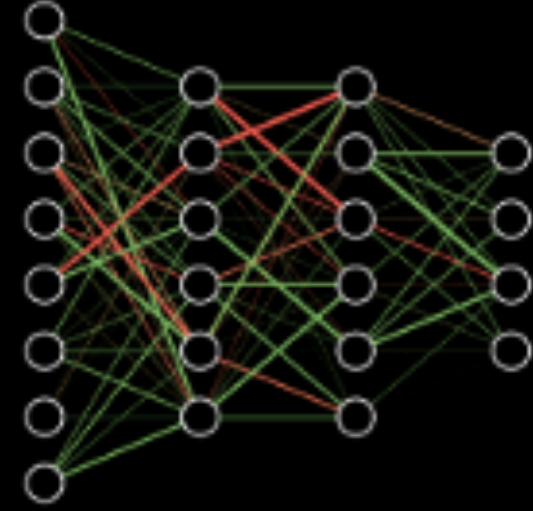
- Attention lets tokens “interact” with other tokens.
- Attention dynamically determines which connections are relevant.
- Attention is expensive for long sequences $O(n^2)$.



The treachery of COS 424



Neural Networks



Neural networks

3Blue1Brown · Course

10 videos Last updated on Jul 21, 2026



▶ Play

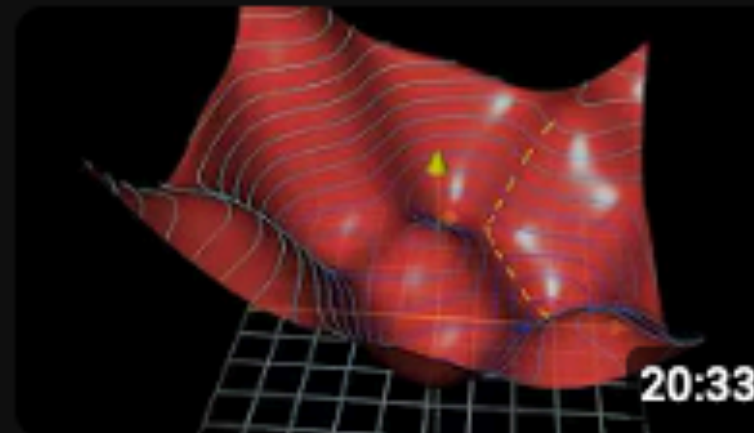
💬 Comments

Learn the basics of neural networks and backpropagation, one of the most important algorithms for the modern world.



But what is a neural network? | Deep learning chapter 1

▷ 24M 8y ago



Gradient descent, how neural networks learn | Deep Learning Chapter 2

▷ 9.5M 8y ago



Backpropagation, intuitively | Deep Learning Chapter 3

▷ 6.2M 8y ago



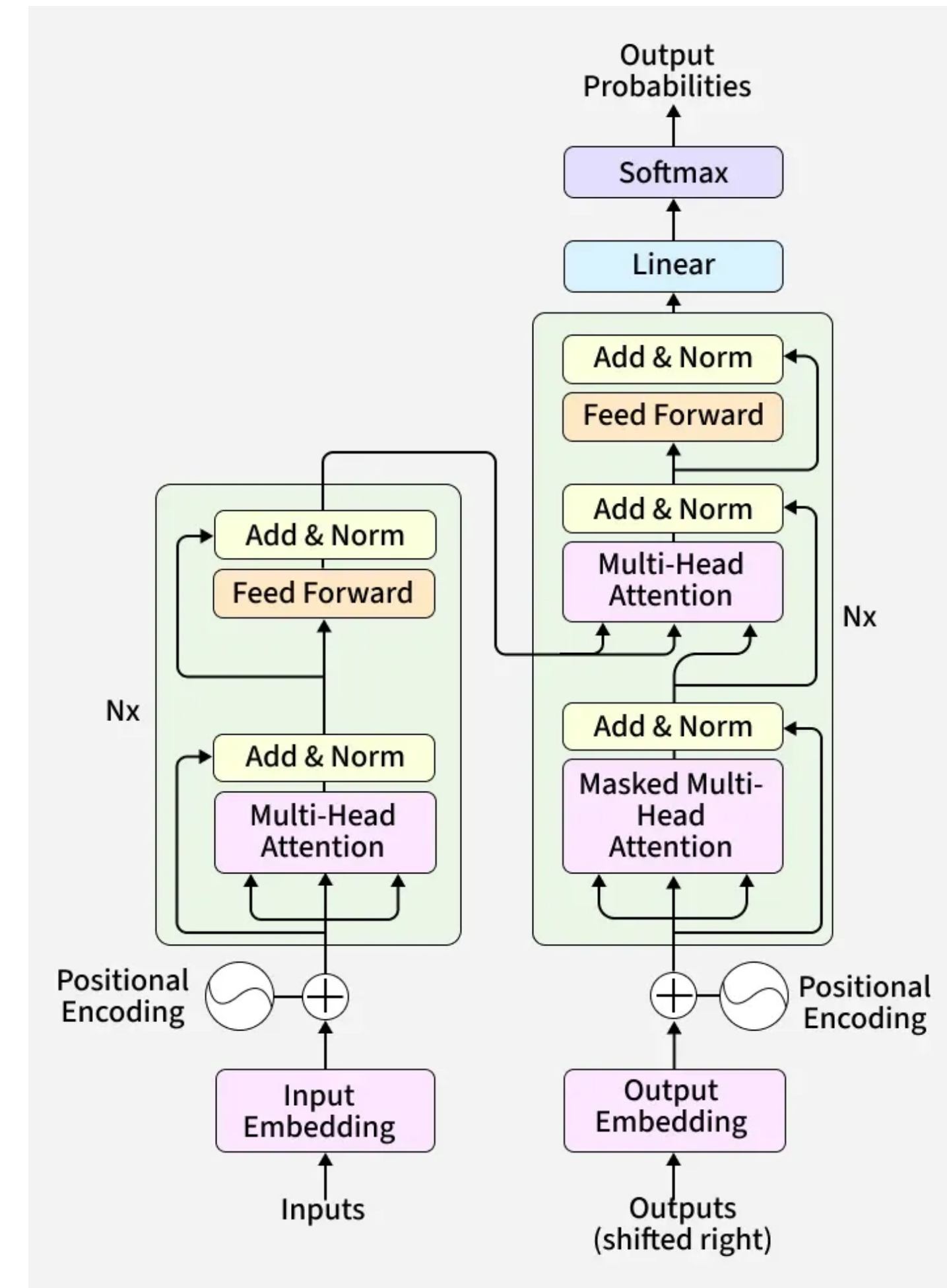
Backpropagation calculus | Deep Learning Chapter 4

▷ 3.9M 8y ago

3Blue1Brown

Modern Day (Large) Language Models

- LLMs are much more than mere “predict next token” machines these days.
- Post-training:
 - Instruction tuning
 - Reasoning
 - RHLF/DPO
 - RLVR
 - Tool use, vision capabilities...



Supervised Fine Tuning (SFT)

- **What?** Create clear examples and train the model to maximize the probability of the assistant's response given the prompt.
- **Why?** Because pretraining teaches a model to continue text, not to follow instructions.

User

~

Give three tips for staying healthy.

Assistant

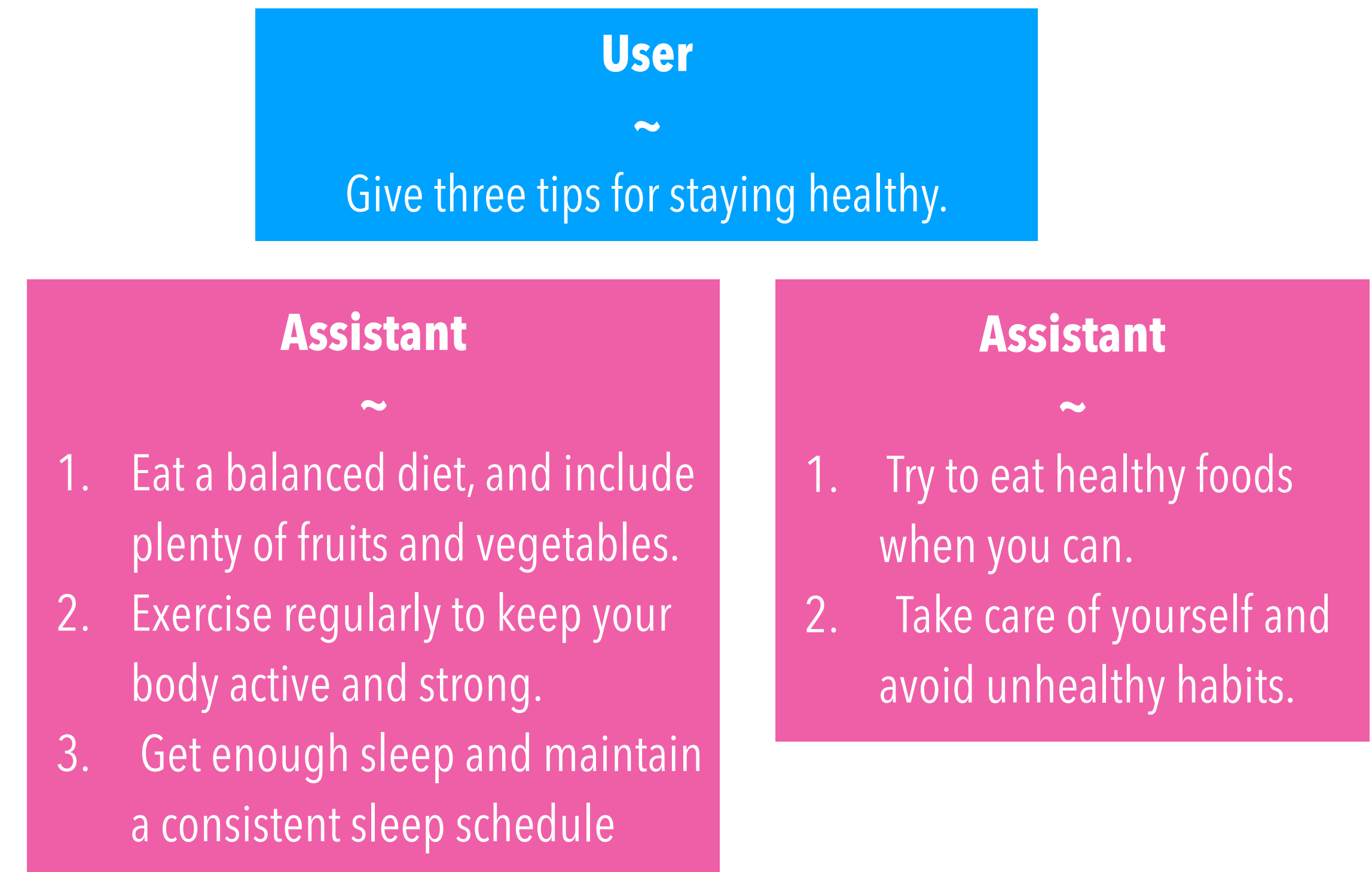
~

1. Eat a balanced diet, and include plenty of fruits and vegetables.
2. Exercise regularly to keep your body active and strong.
3. Get enough sleep and maintain a consistent sleep schedule

“Here is what a good answer looks like. Imitate it.”

Direct Preference Optimization (DPO)

- **What?** Have humans compare multiple model responses; train the model to produce responses that humans prefer.
- **Why?** Because pretraining teaches a model to continue text, not to follow instructions.



“Here are several answers. This one is better. Learn to produce answers like that.”

Chain-of-Thought

- **What?** Instead of producing an answer immediately, have the model generate intermediate reasoning steps. (Originally a prompting trick.)
- **Why?** This can substantially improve performance on tasks requiring multi-step reasoning (e.g., math, logic, planning).

User

~

Roger has 5 tennis balls. He buys 2 cans with 3 balls each. How many does he have?

Assistant

~

<thinking>

2 cans times 3 = 6

6 + 5 = 11

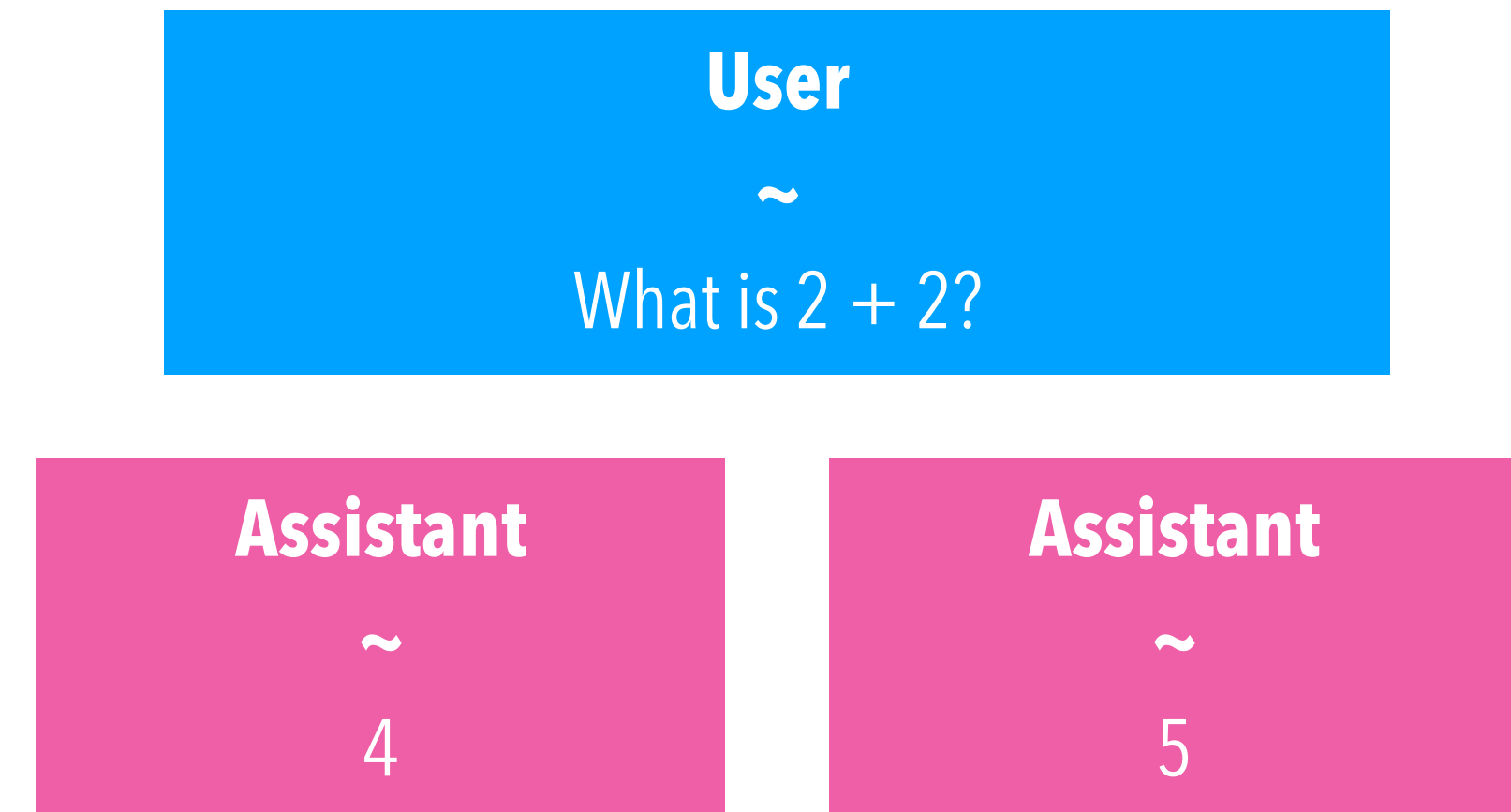
</thinking>

<answer>11</answer>

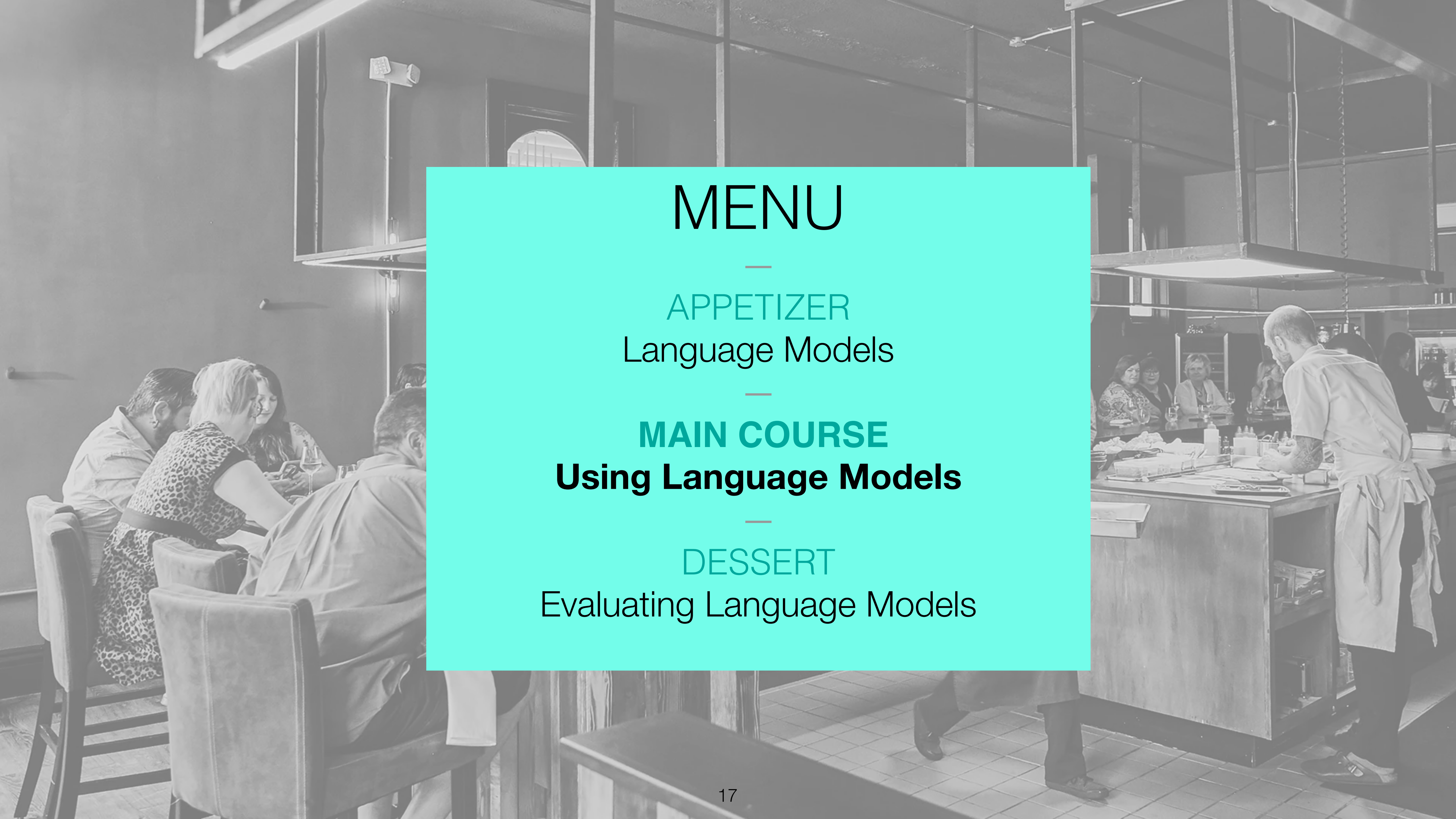
Instead of using compute during training, we can use “test-time compute”!

RL with Verifiable Rewards (RLVR)

- **What?** Train the model with reinforcement learning on whether it got the answers correct.
- **Why?** For tasks like math, we can often check the final answer automatically, allowing training without human labels!



“Try solving the problem. If you get the right answer, you get a reward.”



MENU

APPETIZER

Language Models

MAIN COURSE

Using Language Models

DESSERT

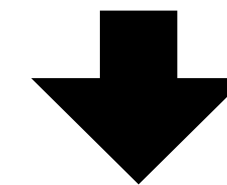
Evaluating Language Models

LLMs created a new paradigm

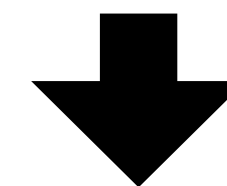
- LLMs perform very well in a broad range of tasks!
 - Classification
 - Extraction
 - Summarization
 - Translation
 - Question answering

Before LLMs

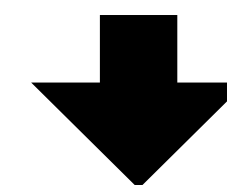
Data + Labels



Train Model



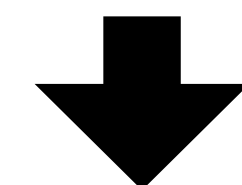
Use Model



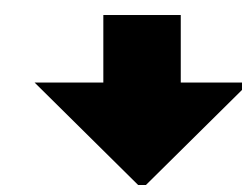
Predictions

After LLMs

Instructions +
Unlabeled data



LLM



Predictions

The OG way: prompting as completion

- Pretrained LMs are trained to predict the next token.
- Prompt the model by turning your task into a completion!
- “**Zero-shot:**” describe the task and let the model complete the answer

Question answering

Question: What is the capital of Brazil?

Answer:

Translation

English: The weather is beautiful today.

French:

Classify each review as **Positive** or **Negative**.

Review: "A wonderful movie!"

Sentiment:

Zero-shot vs. Few-shot prompting

Classify each review as **Positive** or **Negative**.

Review: "A wonderful movie!"

Sentiment:



Classify each review as **Positive** or **Negative**.

Review: "I loved this movie!"

Sentiment: Positive

Review: "A complete waste of time."

Sentiment: Negative

Review: "Absolutely fantastic."

Sentiment: Positive

Review: "I could barely stay awake."

Sentiment: Negative

Review: "A wonderful movie!"

Sentiment:

- Examples help the model infer the task, the format, and the labels used.
- No model parameters are updated — learning happens “in context”

Modern prompts

User

~

Task: Analyze the following social media post.

For a post, classify:

Sentiment: Positive / Neutral / Negative

Emotion: Joy / Anger / Sadness / Fear / None

Topic: Politics / Sports / Entertainment / Technology / Other

Toxicity: Toxic / Not toxic

Stance toward AI: Support / Oppose / Neutral / Not applicable

Return your answer as JSON.

Example input: "I can't believe they're replacing even more workers with AI. This is getting ridiculous"

Example output:

```
{  
  "sentiment": "Negative",  
  "emotion": "Anger",  
  "topic": "Technology",  
  "toxicity": "Not toxic",  
  "stance_ai": "Oppose"  
}
```

Now classify: "The new AI assistant saved me hours of work today"

Getting Predictions From LLMs

- LLMs produce *probabilities* over tokens!
- Two common ways to obtain a prediction:
 - **Generate:** ask the model to produce the output label and parse the response.
 - **Score:** use the probability the model assigns to each label.

User
~
What is 2 + 2?

Assistant
~
5

$$P(4 \mid 2+2=?) = 0.6$$

$$P(5 \mid 2+2=?) = 0.3$$

...

Getting Predictions From LLMs

- LLMs produce *probabilities* over tokens!
- Two common ways to obtain a prediction:
 - **Generate:** ask the model to produce the output label and parse the response.
 - **Score:** use the probability the model assigns to each label.

User
~
What is 2 + 2?

Assistant
~
5

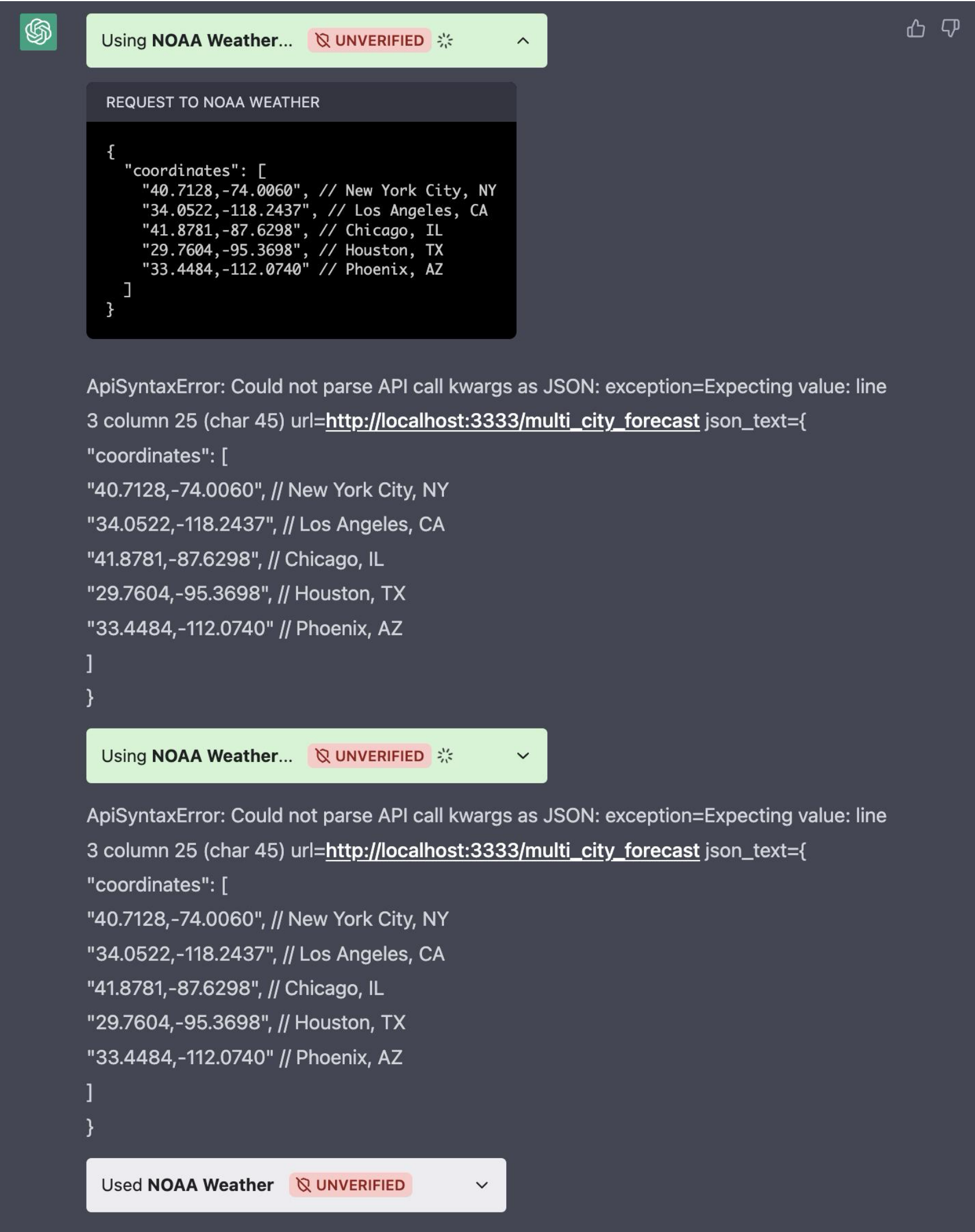
$$P(4 \mid 2+2=?) = 0.6$$

$$P(5 \mid 2+2=?) = 0.3$$

...

Challenge #1: Output Formatting

- LLMs generate free-form text
- Data pipelines need predictable, machine-readable outputs
- Prompting alone does not guarantee the requested format
- **Common problems:** extra prose, invalid JSON, missing fields, wrong labels



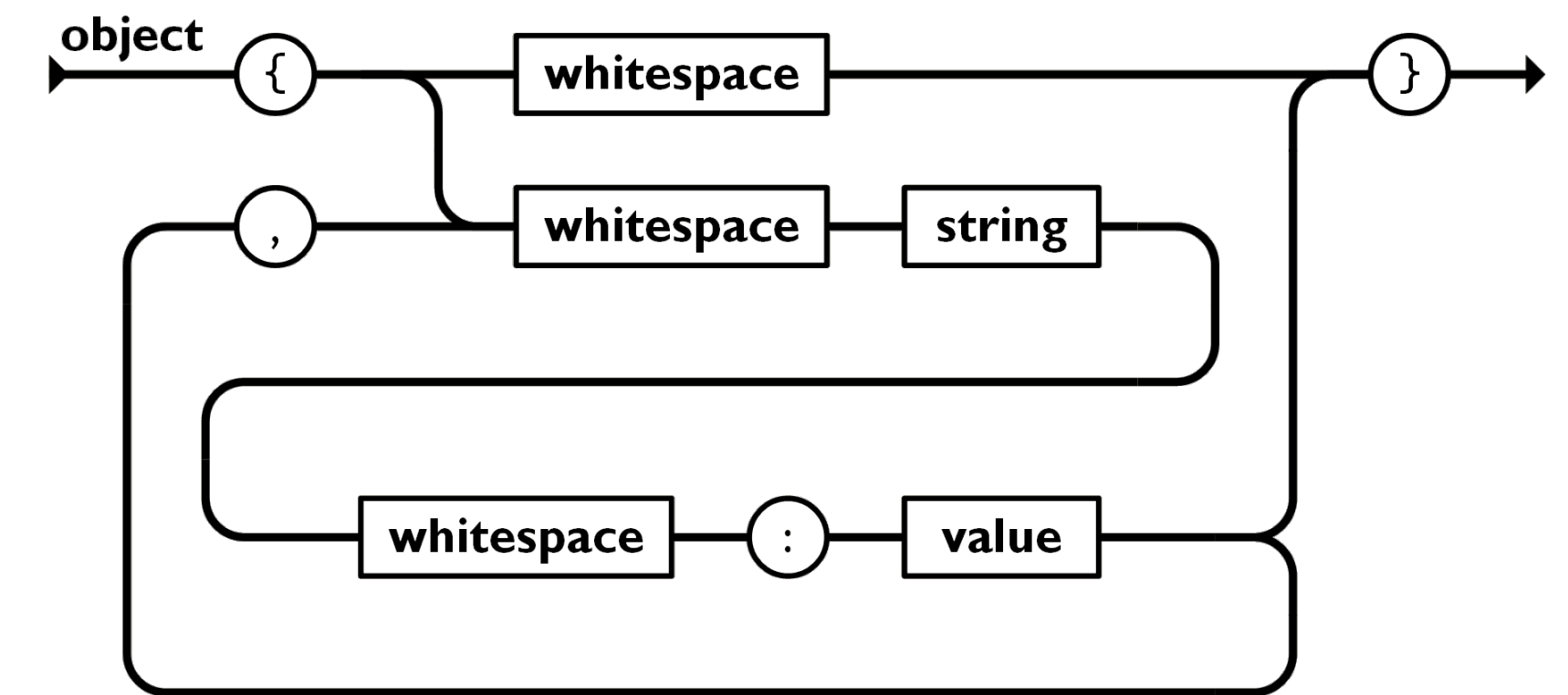
The screenshot shows a chat interface with a dark background. At the top, there is a green header bar with the OpenAI logo on the left, the text "Using NOAA Weather..." in the center, a red "UNVERIFIED" label with a magnifying glass icon, and a small upward arrow on the right. Below this, a dark box contains a JSON object labeled "REQUEST TO NOAA WEATHER". The JSON is as follows:

```
{
  "coordinates": [
    "40.7128,-74.0060", // New York City, NY
    "34.0522,-118.2437", // Los Angeles, CA
    "41.8781,-87.6298", // Chicago, IL
    "29.7604,-95.3698", // Houston, TX
    "33.4484,-112.0740" // Phoenix, AZ
  ]
}
```

Below the JSON, an error message is displayed: "ApiSyntaxError: Could not parse API call kwargs as JSON: exception=Expecting value: line 3 column 25 (char 45) url=[http://localhost:3333/multi_city_forecast](\"http://localhost:3333/multi_city_forecast\") json_text={". The error message is repeated twice. At the bottom, there is another green header bar with the text "Used NOAA Weather", a red "UNVERIFIED" label, and a small downward arrow.

Solution: Constrained Decoding

- Sometimes we can verify which outputs are valid!
 - Only a few tokens are valid (Yes/No)
 - We can specify a grammar that only generates valid outputs.
- Idea: use this information to “mask” which tokens are valid!
- Guarantees structure, not correctness!



Assistant

2

```
{ "sentiment":
```

Token	Probability	Masked
positive	.60	.71
negative	.25	.29
I think	.08	—
Positive	.04	—
...	...	...

Getting Predictions From LLMs

- LLMs produce *probabilities* over tokens!
- Two common ways to obtain a prediction:
 - **Generate:** ask the model to produce the output label and parse the response.
 - **Score:** use the probability the model assigns to each label.

User
~
What is 2 + 2?

Assistant
~
5

$$P(4 \mid 2+2=?) = 0.6$$

$$P(5 \mid 2+2=?) = 0.3$$

...

Challenge #2: Tokens $\neq$ Labels

- LLMs output probabilities over tokens, not labels.
- A label may correspond to one or multiple tokens.
- Tokenization can therefore affect seemingly simple classification tasks.

User

~

How positive is this review?
"It was pretty good!"

Labels: Very Positive / Positive /
Negative / Very Negative

Token	Probability
Positive	.30
Very	.40
Negative	.05
I think	.04
...	...

Solutions

- Ensure the labels are a single token.
- Compute the joint probability of all tokens in each candidate label
- Ask for the label probabilities.
- **Caveat:** Post-training distorts calibration!

User
~
How positive is this review?
"It was pretty good!"

Labels:
A = Very positive
B = Somewhat positive
C = Somewhat negative
D = Very negative

$$P(\text{Very positive}) = P(\text{Very}) \times P(\text{positive} \mid \text{Very})$$

Prompting is surprisingly brittle

- Small changes to a prompt can cause large changes in performance
- Example selection and ordering can matter
- Effects vary across tasks and models

[...] 88.5% accuracy

Review: the whole thing's fairly lame, making it par for the course for disney sequels.

Answer: Negative

Review: this quiet, introspective and entertaining independent is worth seeking.

Answer: Positive

[...] 51.3% accuracy

Review: this quiet, introspective and entertaining independent is worth seeking.

Answer: Positive

Review: the whole thing's fairly lame, making it par for the course for Disney sequels.

Answer: Negative

Prompting is surprisingly brittle

- Modern models are better...
- ... but prompt sensitivity remains an important source of measurement variance.
- **Best practice:** evaluate across prompts.
- **Prompt engineering:** find the best prompt for your task!

[...] 88.5% accuracy

Review: the whole thing's fairly lame, making it par for the course for disney sequels.

Answer: Negative

Review: this quiet, introspective and entertaining independent is worth seeking.

Answer: Positive

[...] 51.3% accuracy

Review: this quiet, introspective and entertaining independent is worth seeking.

Answer: Positive

Review: the whole thing's fairly lame, making it par for the course for Disney sequels.

Answer: Negative

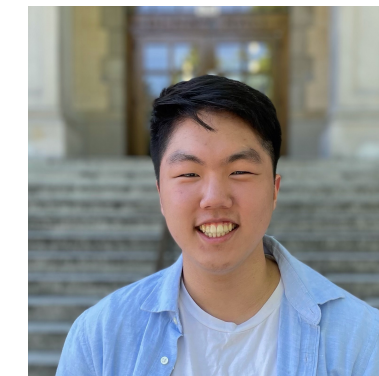
Evaluation ideas from ML hold!

- Create a development set for designing and iterating your prompt!
- Keep a held-out test set for final evaluation.
- Compare prompt variants using meaningful metrics!



But also, new problems emerge!

- LLMs are trained on massive amounts of Internet data
- Your questions and answers may be in the training data
- Models may “cheat” and perform well in unexpected ways!



Question

Question

Find studies

Synthesize
evidence

Answer

Find
systematic
review

Answer



MENU

APPETIZER

Language Models

MAIN COURSE

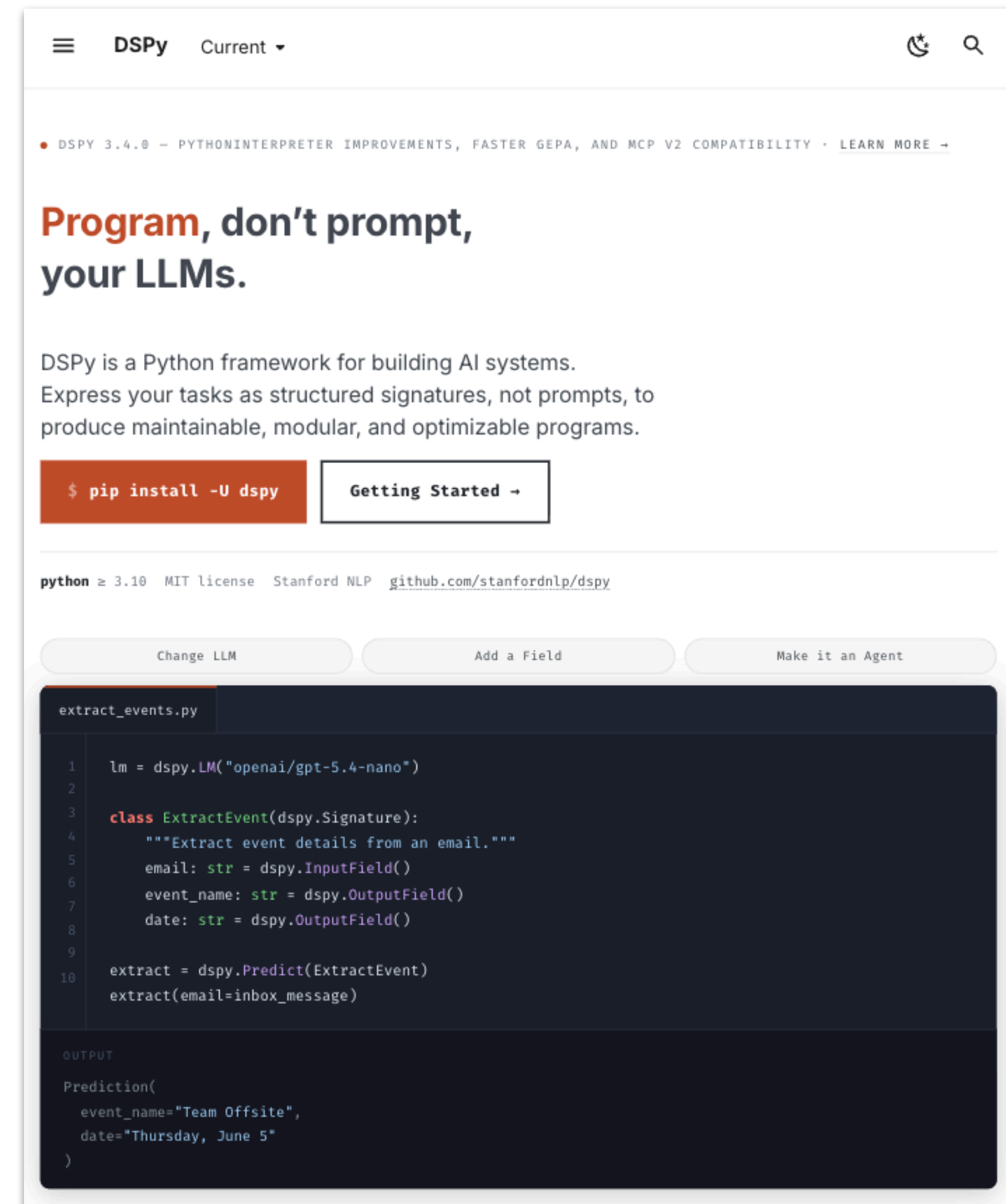
Using Language Models

DESSERT

Miscellanea

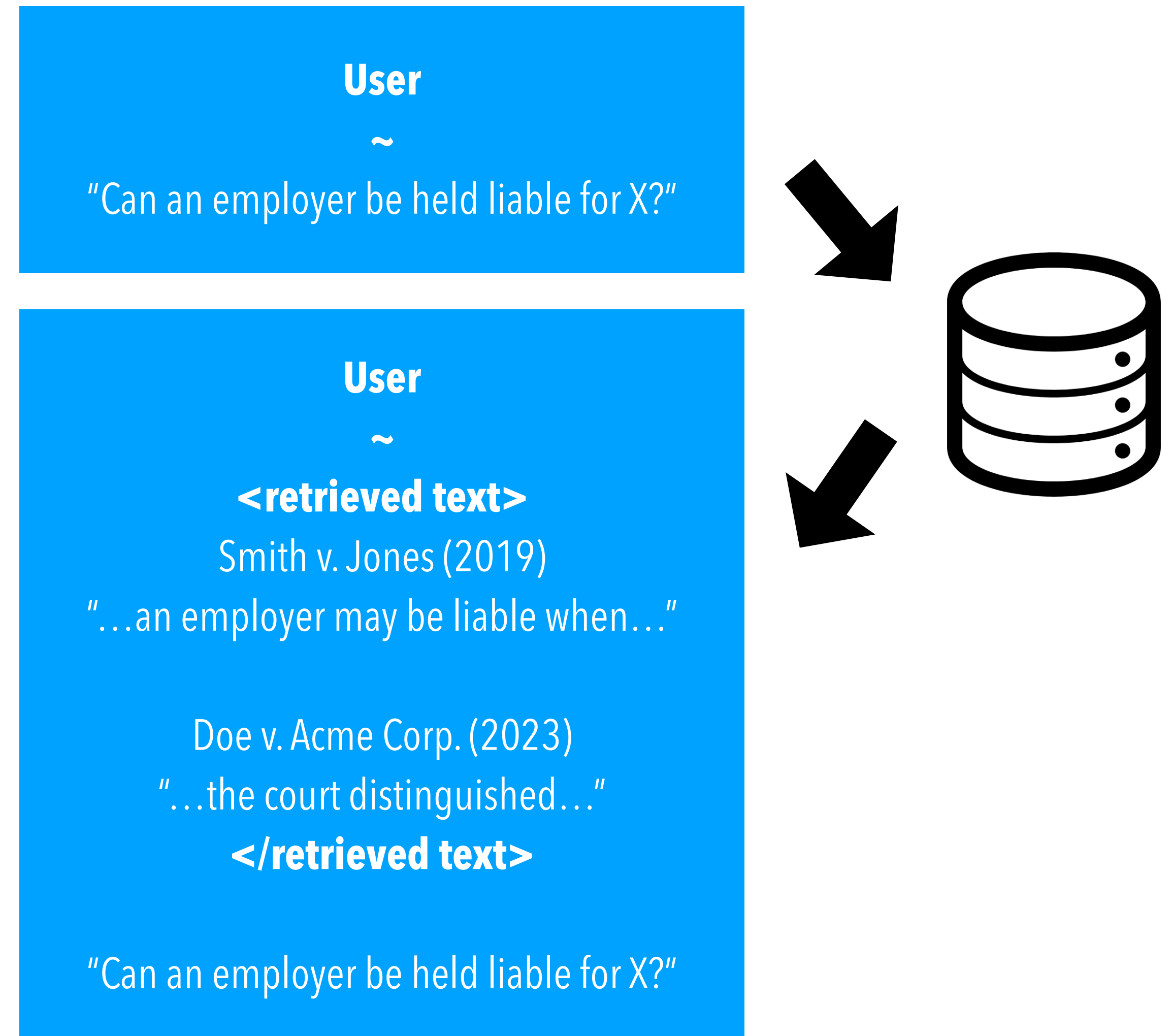
DSPy: Beyond Manual Prompts

- Programmatic way to specify what the LLM should do
- DSPy automatically searches for instructions and examples that improve performance
- Prompts as hyperparameters



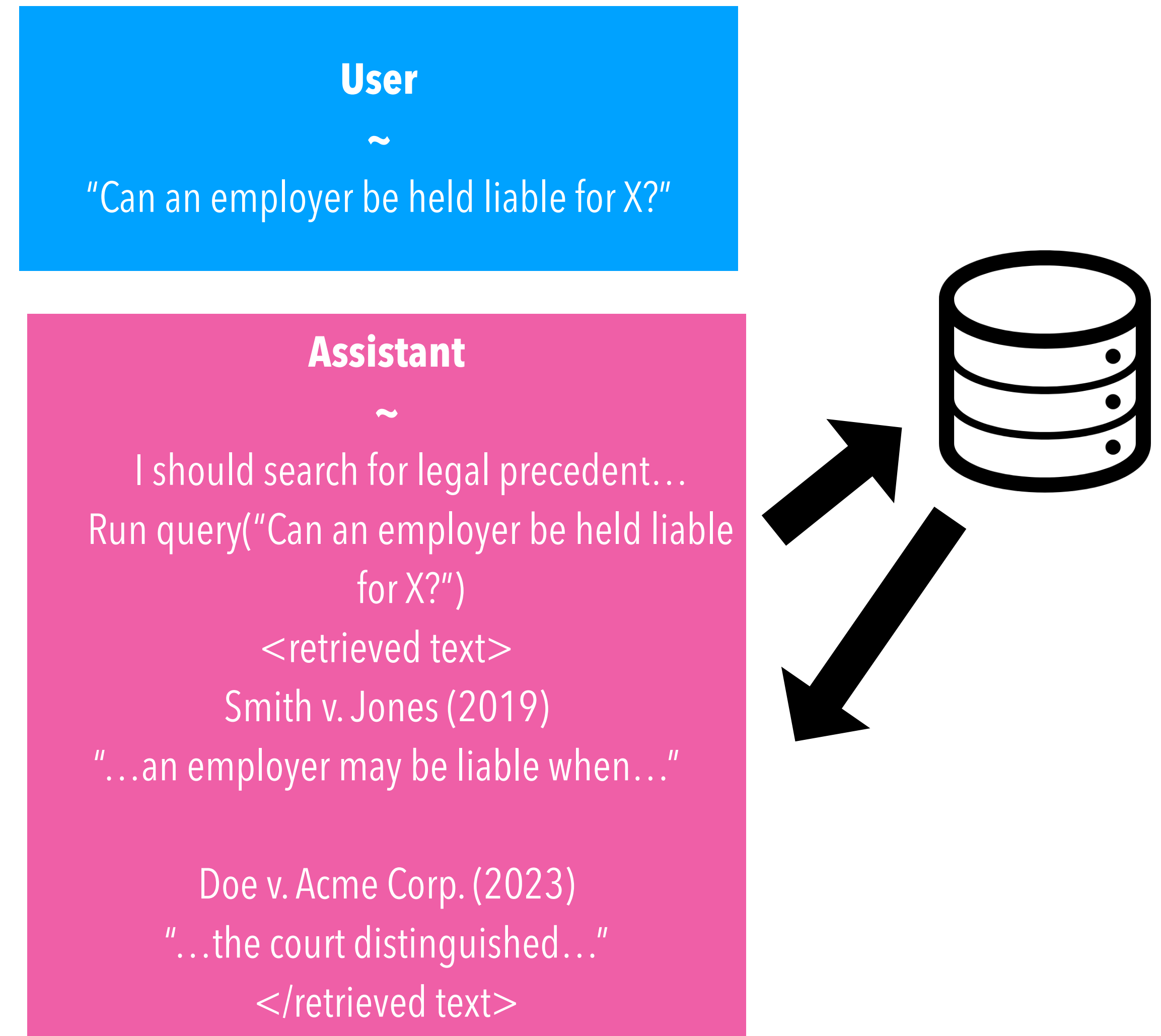
Retrieval Augmented Generation

- Search for information to add to the prompt.
- Typically, you have a separate database with relevant data!
- You may want to search in this database using embeddings!



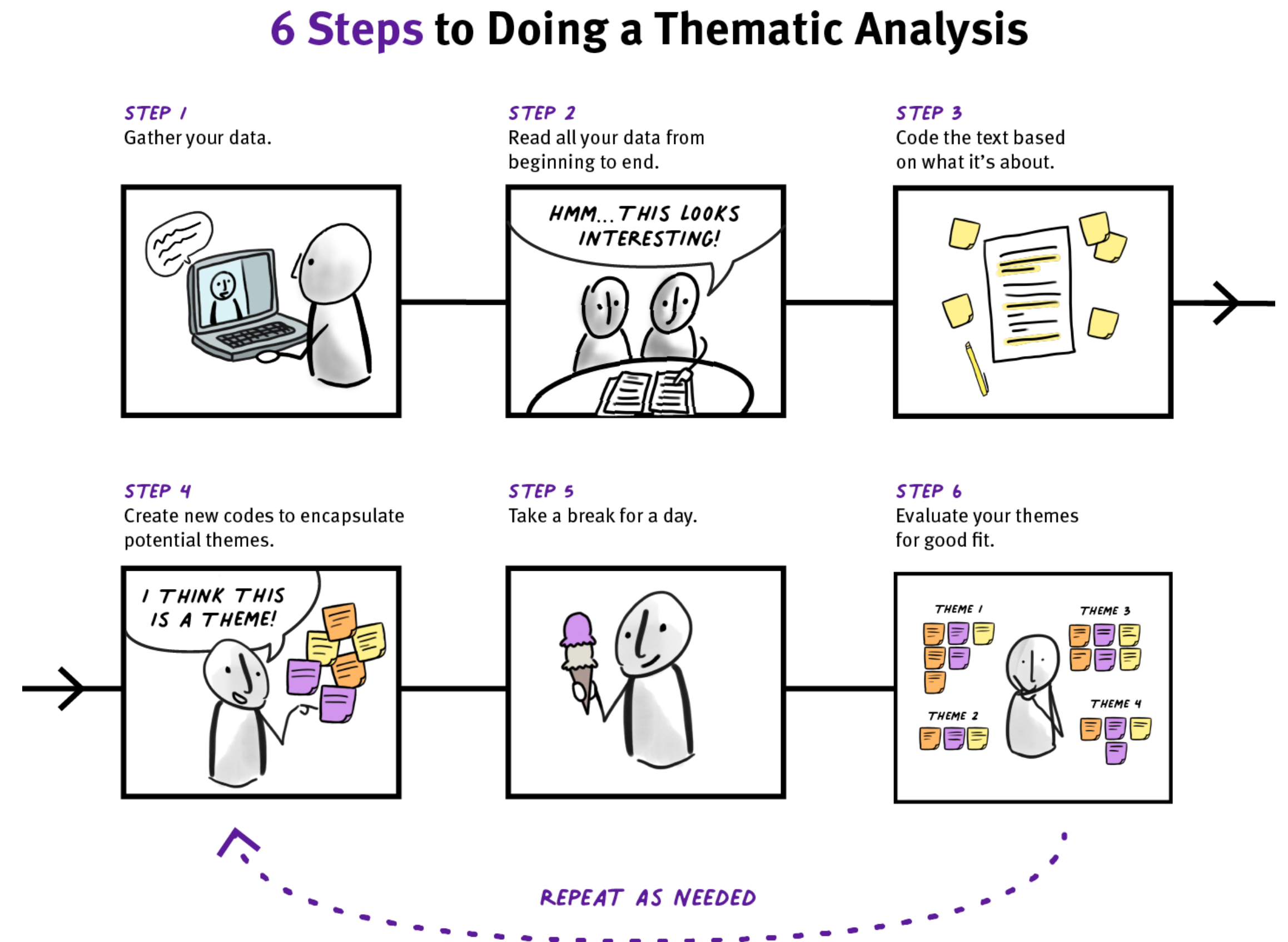
Agentic RAG

- **Alternative approach:** teach the LLM to use a search tool!
- Very common in, e.g., Claude Code.



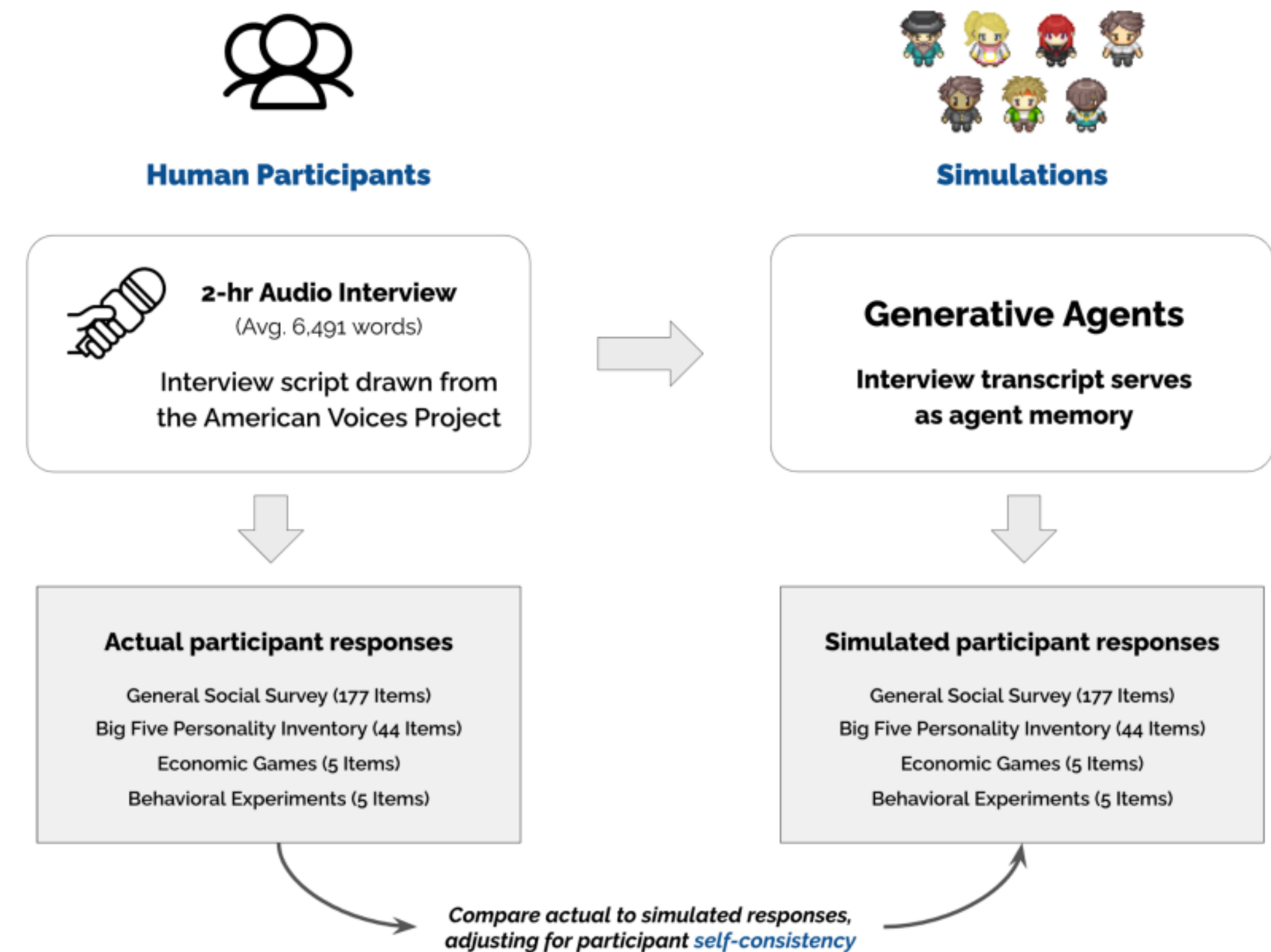
Qualitative Coding with LLMs

- LLMs can help **develop a codebook**, not just apply one
- E.g., give the model a sample of qualitative data and ask it to identify recurring themes
- Iteratively merge, split, and refine codes.



Simulating People with LLMs

- LLMs can be prompted to simulate how a particular person might respond...
- Given information about said person, e.g., demographics, attitudes, or interview transcripts
- Park et al. (2026): 83% of accuracy GSS questions





Give us some feedback!



[https://tinyurl.com/
reasoningwithdata](https://tinyurl.com/reasoningwithdata)