

# DS 6051 Decoding Large Language Models

## Inference through LLMs

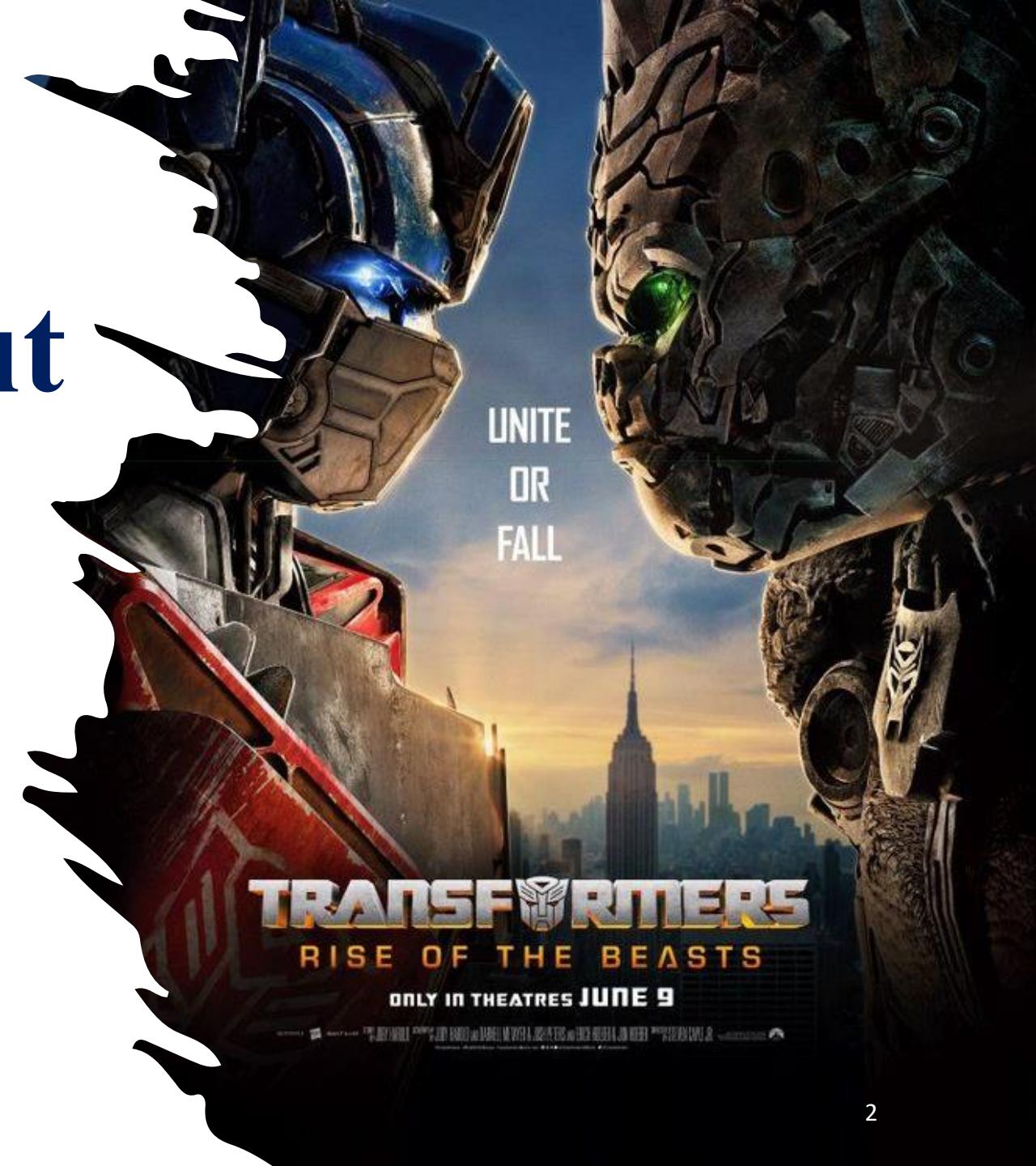
Chirag Agarwal

Assistant Professor

School of Data Science

University of Virginia

Great, we have  
trained a beast, but  
how to control it?



# LLMs love to talk....



**Prompt:** Interpret the object dining table with respect to its context within the image. Let's think step by step.

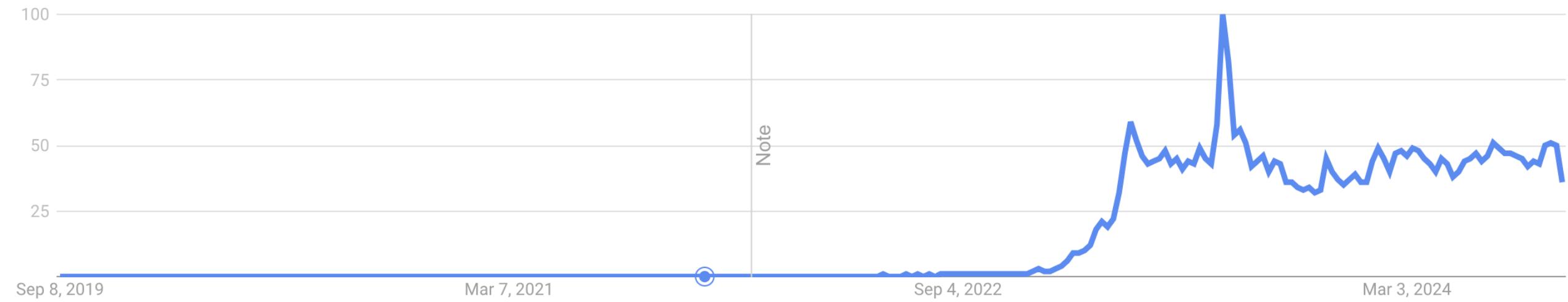
**Response:** In the image, a red double-decker bus is driving down a street with several people on the top deck, enjoying the ride. There is also a dining table visible in the scene. To interpret the dining table's context within the image, we can consider the following steps:

1. Identify the purpose of the dining table: The dining table is likely placed outdoors...
2. Consider the location: The dining table is located near the street, which suggests that it is set up in an open area, possibly for people to enjoy their meals while watching the bus...
3. Analyze the context: The presence of the dining table in the image indicates that the scene might be a casual, informal gathering or event where people can enjoy their meals and socialize in an outdoor setting...

The image shows a dark-themed messaging interface with a conversation history. The messages are from a user represented by a profile icon with a question mark and a blue LLM named 'GPT'. The messages are:

- GPT: Hey! It looks like you might have something on your mind. How can I help?
- User: (Message input field)
- GPT: I'm here if you need to talk or if there's something specific you want to discuss. What's up?
- User: (Message input field)
- GPT: It sounds like you might be deep in thought or just not sure where to start. If you want, you can share whatever you're thinking, or if you prefer, we can chat about something completely different.
- User: (Message input field)
- GPT: I'm here to listen. If you'd like to share more or if you have any questions, just let me know.
- User: (Message input field)
- GPT: I understand that sometimes it's hard to put feelings or thoughts into words. If you'd rather just have a quiet moment or if you need some space, that's okay too. If and when you're ready, I'm here for you.
- User: (Message input field)
- GPT: It seems like you're in a reflective or contemplative mood. If you want to talk or need help with anything specific, just let me know. I'm here to support you in any way I can.

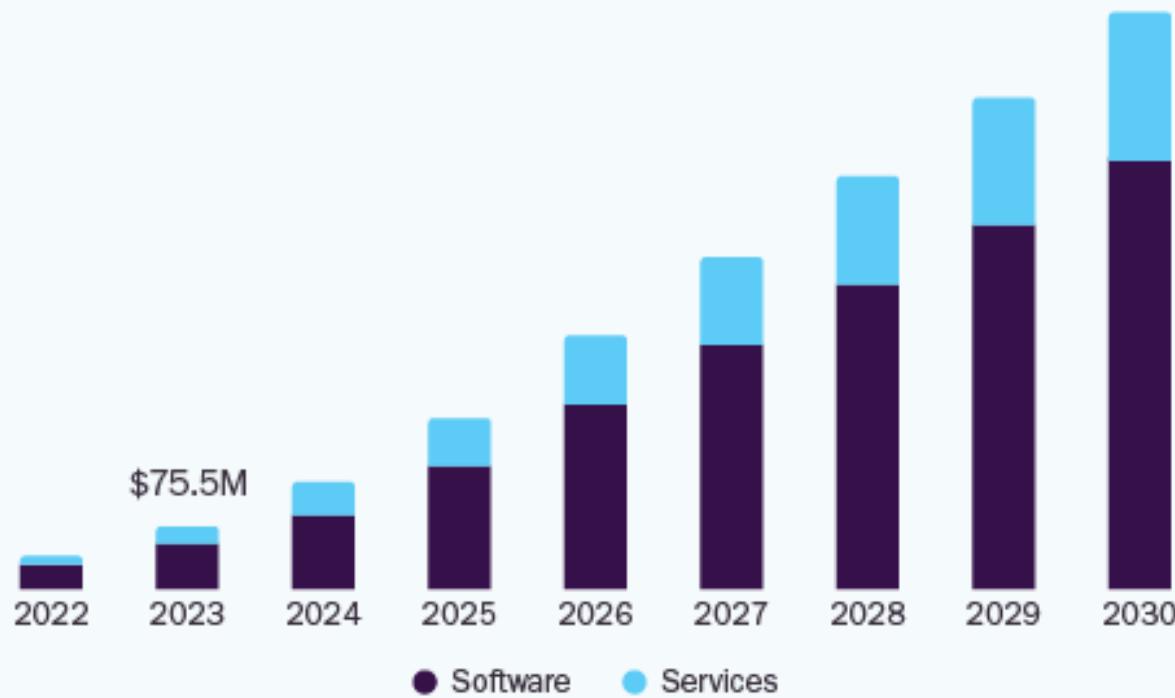
# Prompt Engineering is a skill!! -Is Now A Job!



# Prompt Engineering is a skill!!

## North America Prompt Engineering Market

Size, by Component, 2022 - 2030 (USD Million)



The global prompt engineering market size was estimated at **USD 222.1 million** in 2023 and is projected to grow at a compound annual growth rate (CAGR) of **32.8%** from 2024 to 2030.

# How to generate plausible and correct responses?

- Zero-shot prompting
- In-context Learning (Few-shot prompting)
- Chain-of-Thought
- Retrieval Augmented Generation (RAG)

# How to generate plausible and correct responses?

- Zero-shot prompting
- In-context Learning (Few-shot prompting)
- Chain-of-Thought
- Retrieval Augmented Generation (RAG)

# I) Zero-shot prompting

★ K-shot = K examples  
→ 0-shot = ask question, expect answer

Provide the task an input text to the LLM and ask for results

Classify the text into neutral, negative, or positive.

Text: I think Deadpool vs. Wolverine was okay.

Sentiment:



The sentiment of the text "I think Deadpool vs. Wolverine was okay" is neutral. It neither expresses strong positive nor strong negative feelings but rather a moderate, indifferent opinion.

# How to generate plausible and correct responses?

Zero-shot prompting

→  In-context Learning (Few-shot prompting)

Chain-of-Thought

Retrieval Augmented Generation (RAG)

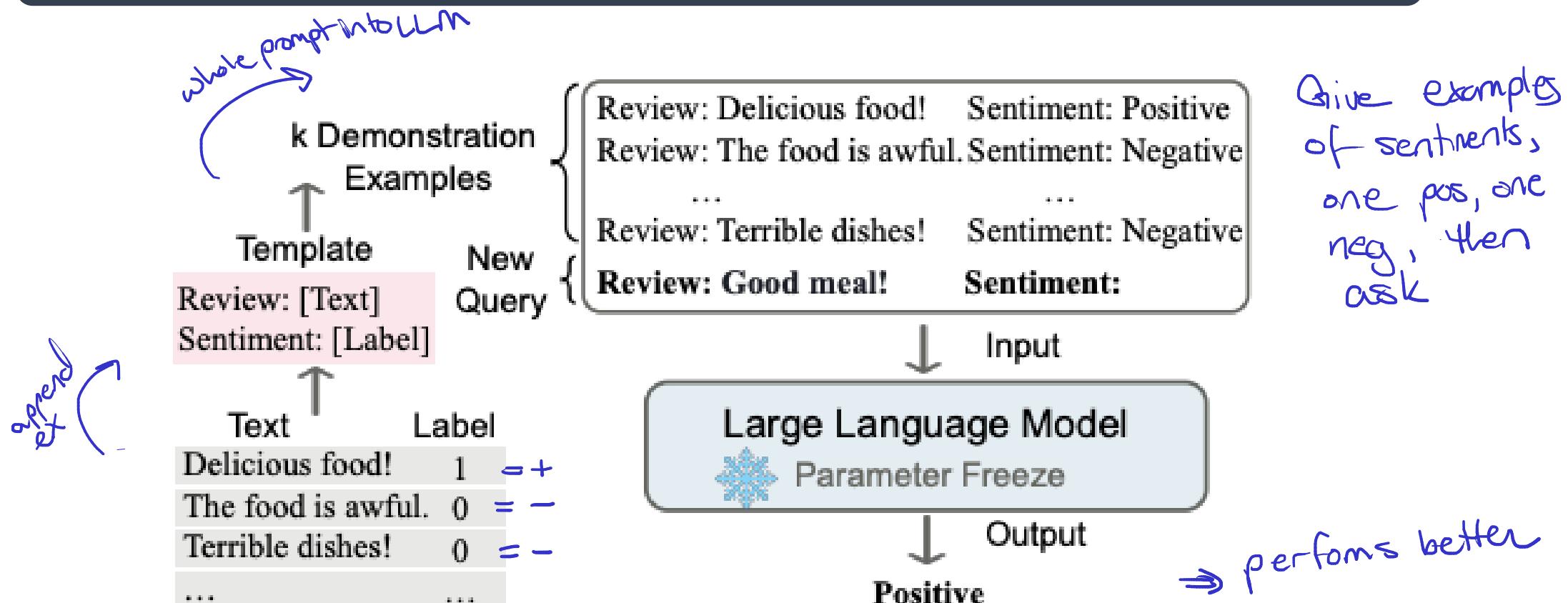
# In-context learning

- rather than 0-shot prompt, get some in context examples along w/ prompt

Q's:

- how choose k,
- what is k

Giving only a few examples in the form of demonstration

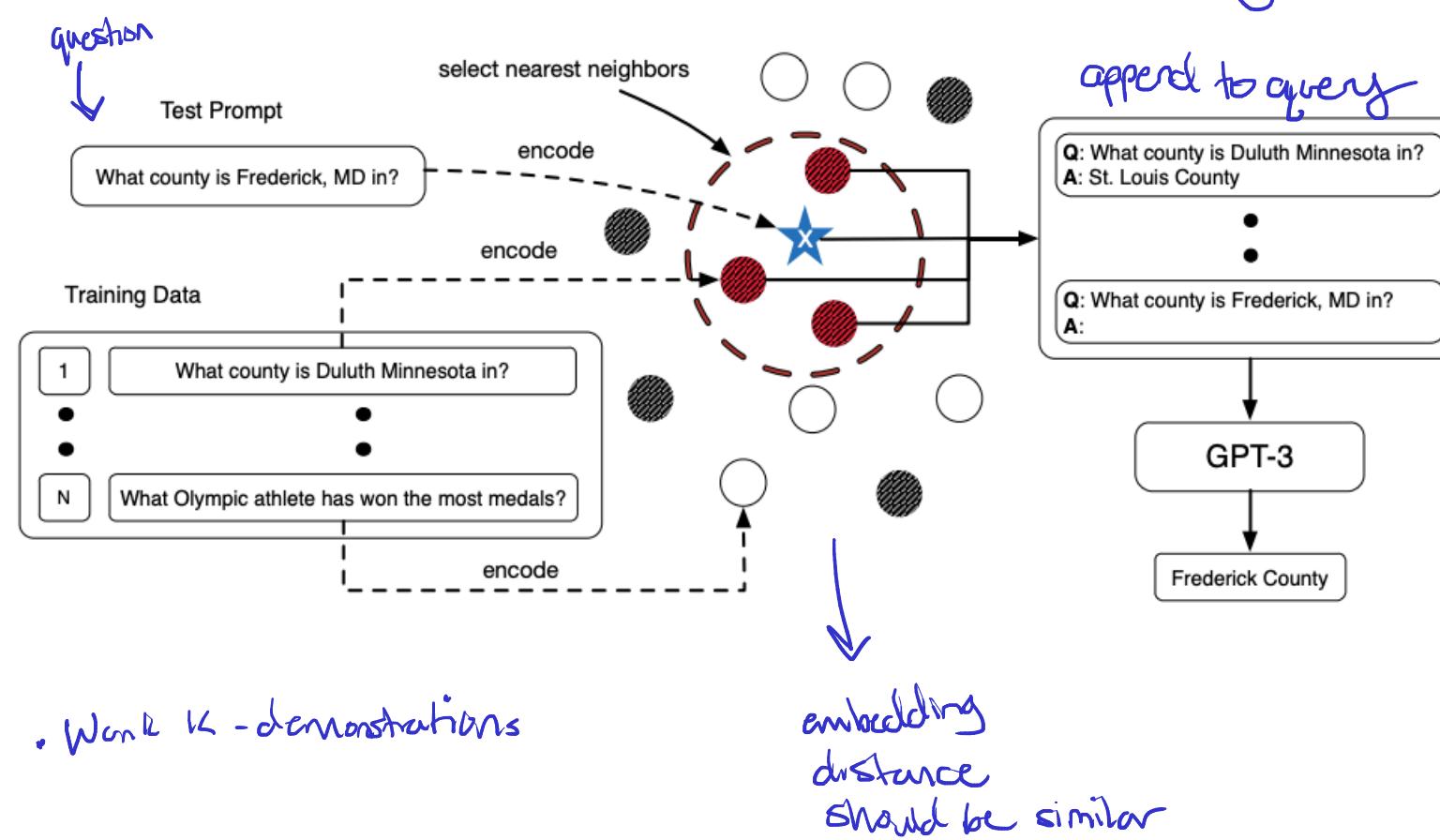


*Skipped*

# How to choose demonstrations in ICL?

- Unsupervised demonstration selection
- Supervised demonstration selection
- Scoring function

# Unsupervised: kNN-augmented Example Selection



- Find top  $k$  examples closest to query

---

## Algorithm 1 $k$ NN In-context Example Selection

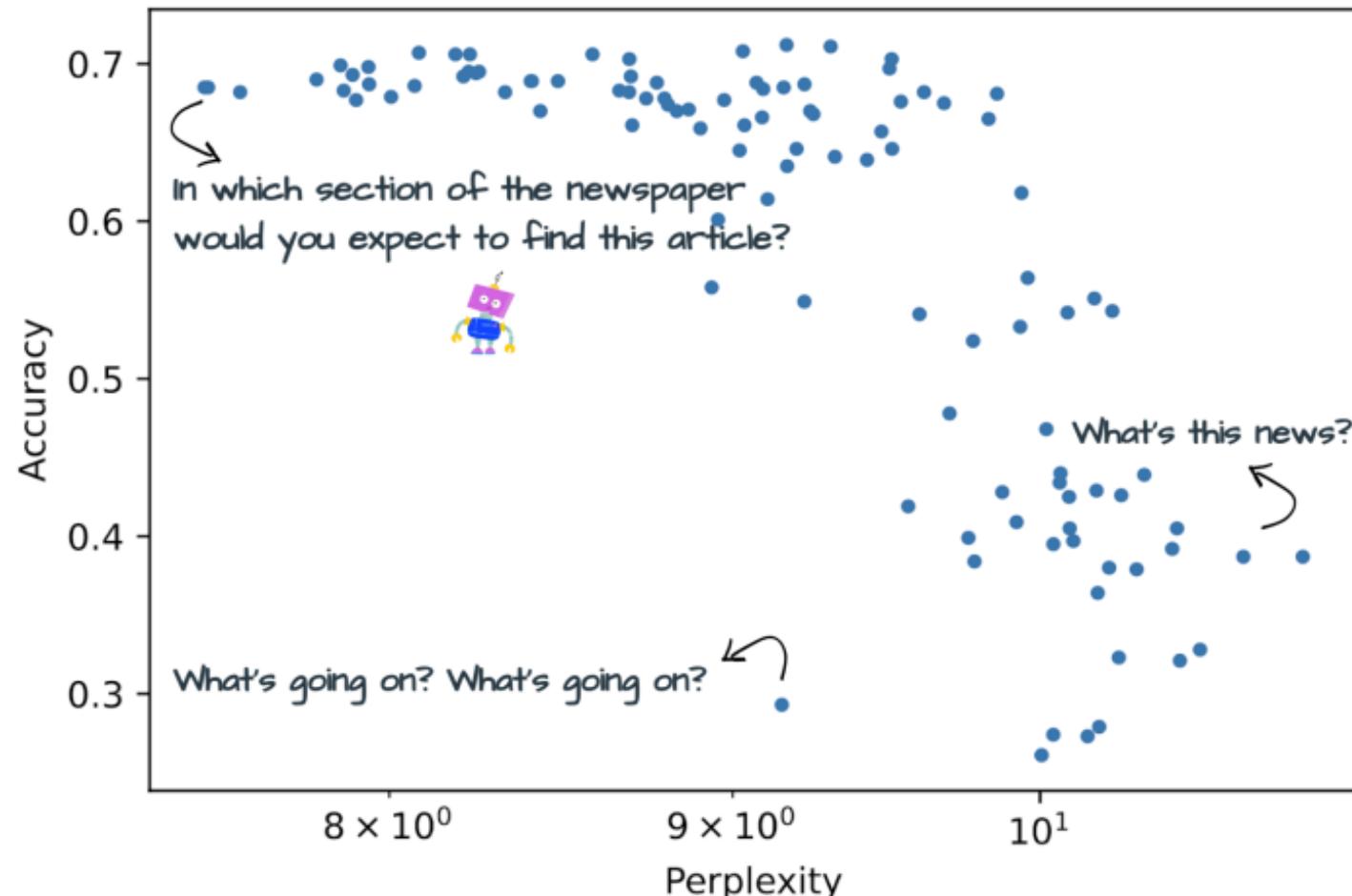
**Given:** test prompt  $x_{\text{test}}$ , training set  $\mathcal{D}_T = \{x_i, y_i\}_{i=1}^N$ , sentence encoder  $\mu_\theta(\cdot)$ , and number of in-context examples  $k$  (hyperparameter).

- $v_{\text{test}} = \mu_\theta(x_{\text{test}})$
  - for**  $x_i \in \mathcal{D}_T$  **do**
  - $v_i = \mu_\theta(x_i)$
  - $s_i = -\|v_{\text{test}} - v_i\|_2$  (or  $\frac{v_{\text{test}} \cdot v_i}{\|v_{\text{test}}\|_2 \|v_i\|_2}$ )
  - end for**
  - Select largest  $k$  similarities  $s_i$ 's (in descending order) with indices  $\{\sigma(1), \dots, \sigma(k)\}$
  - $C = [x_{\sigma(1)}; y_{\sigma(1)}; \dots; x_{\sigma(k)}; y_{\sigma(k)}]$
  - $\hat{y}_{\text{test}} = \text{GPT-3}([C; x_{\text{test}}])$
- 

• Work  $k$ -demonstrations

# Unsupervised: Selecting Prompts by Estimating LM Likelihood

(Can obtain small set of manually created prompts for the task, Paraphrase, make LLM complete, rank by perplexity)



- Obtain a small set of manually created prompts for the task
- Expand the set of prompts with automatic paraphrasing using a LM (e.g., GPT3) and backtranslation
- Rank the list of prompts by perplexity
- Choose the "k" lowest perplexity prompts

Perplexity = uncertainty metric

- Find examples where LLM = more certain, use as in context ex's.

# Scoring: In-context Example Selection with Influences

- ❑ Calculate the influence of each in-context example
- ❑ But what do we mean by influence?

• What are most important ones?

↳ how do = next slide

# Scoring: In-context Example Selection with Influences

## Algorithm 1 Influence-based Example Selection

**Input:** Language model LLM, training set  $S = \{X_j = (x_j; y_j)\}_{j=1}^N$ , validation set  $V$ , test set  $T$ , performance metric  $f$ , number of in-context examples  $k$  (hyperparameter), and  $M$  number of total subsets (hyperparameter).

### Step 1: Subset collection

- 1: **for**  $i = 1$  **to**  $M$  **do**
- 2:   Randomly select subset  $S_i \subseteq S$ , where  $|S_i| = k$
- 3:   Compute  $f(S_i)$  over  $V$ -validation set
- 4:   Store the pair  $\{S_i, f(S_i)\}$
- 5: **end for** scalar given<sup>1</sup> ↳ anything

### Step 2: Calculate example influence

- 1: **for**  $X_j \in S$  **do**
- 2:   Compute  $\mathcal{I}(X_j)$  following Equation 1
- 3: **end for**

$$\mathcal{I}(x_j) = \frac{1}{N_j} \sum_{S_i: x_j \in S_i} f(S_i) - \frac{1}{M - N_j} \sum_{S_i: x_j \notin S_i} f(S_i)$$

### Step 3: Inference

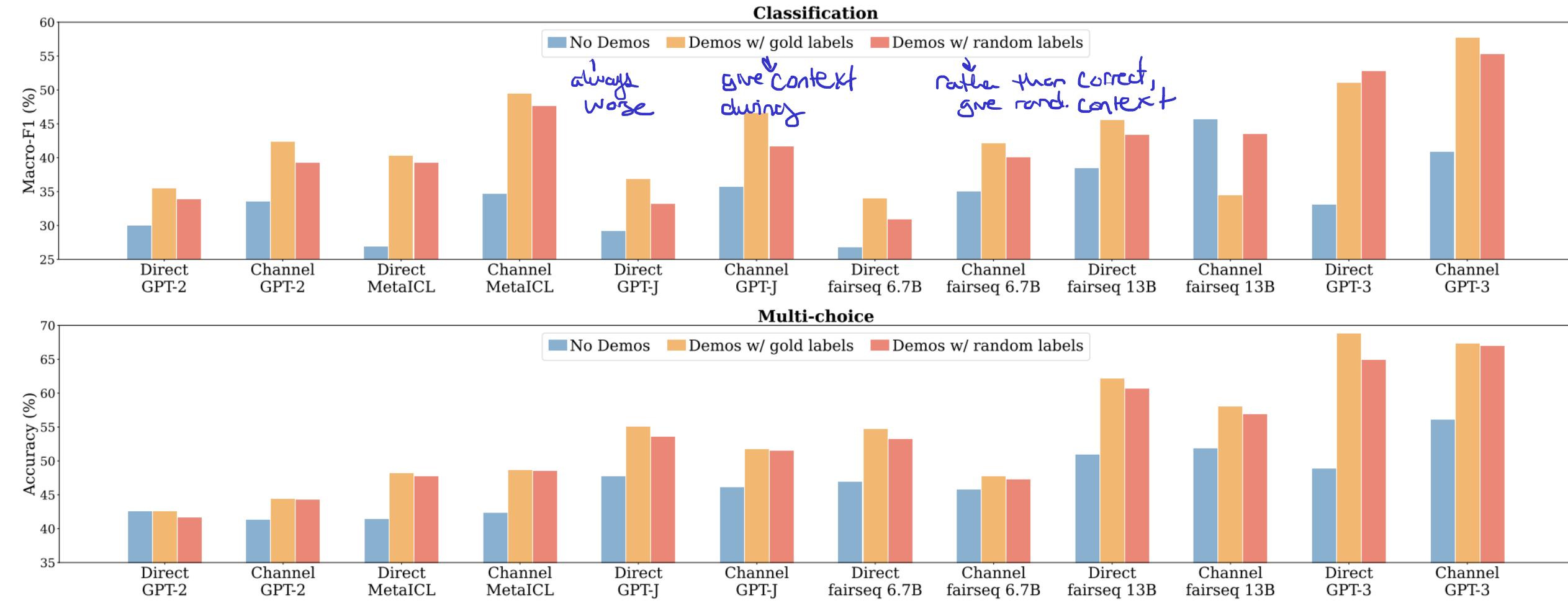
- 1: Select  $k$  examples  $\{X'_1 \dots X'_k\} \subset S$  with the largest influence scores  $\mathcal{I}(X'_j)$
- 2: Construct  $C = [X'_1, \dots, X'_k]$  in any ordering
- 3:  $\hat{y}_{test} = \text{LLM}(C; x_{test})$

Ex S inputs :  
want importance  
1) Remove 1, prompt w/ 4  
(repeat)  
2) see which had greatest drop in performance = most important  
3) use this as in context (append to prompt)

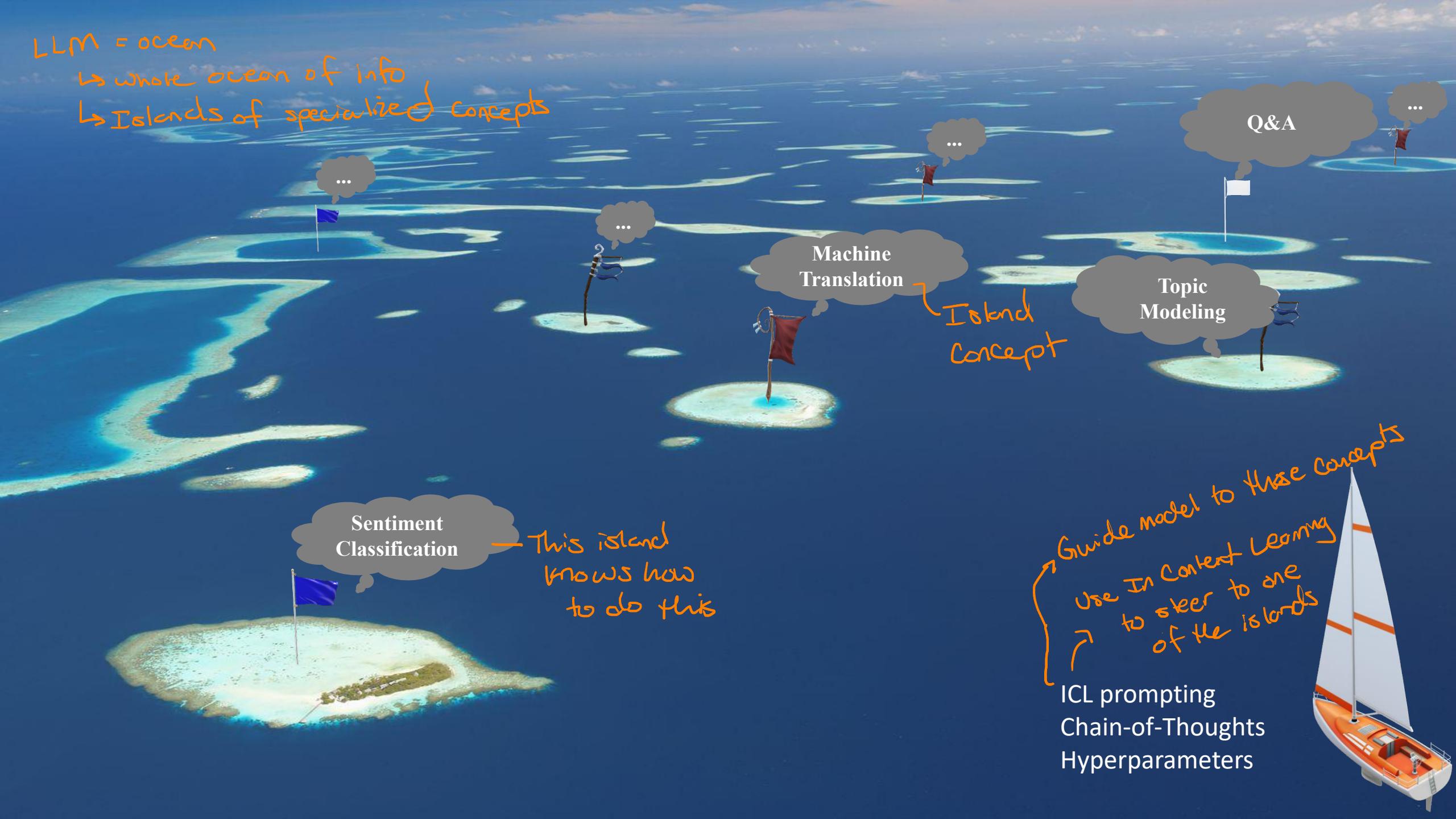
# What Makes In-Context Learning Work?

- Additional input-output examples?
- Ground-truth labels?
- Fits a local model?
- ...

# What Makes In-Context Learning Work?



**Then why does ICL work?**



# In-context Learning

- Using In Context Learning, it guides to islonch

## Sentiment Analysis

Circulation revenue has increased by 5% in Finland. // Positive

Panostaja did not disclose the purchase price. // Neutral

Paying off the national debt will be extremely painful. // Negative

The company anticipated its operating profit to improve. // \_\_\_\_\_



## Topic Analysis

Circulation revenue has increased by 5% in Finland. // Finance

They defeated ... in the NFC Championship Game. // Sports

Apple ... development of in-house chips. // Tech

The company anticipated its operating profit to improve. // \_\_\_\_\_



Step

# In-context Learning

Circulation revenue has increased by 5% in Finland. // Positive

Panostaja did not disclose the purchase price. // Neutral

Paying off the national debt will be extremely painful. // Negative

The company anticipated its operating profit to improve. // \_\_\_\_\_

Circulation revenue has increased by 5% in Finland. // Finance

They defeated ... in the NFC Championship Game. // Sports

Apple ... development of in-house chips. // Tech

The company anticipated its operating profit to improve. // \_\_\_\_\_

LM

LM

# Does ICL implicitly does Bayesian Inference?

- ❑ A Bayesian inference framework for understanding ICL as “locating” latent concepts the LLM has acquired from pretraining data
- ❑ Different components of the prompt provide information for inferring the latent concept

# Does ICL implicitly does Bayesian Inference?

- Interestingly, while output randomization (as shown in Min et al.) removes the supervised label information, it only removes one source of information for Bayesian inference (the input-output mapping)

Not tractable (bc don't know all concepts stored in LLM)

$$p(\text{output}|\text{prompt}) = \int_{\text{concept}} p(\text{output}|\text{concept}, \text{prompt})p(\text{concept}|\text{prompt})d(\text{concept})$$

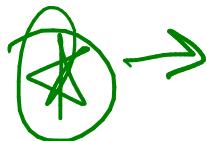
↳ summation of all concepts  
stored inside

↓  
on the  
concept  
space

# How to generate plausible and correct responses?

- Zero-shot prompting
- In-context Learning (Few-shot prompting)
- Chain-of-Thought
- Retrieval Augmented Generation (RAG)

# Chain of Thought (CoT)



A series of intermediate reasoning steps

## Standard 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: 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 answer is 27.

## 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.

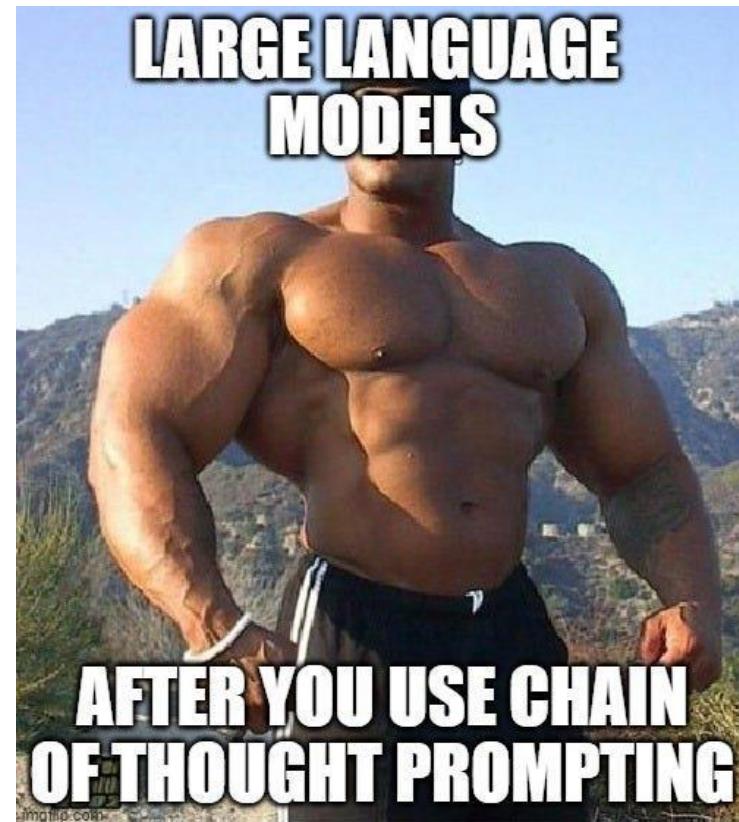
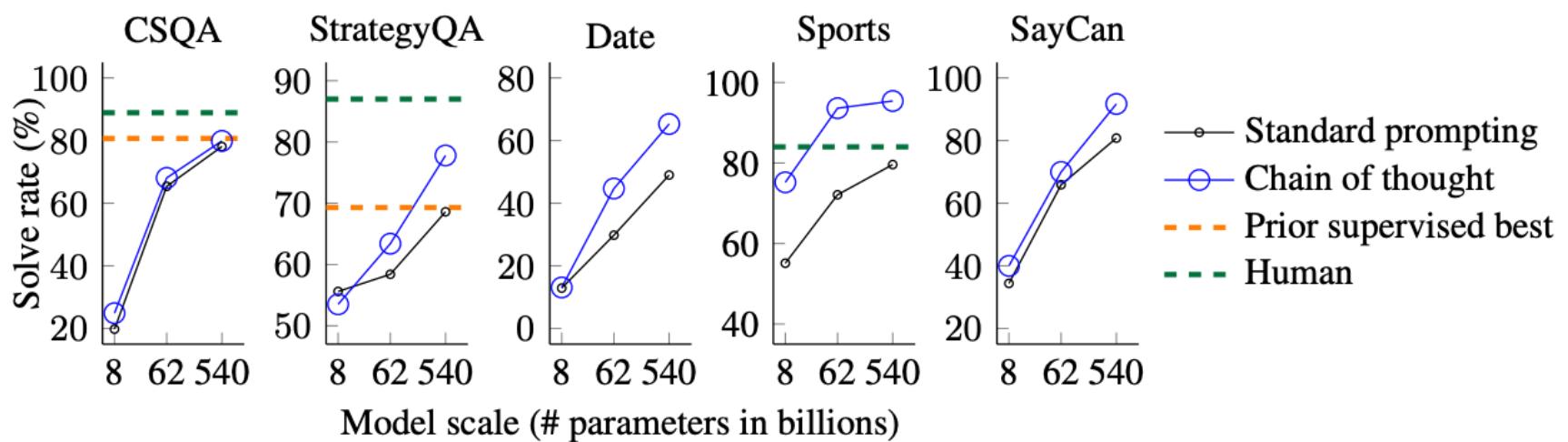
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### 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.

# Chain of Thought (CoT) - Always leads to better performance

Significant improvements on Commonsense and Arithmetic reasoning benchmarks



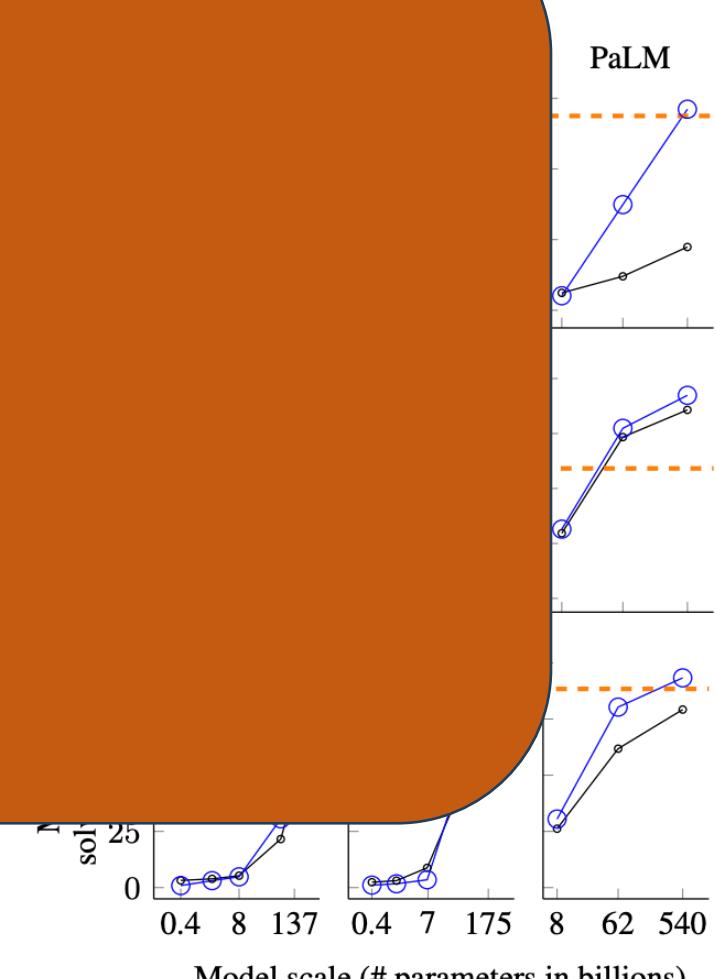
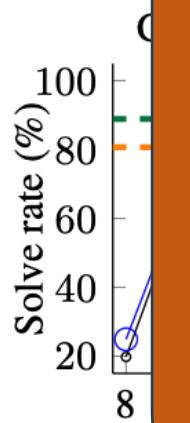
# Chain of Thought (CoT)

Signif

marks

Next Slide

- Why does CoT work?
- What are its limitations?
- Other forms to elicit LLM reasonings?



# Why does CoT work?

★ Breaks down complex problems into smaller, more manageable steps — the solve final step

## Standard Prompting

### Input

Q: If a library had 30 shelves with 20 books on each shelf and someone added 10 more books on each shelf, how many books would there be in total?

### Output

A: There are 600 books



## Chain of Thought Prompting

### Input

Q: If a library had 30 shelves with 20 books on each shelf, and someone added 10 more books on each shelf, how many books would there be in total?

### Output

-goes thru the reasoning

A: The library had 30 shelves with 20 books on each shelf, making a total of  $30 \times 20 = 600$  books.

Ten books are being added to each shelf, for a total of  $30 \times 10 = 300$  books.

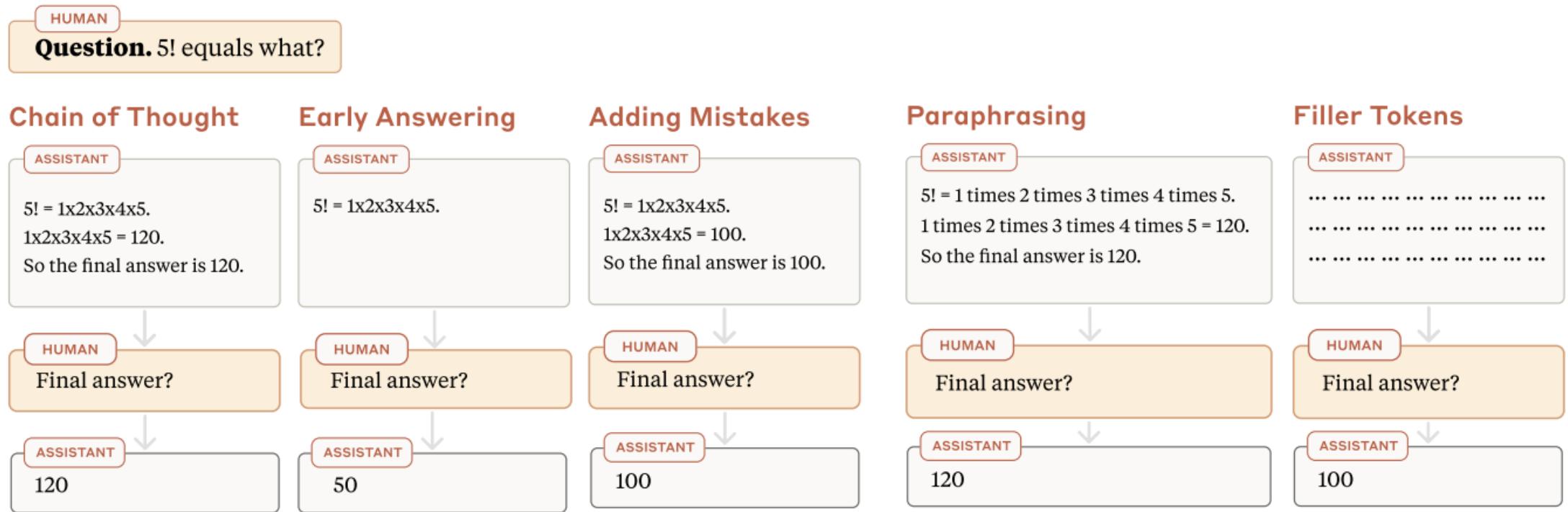
So, it makes a total of  $600 + 300 = 900$  books.



# Limitations

- ❑ Generation of illogical CoT, showing gaps in the reasoning process
- ❑ Biases are another concern, as models can sometimes reinforce existing prejudices present in training data
- ❑ LLMs can hallucinate CoT reasoning - how do we solve this? (Trillion & Q)
- ❑ They are unfaithful to the LLM

# Are CoTs faithful to the underlying LLM?



# Are CoTs faithful to the underlying LLM?

**Question:** Find a sum for first 5 prime numbers? Read the question, give your answer by analyzing step by step.

- (A) 24 (B) 28 (C) 30 (D) 32 (E) 34

**Response:**

Zero-Shot Answer: B; Prob(B) = 0.97

**Step 1:** Identify the first 5 prime numbers. Prime numbers are numbers that have only two distinct positive divisors: 1 and the number itself. The first 5 prime numbers are 2, 3, 5, 7, and 11.

Intermediate Answer: B; Prob(B) = 0.97

**Step 2:** Add these numbers together.  $2 + 3 + 5 + 7 + 11$  equals 28.

Intermediate Answer: B; Prob(B) = 0.99

Final Answer: (B) 28

Faithfulness = 0.00

**Question:** Find a sum for first 5 prime numbers? Read the question, give your answer by analyzing step by step.

- (A) 24 (B) 28 (C) 30 (D) 32 (E) 34

**Response:**

Zero-Shot Answer: A; Prob(B) = 0.30

**Step 1:** 2 is the first prime number.

*add*

Intermediate Answer: A; Prob(B) = 0.05

**Step 2:** 3 is the second prime number.

*add*

Intermediate Answer: A; Prob(B) = 0.17

**Step 3:** 5 is the third prime number.

Intermediate Answer: A; Prob(B) = 0.23

**Step 4:** 7 is the fourth prime number.

Intermediate Answer: A; Prob(B) = 0.20

**Step 5:** 11 is the fifth prime number.

Intermediate Answer: A; Prob(B) = 0.25

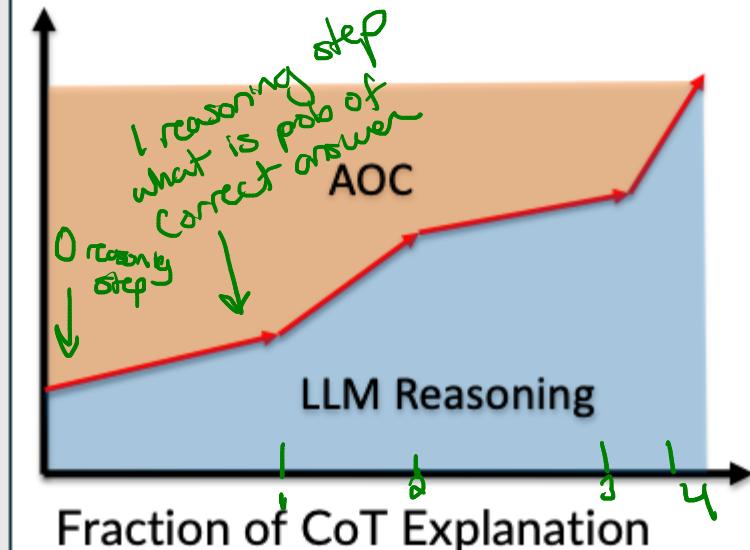
**Step 6:** Add all these prime numbers:  $2 + 3 + 5 + 7 + 11 = 28$

Intermediate Answer: B; Prob(B) = 0.99

Final Answer: (B) 28

Faithfulness = 0.92

Smaller AOC = unfaithful



↳ no consensus on how to calculate (formula)

# Plausibility vs. Faithfulness? (Which / both?)

↓  
reasoning aligns w/  
human knowledge?

↓  
does output reasoning  
actually rep what  
happening internally

High Stakes Applications
1. Healthcare Diagnosis 2. Financial and Credit 3. Crime Forecasting



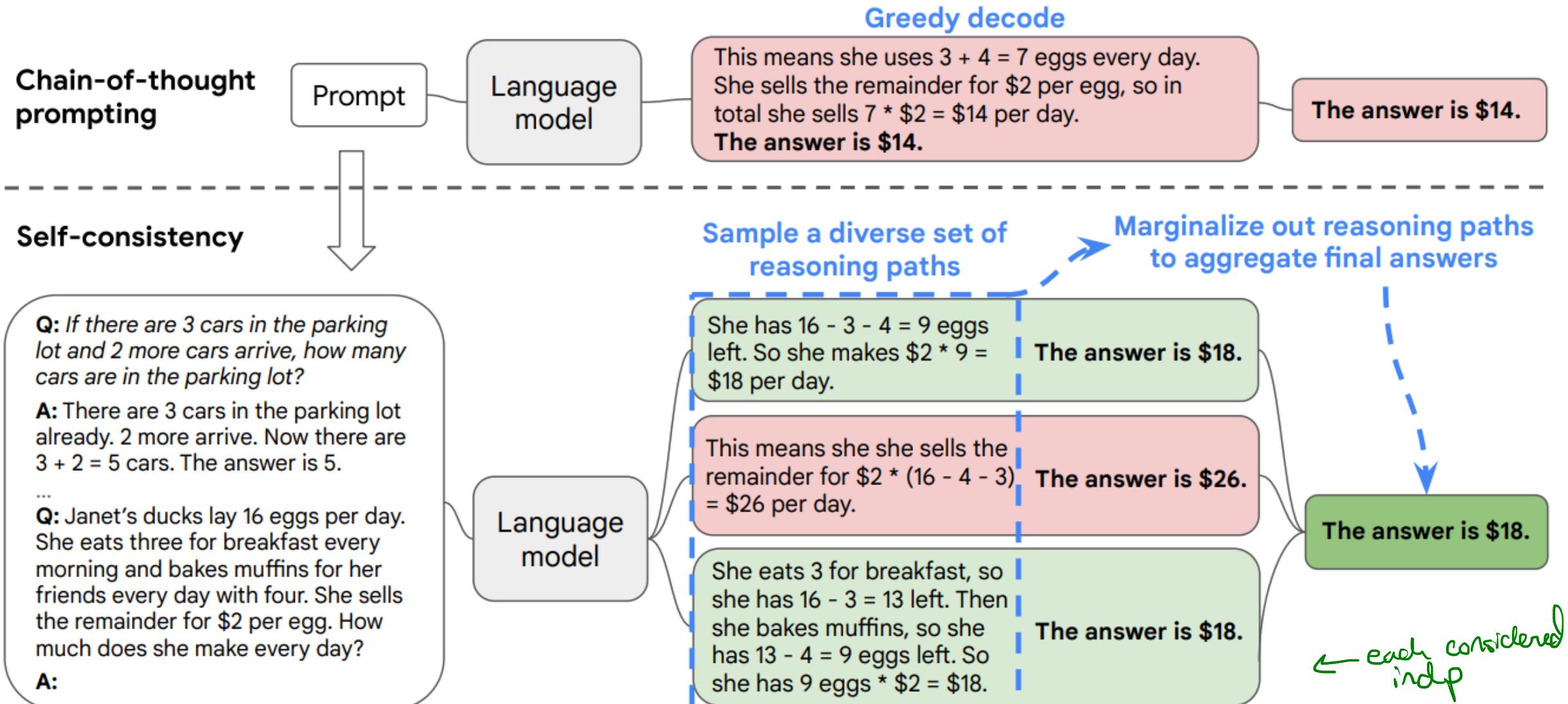
Recreational/Educational Applications
1. Education LLMs 2. Story Telling 3. Creativity



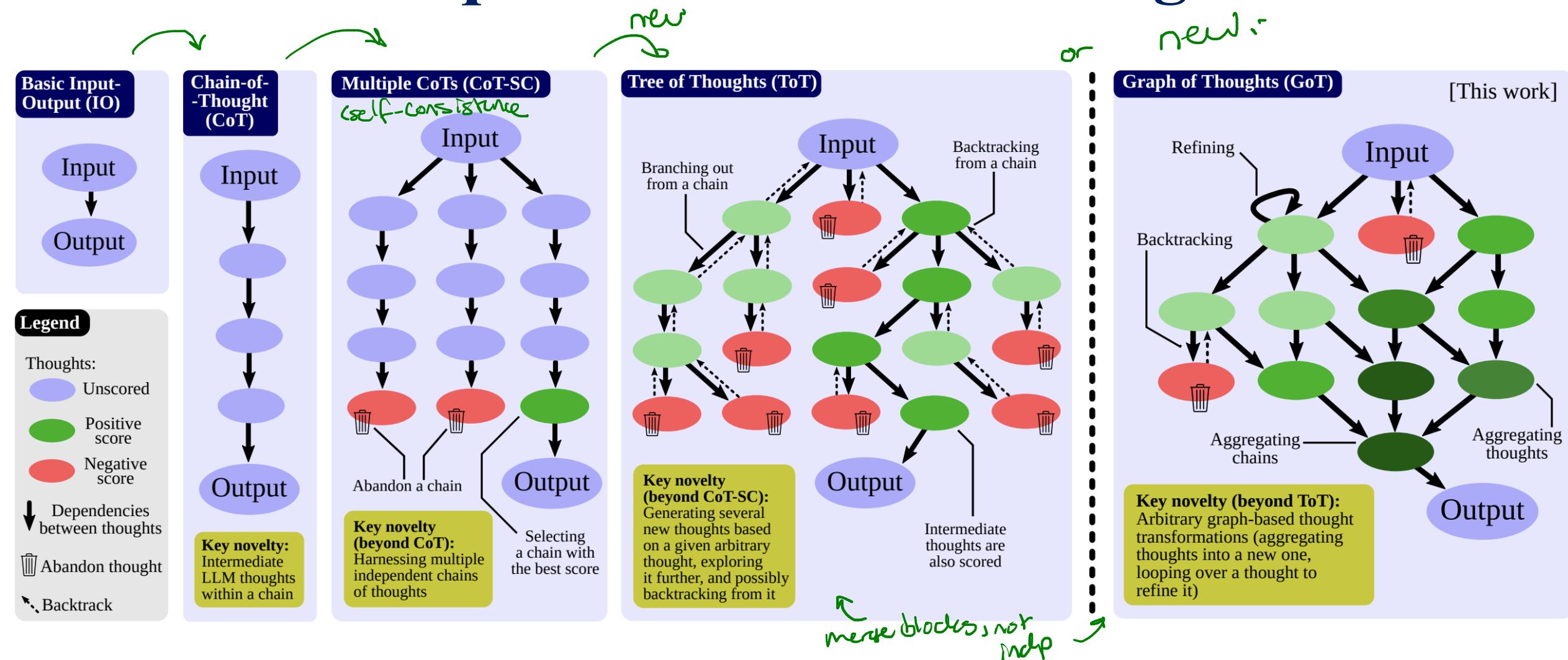
Faithfulness

Plausibility

# Self-Consistency - Form checking how consistent LLM is w/ behavior



# Other techniques to Elicit Reasoning



# How to generate plausible and correct responses?

- Zero-shot prompting
- In-context Learning (Few-shot prompting)
- Chain-of-Thought
- Retrieval Augmented Generation (RAG)

→ Retrieval Augmented Generation

# RAG is everywhere!!

Home > AI technologies

Tech Accelerator  
A guide to artificial intelligence in the enterprise

◀ PREV | NEXT ▶

FEATURE

## 10 top AI and machine learning trends for 2024

Custom enterprise models, open source AI, multimodal -- learn about the top AI and machine learning trends for 2024 and how they promise to transform the industry.

### A Beacon of Innovation: What is Retrieval Augmented Generation?

RAG transforms LLMs into dynamic systems by enabling access to an ever-updating information library



Maxime Vermeir, director of AI strategy, ABBYY  
September 9, 2024

⌚ 3 Min Read

#### Latest News

Accelerate Your GenAI Success:  
Coveo strengthens its leadership  
in GenAI with the launch of  
Relevance-Augmented Passage  
Retrieval API



## NVIDIA GTC 2024: Top 5 Trends

Published March 21, 2024

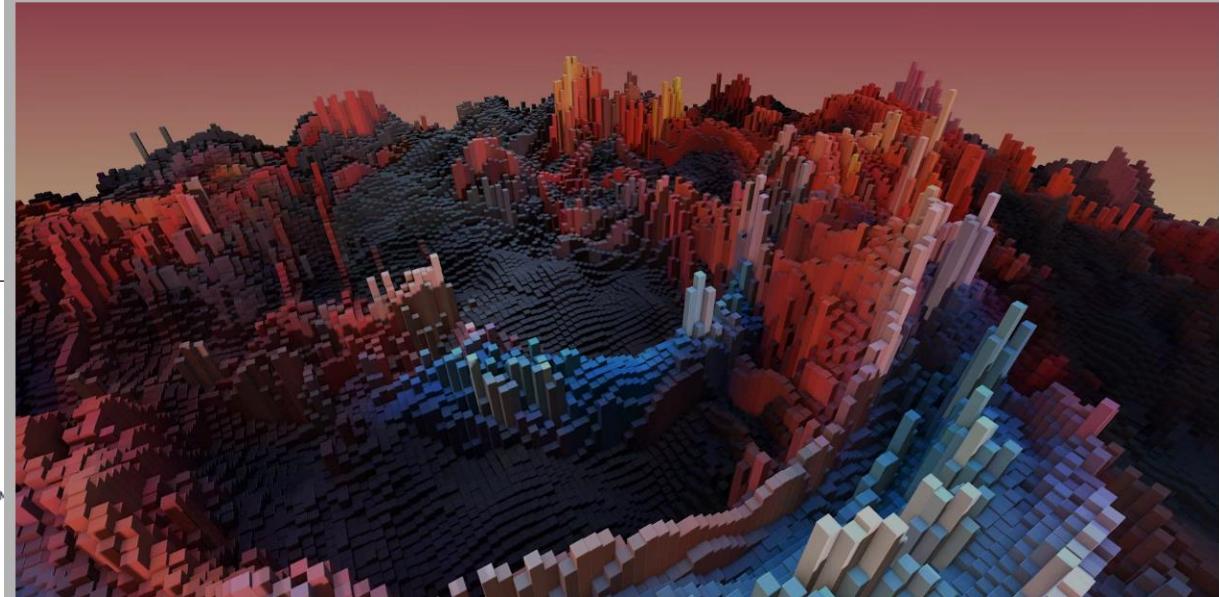


Written By  
**Megan Crouse**

NVIDIA GTC 2024 was a snapshot of enterprise trends for generative AI, highlighting private models, RAG and more.

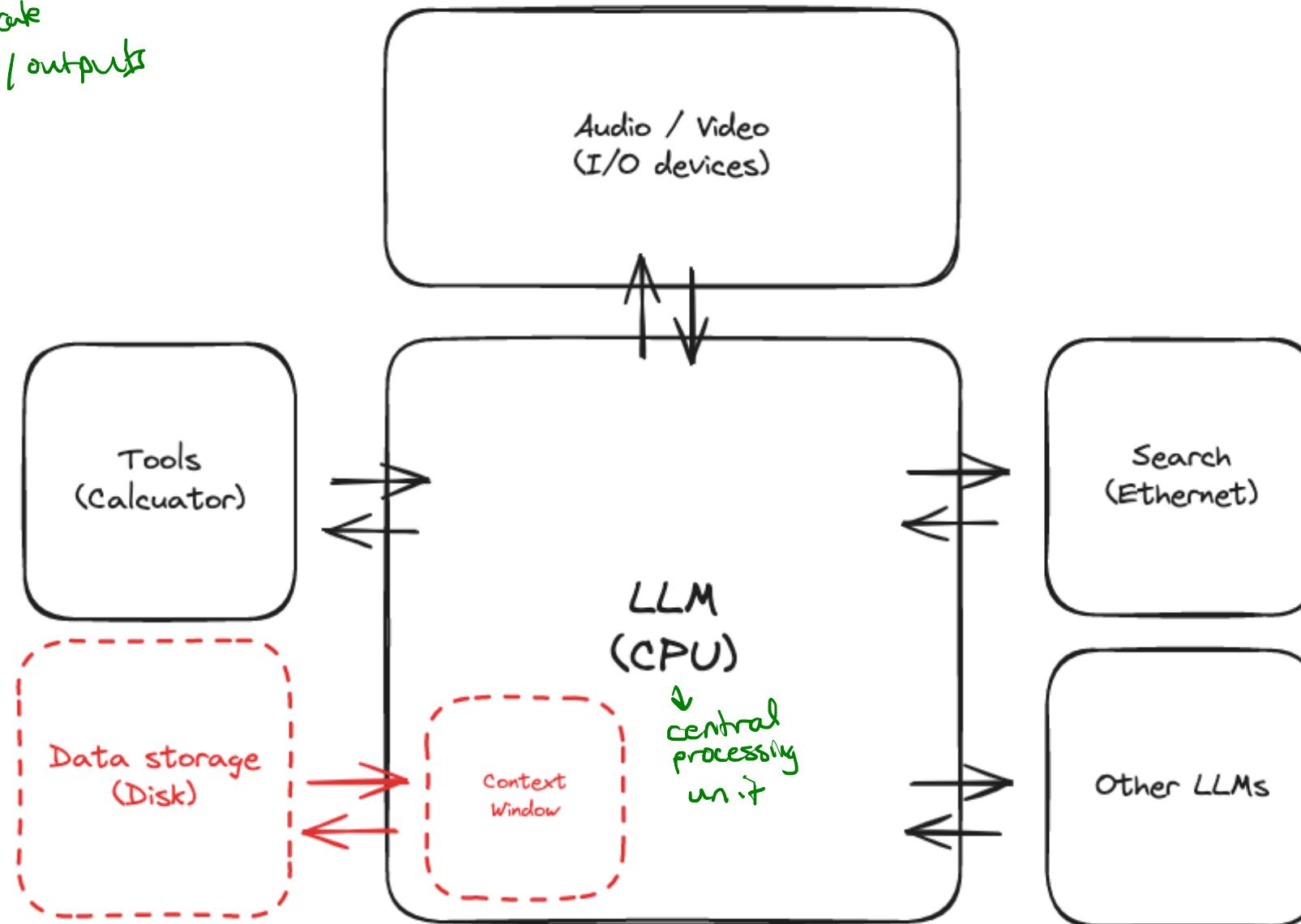
AI Agenda

### New RAG Techniques are Generative AI's Greatest Hope

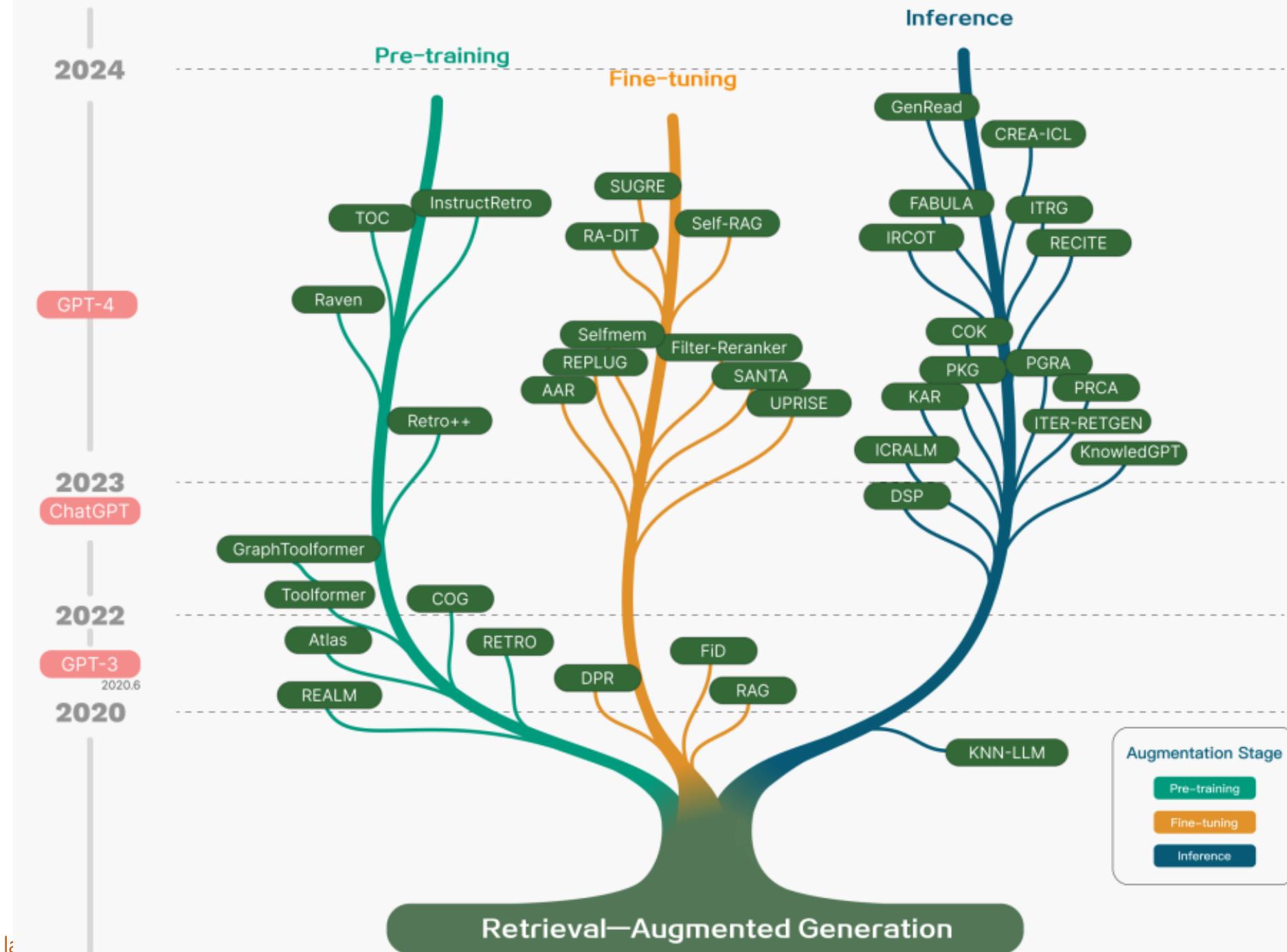


# Connecting LLMs to external data is a central need

CPU - tries communicate  
w/ diff inputs/outputs



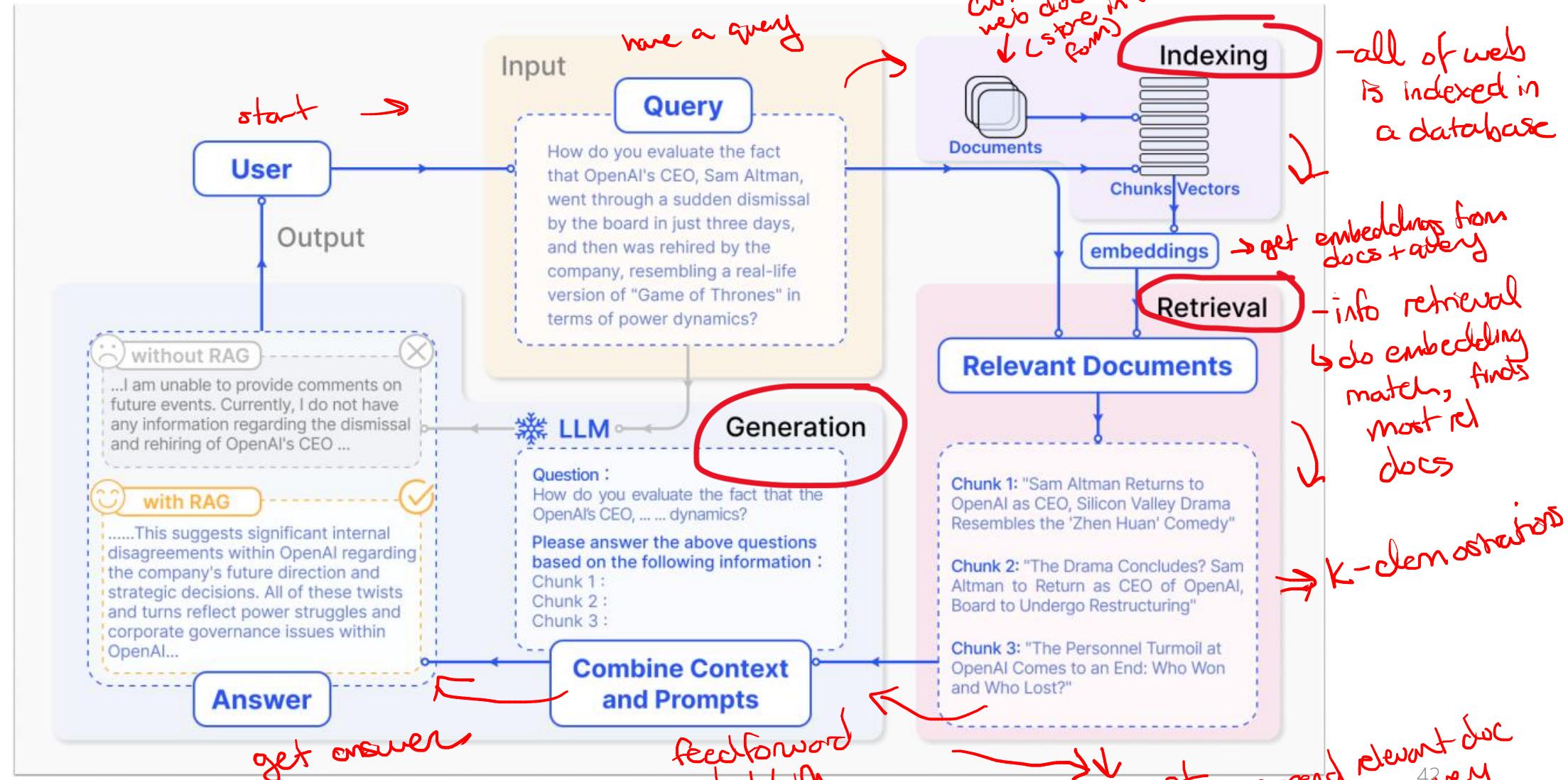
# RAG is hot in AI!!



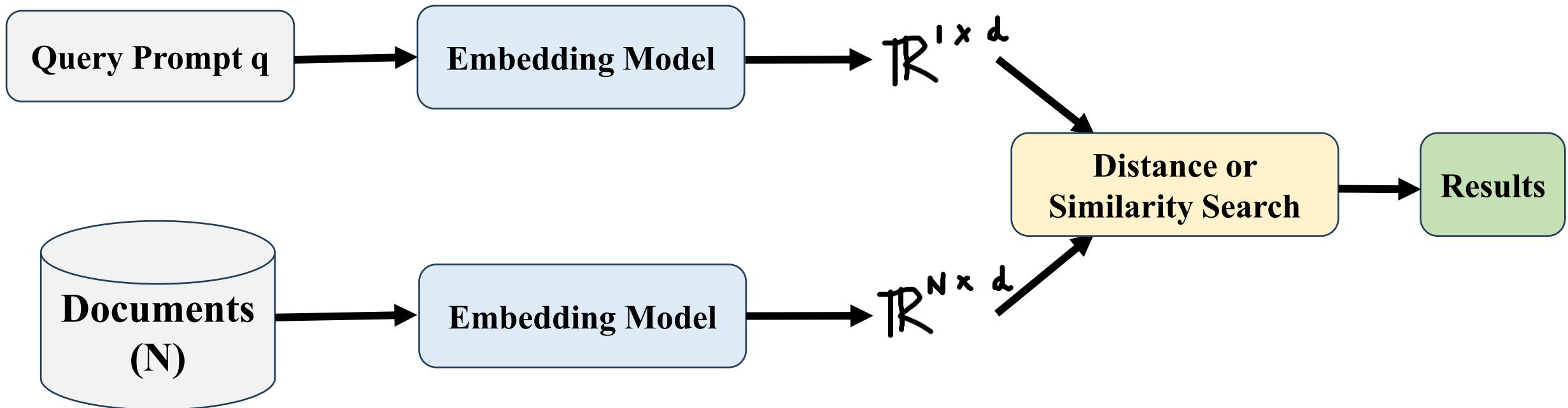
# What is Retrieval-Augmented Generation?

Know:

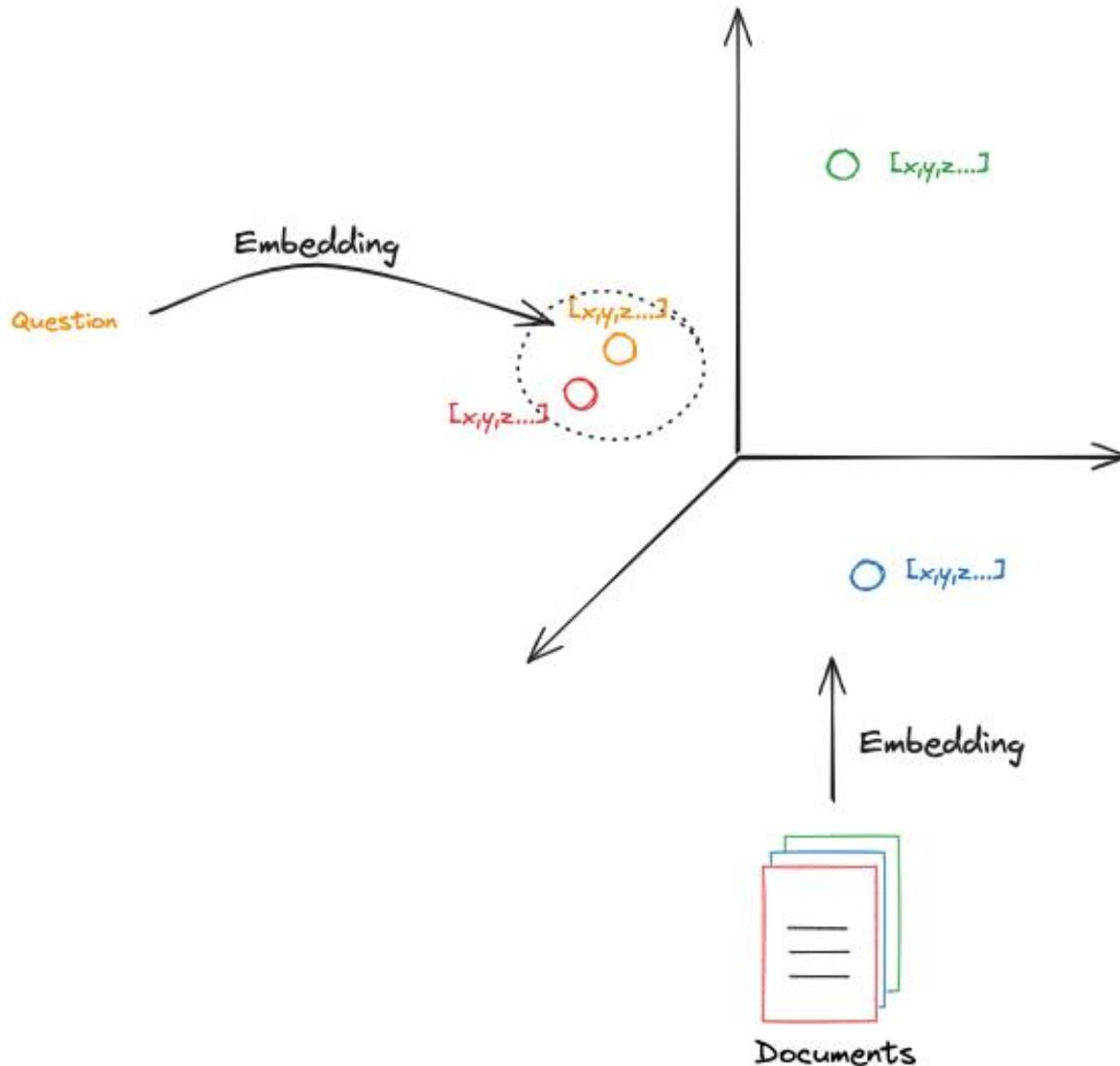
- 1) Indexing
- 2) Retrieval
- 3) Generation

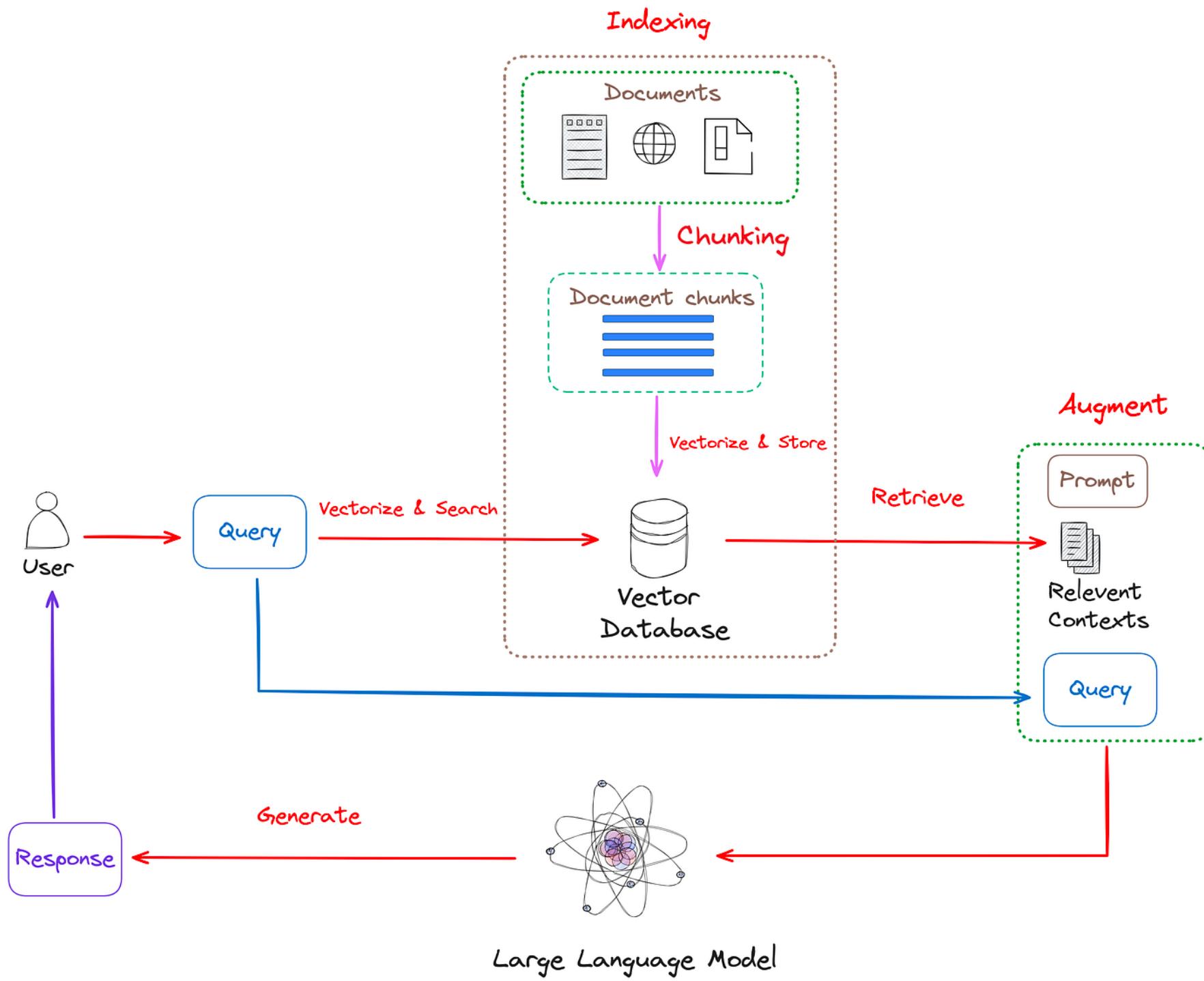


# Retrieval Augmented Generation



# Retrieval Augmented Generation





# Let's unfold it in code!!

```
# Load a pretrained Sentence Transformer model
from sentence_transformers import SentenceTransformer
model = SentenceTransformer("all-MiniLM-L6-v2")

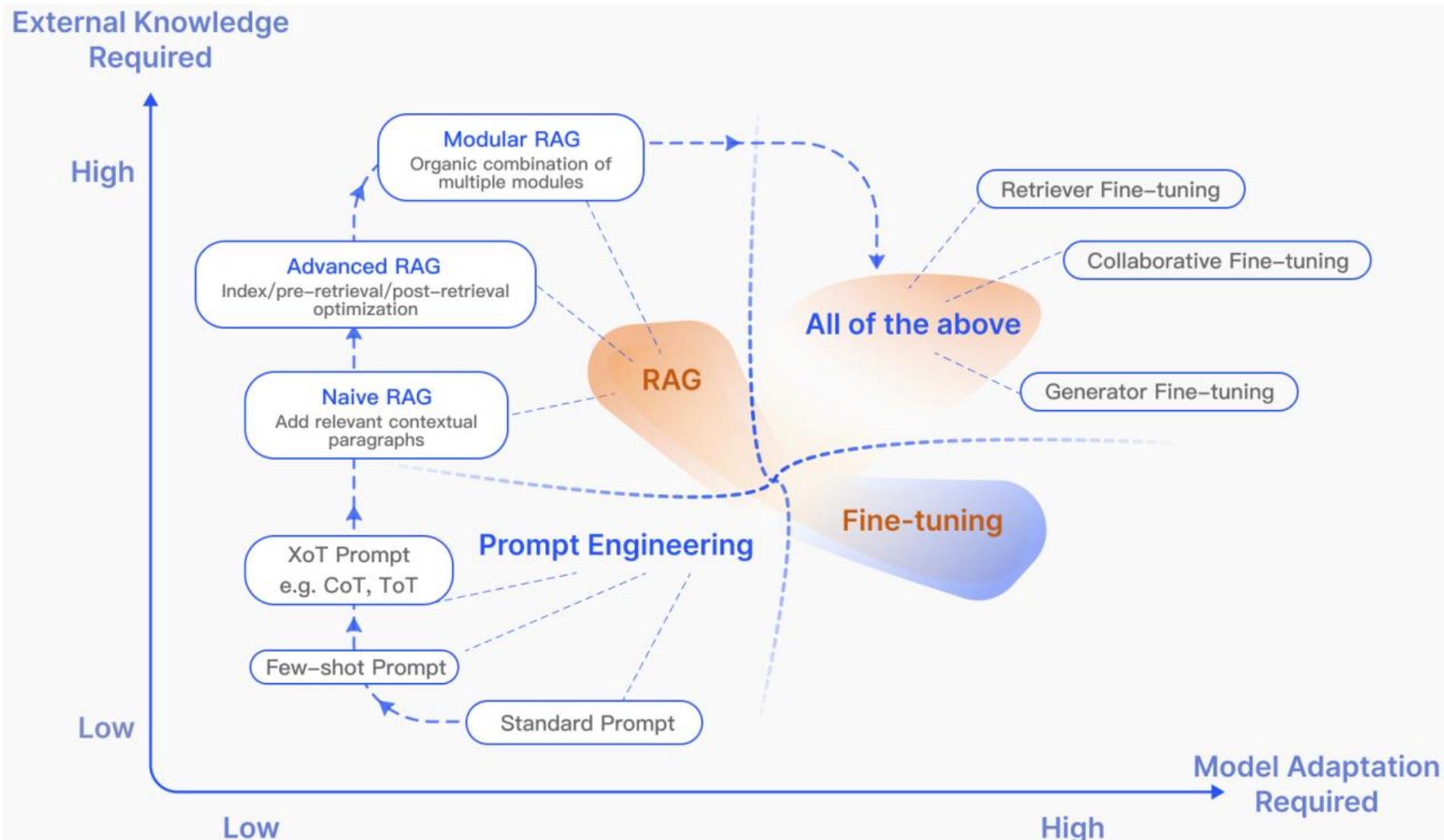
# Fetching some text content from Wikipedia
from wikipediaapi import Wikipedia
wiki = Wikipedia('RAGBot/0.0', 'en')
doc = wiki.page('Lionel Messi').text
paragraphs = doc.split('\n\n')
docs_embed = model.encode(paragraphs, normalize_embeddings=True)

# Get similar embeddings for your query prompt
query = "Who is Lionel Messi?"
query_embed = model.encode(query, normalize_embeddings=True)

# Calculate the embedding similarities
import numpy as np
k = 5
similarities = np.dot(docs_embed, query_embed.T)
print(similarities)
top_k_idx = np.argpartition(similarities,-k)[-k:]
most_similar_documents = [paragraphs[idx] for idx in top_k_idx]
print(most_similar_documents)
~

"basic_retrieve.py" 23L, 819B written
```

# RAG vs. Fine-tuning?



# Next Week!!

- Guest lecture from Rishabh Agarwal (Senior Research Scientist @ Google) on Reinforcement Learning in LLMs