

Beyond Autocomplete: Instruction Following & Chain-of-Thought Reasoning in LLM Agents

FOCI LLM USER GROUP EPISODE #3

NOVEMBER 15TH, 2023



Agenda

Introduction

Recap: How do LLMs Work?

Instruction Tuning

Chain-of-Thought (CoT)

Retrieval Augmented Generation
(RAG)

LLM Agents

Discussion

Introduction

What makes LLMs so useful?

LLMs were born from NLP research, but their capabilities have continuously evolved:

2013: How to represent text?

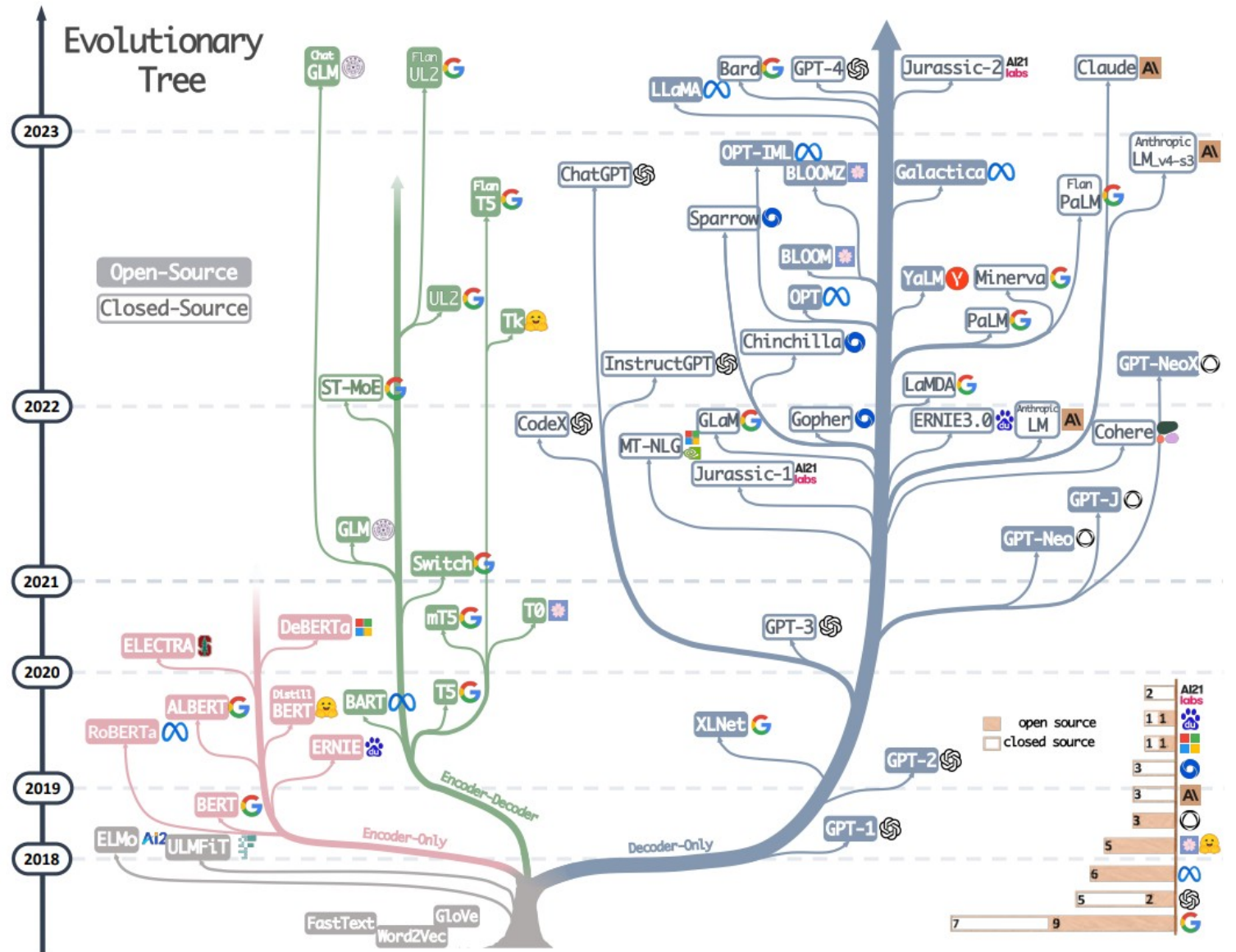
2014: How to generate text?

2015-2017: Focused NLP applications with RNNs

2017-2019: Contextual representation & scalable transfer learning

2020 - 2023: General NLP applications

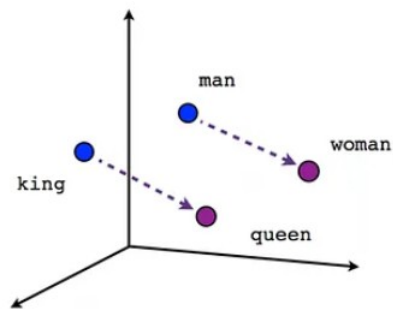
2023 - ??? : General multimodal applications



Introduction

2013: How to represent text?

Source



2014: How to generate text?

Source

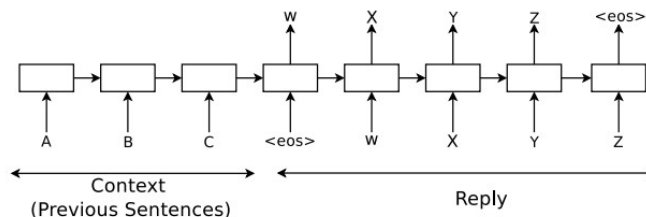
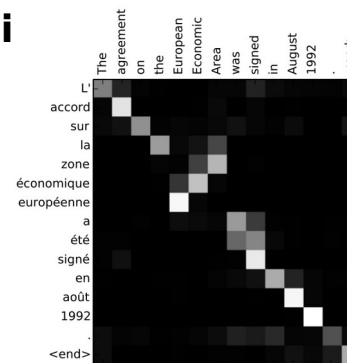


Figure 1. Using the seq2seq framework for modeling conversations.

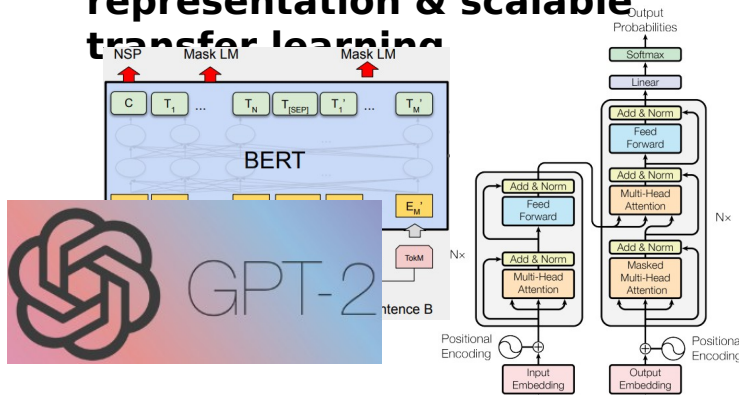
2015-2016: Focused NLP applications

Source

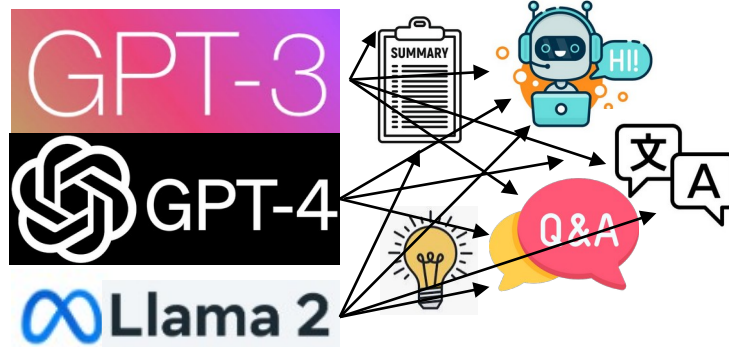


(e.g., Machine Translation)

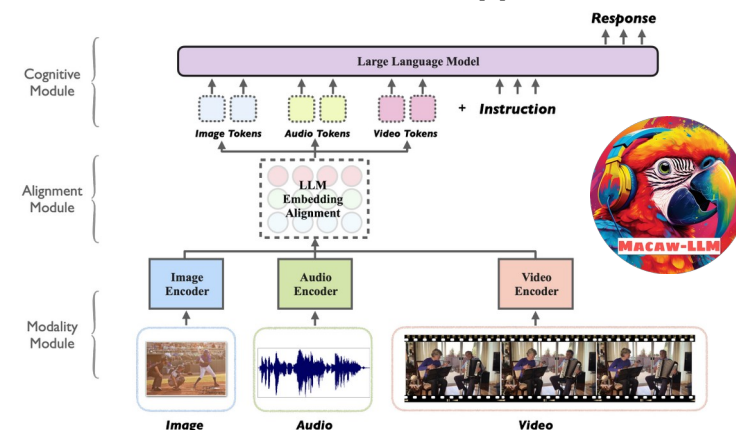
2017-2019: Contextual representation & scalable transfer learning



2020 - 2023: General NLP applications

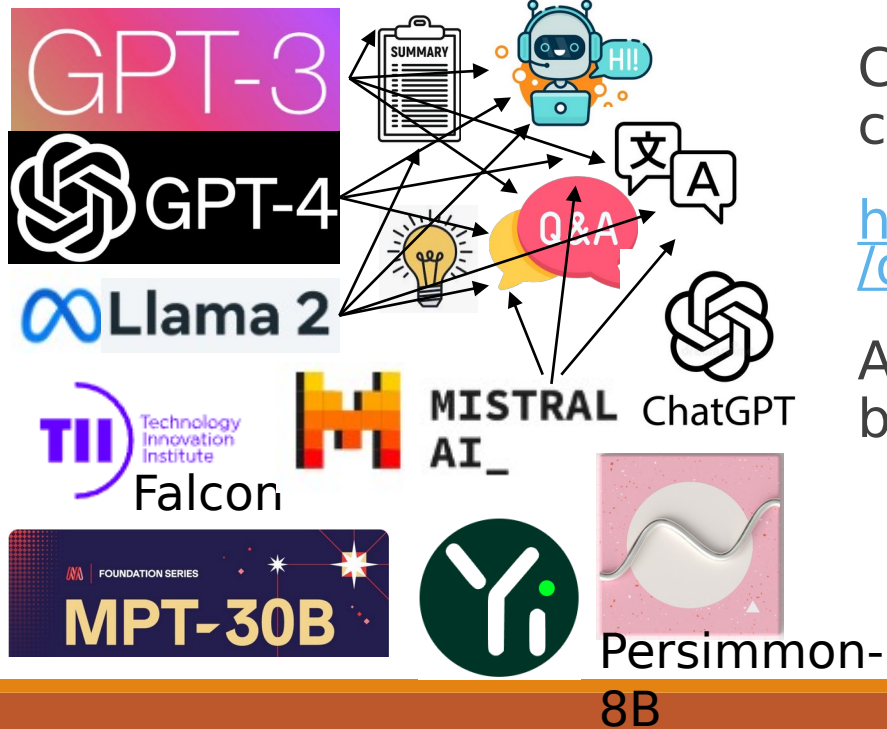


2023 - ??? : General multimodal applications



Introduction

2020 - 2023: General NLP applications

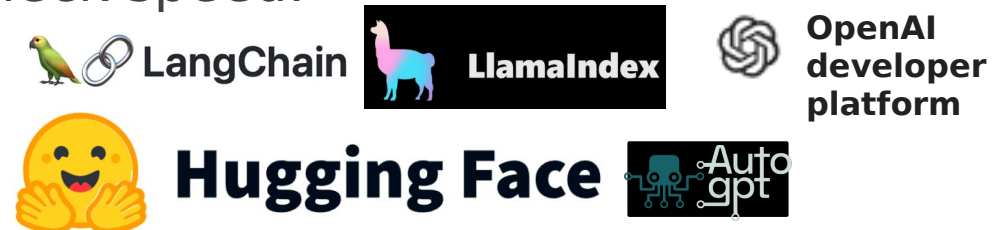


Since ChatGPT's release, open and commercial LLMs have become ubiquitous...

Capable of many tasks and catalyzed by constant competition to be the best...

https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

Applications and ecosystems have grown at breakneck speed:

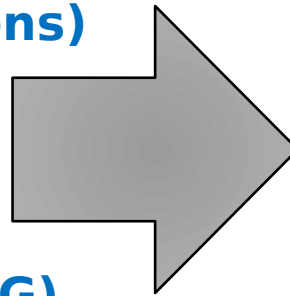


Introduction

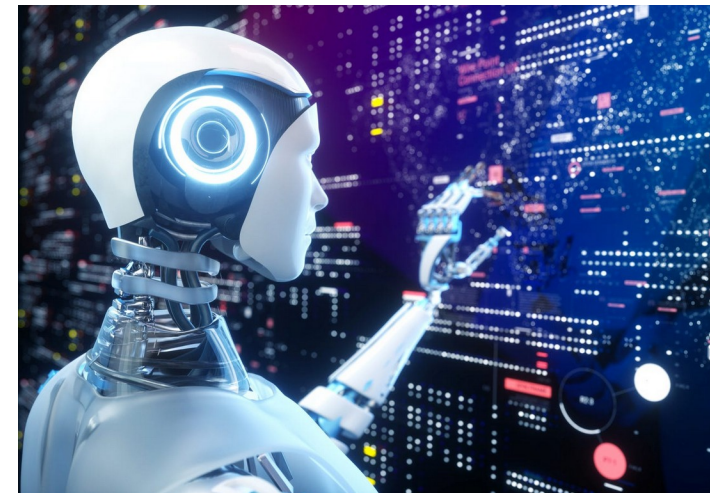
Q: Where was the fundamental “jump” in LLM capability in 2021-2023?

A: Several contributing factors:

- **Scale (# parameters, # tokens)**
- **Instruction Tuning**
- **Alignment (RLHF)**
- **Advanced Prompting (CoT)**
- **Retrieval Augmentation (RAG)**
- **Tool Use**
- **Perception-Action loop**



Autonomous LLM Agents!



[Image Source](#)
[e](#)

Agenda

Introduction

Recap: How do LLMs Work?

Instruction Tuning

Chain-of-Thought (CoT)

Retrieval Augmented Generation
(RAG)

LLM Agents

Discussion

Recap: How do LLMs Work?

He opened the door and slowly stepped inside. Immediately he felt a sharp pain stabbing through his foot, for the floor was covered with



Next word	Probability
hundred	0.01
pepperoni	0.02
burning	0.85
nothing	0.02
something	0.10

Recap: How do LLMs Work?

He opened the door and slowly stepped inside. Immediately he felt a sharp pain stabbing through his foot, for the floor was covered with



Next word	Probability
hundred	0.01
pepperoni	0.02
burning	0.85
nothing	0.02
something	0.10

Recap: How do LLMs Work?

He opened the door and slowly stepped inside. Immediately he felt a sharp pain stabbing through his foot, for the floor was covered with **burning**



Next word	Probability
rubber	0.01
coal	0.04
pizza	0.03
sensation	0.02
hot	0.90

Recap: How do LLMs Work?

He opened the door and slowly stepped inside. Immediately he felt a sharp pain stabbing through his foot, for the floor was covered with **burning**



Next word	Probability
rubber	0.01
coal	0.04
pizza	0.03
sensation	0.02
hot	0.90

Recap: How do LLMs Work?

He opened the door and slowly stepped inside. Immediately he felt a sharp pain stabbing through his foot, for the floor was covered with **burning hot**



Next word	Probability
lava	0.50
dogs	0.03
coal	0.45
ice	0.01
iphones	0.01

Recap: How do LLMs Work?

He opened the door and slowly stepped inside. Immediately he felt a sharp pain stabbing through his foot, for the floor was covered with **burning hot**



Next word	Probability
lava	0.50
dogs	0.03
coal	0.45
ice	0.01
iphones	0.01

Recap: How do LLMs Work?

He opened the door and slowly stepped inside. Immediately he felt a sharp pain stabbing through his foot, for the floor was covered with **burning hot lava**



Next word	Probability
and	0.10
from	0.25
.	0.30
lamps	0.20
beans	0.15

Recap: How do LLMs Work?

He opened the door and slowly stepped inside. Immediately he felt a sharp pain stabbing through his foot, for the floor was covered with **burning hot lava**



Next word	Probability
and	0.10
from	0.25
.	0.30
lamps	0.20
beans	0.15

Recap: How do LLMs Work?

He opened the door and slowly stepped inside. Immediately he felt a sharp pain stabbing through his foot, for the floor was covered with **burning hot lava**.



Next word	Probability
This	0.05
He	0.05
It	0.03
<eos>	0.85
Then	0.02

Recap: How do LLMs Work?

He opened the door and slowly stepped inside. Immediately he felt a sharp pain stabbing through his foot, for the floor was covered with **burning hot lava**.



Next word	Probability
This	0.05
He	0.05
It	0.03
<eos>	0.85
Then	0.02

Recap: How do LLMs Work?

He opened the door and slowly stepped inside. Immediately he felt a sharp pain stabbing through his foot, for the floor was covered with **burning hot lava.** <eos>



Next word	Probability
DONE	
!	

Recap: How do LLMs Work?

In text generation, we feed tokens in to an LLM and predict the next ones **autoregressively**.

Input text is first preprocessed by **tokenization** into words or subwords:

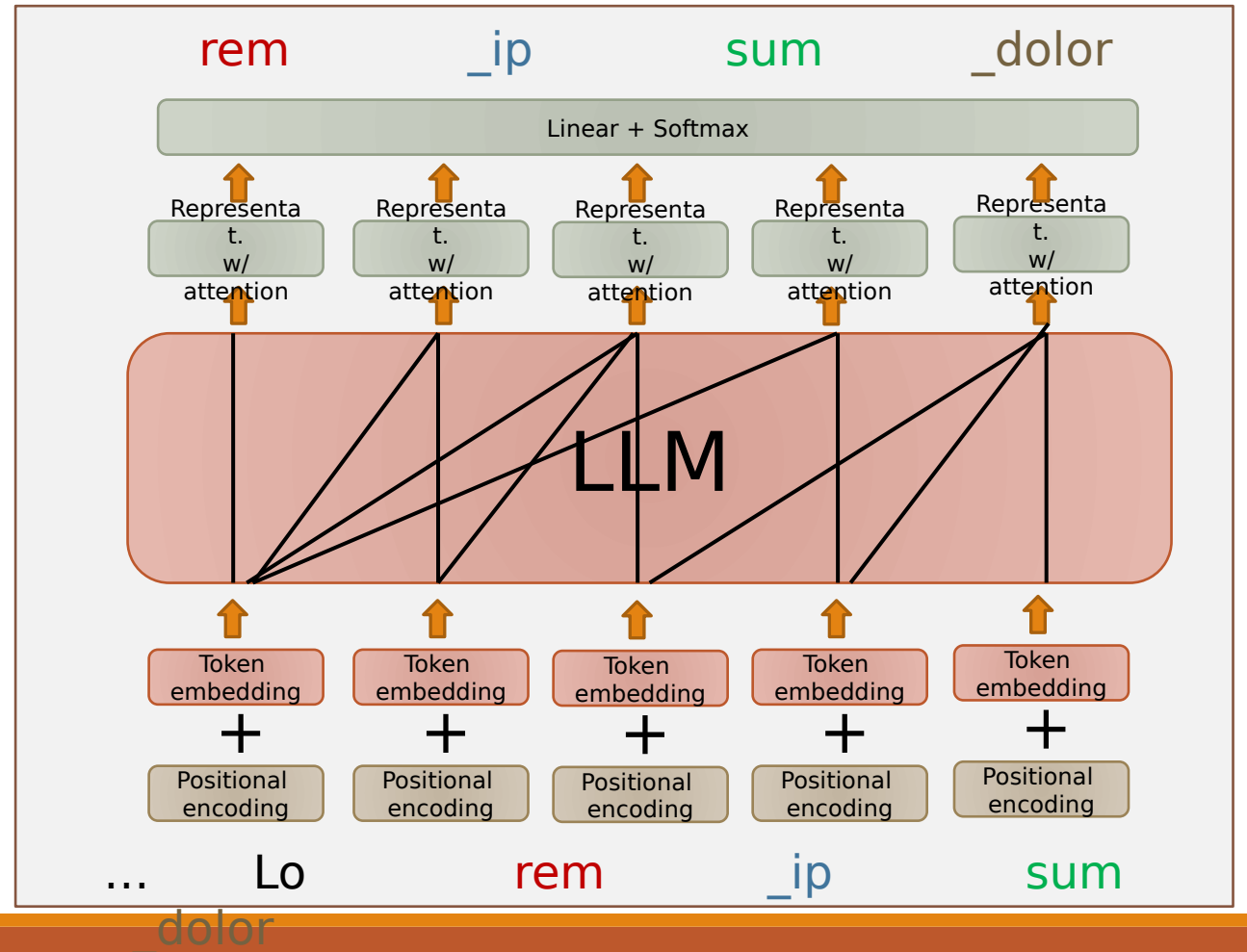
"Lorem ipsum dolor sit amet"



["Lo", "rem", "_ip", "sum", "_dolor", "_sit", "_a", "met"]



[5643, 6568, 332, 2224, 99, 129, 22931, 2321]



Recap: How do LLMs Work?

Input text is first preprocessed by **tokenization** into words or subwords:

"Lorem ipsum dolor sit amet"

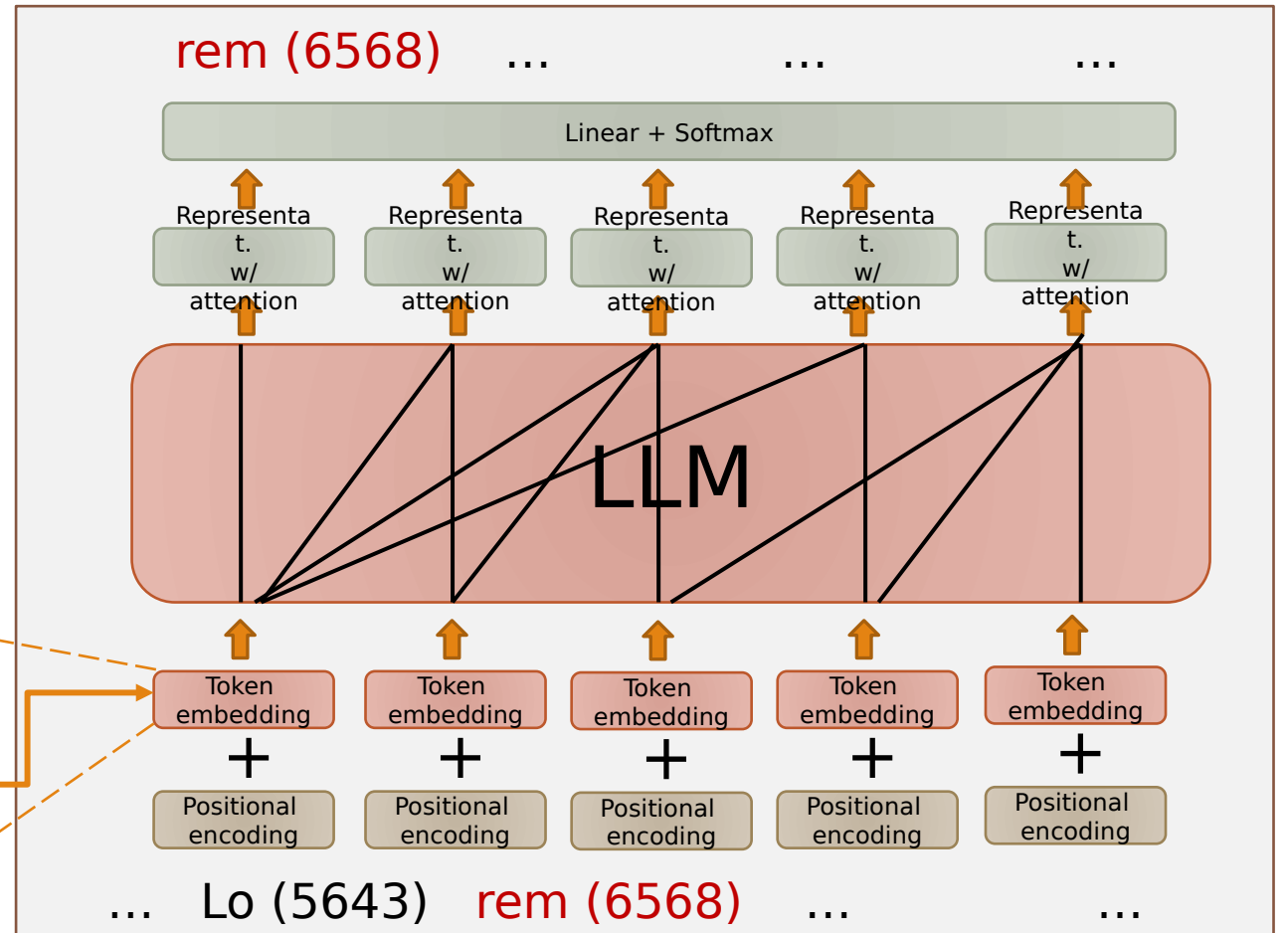
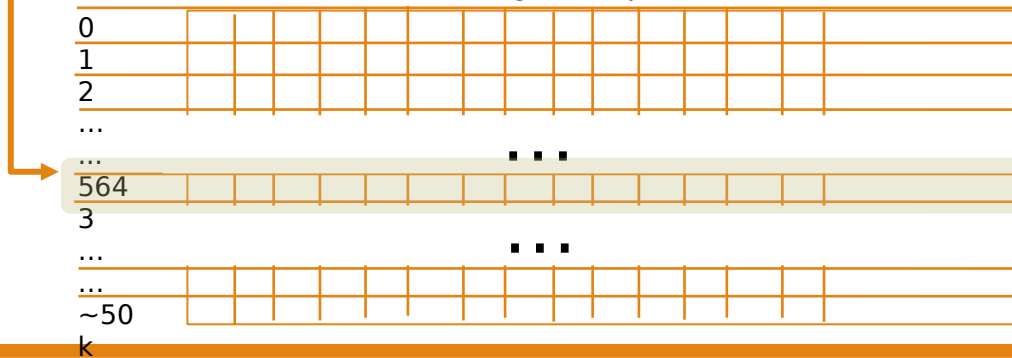


["Lo", "rem", "_ip", "sum", "_dolor", "_sit", "_a", "met"]



[5643, 6568, 332, 2224, 99, 129, 22931, 2321]

Embedding lookup table



Recap: How do LLMs Work?

- Embeddings are the way tokens are fed into the LLM. An embedding is a numeric array (vector) which encodes the contextual similarity of a token with other tokens.

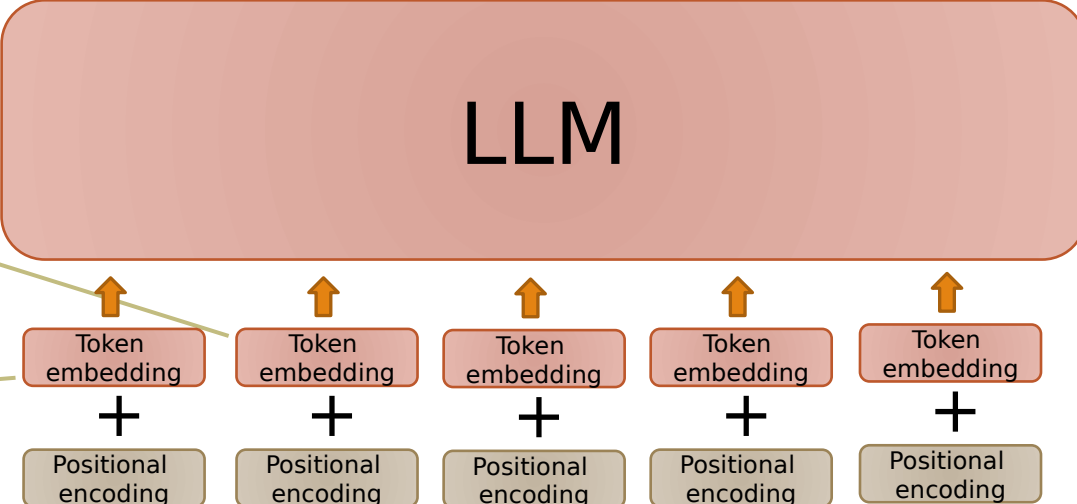
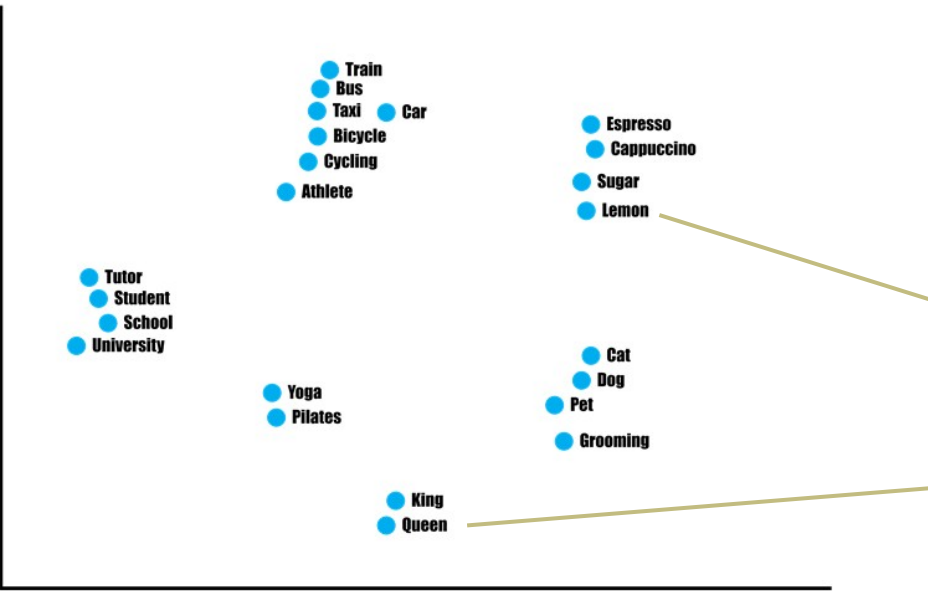
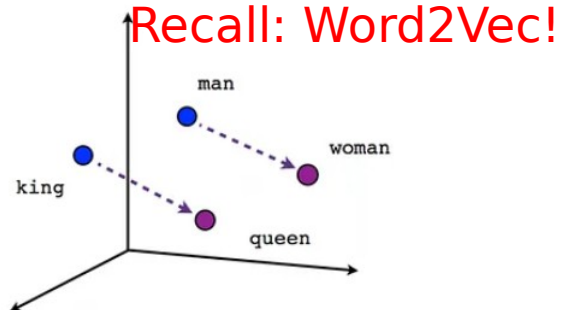


Image source: <https://predictivehacks.com/a-high-level-introduction-to-word-embeddings/>

Recap: How do LLMs Work?

Autoregressive Language Models come in encoder-decoder or decoder-only setups.

Early work (e.g., Sutskever et al., 2014, Vinyals & Le, 2015) used Long-short Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997) to **probabilistically** model the sequence of words in a

$$p(y_1, \dots, y_{T'} | x_1, \dots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \dots, y_{t-1})$$

Image Sources:

- Sequence to Sequence Learning with Neural Networks (Sutskever et al., 2014)
- A Neural Conversational Model (Vinyals & Le, 2015)
- Chris Olah's blog <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

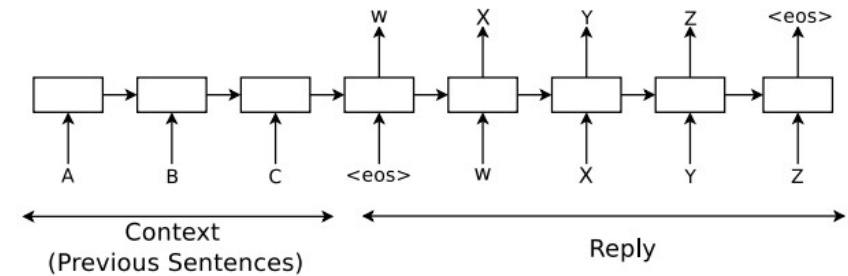
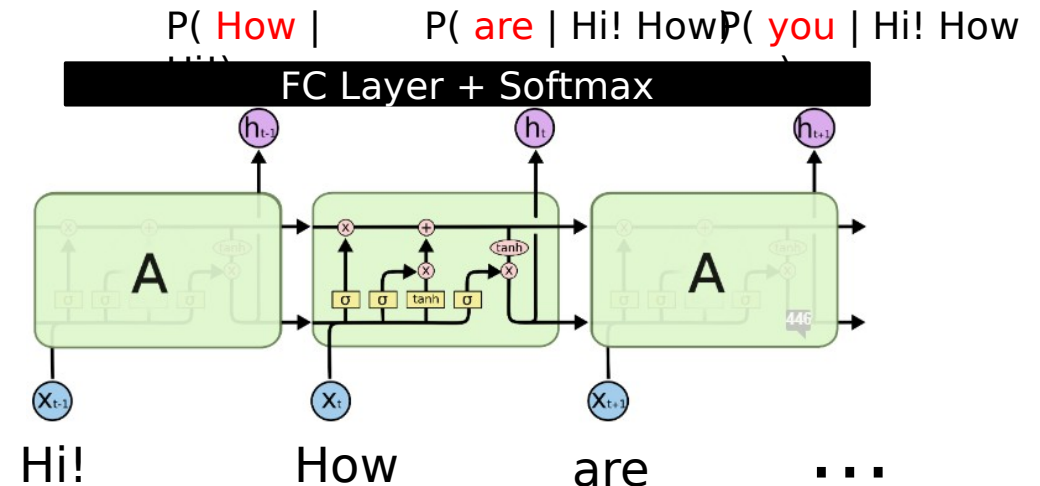


Figure 1. Using the *seq2seq* framework for modeling conversations.



Recap: How do LLMs Work?

Modern LLMs are still **Autoregressive Language Models** that model language **probabilistically** – just with a different backbone:

The **Transformer** (Vaswani et al., 2017):
 Transformers replace recurrence with **Positional Encodings** and **Self-Attention**!

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

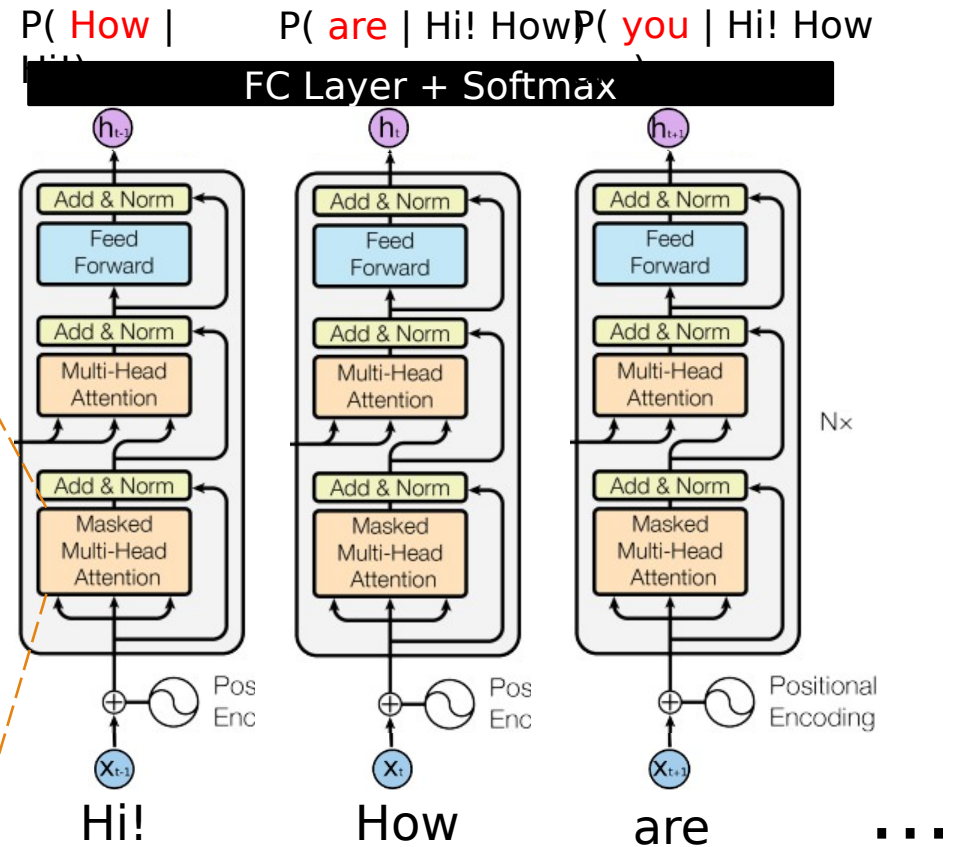
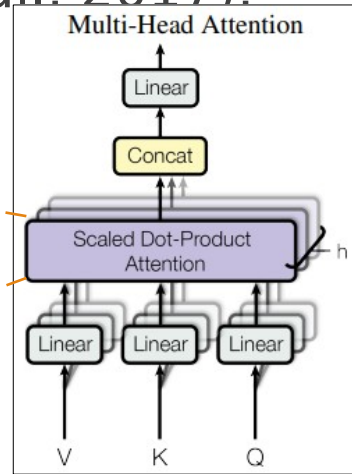


Image Source: Attention is All You Need (Vaswani et al., 2017)

Recap: How do LLMs Work?

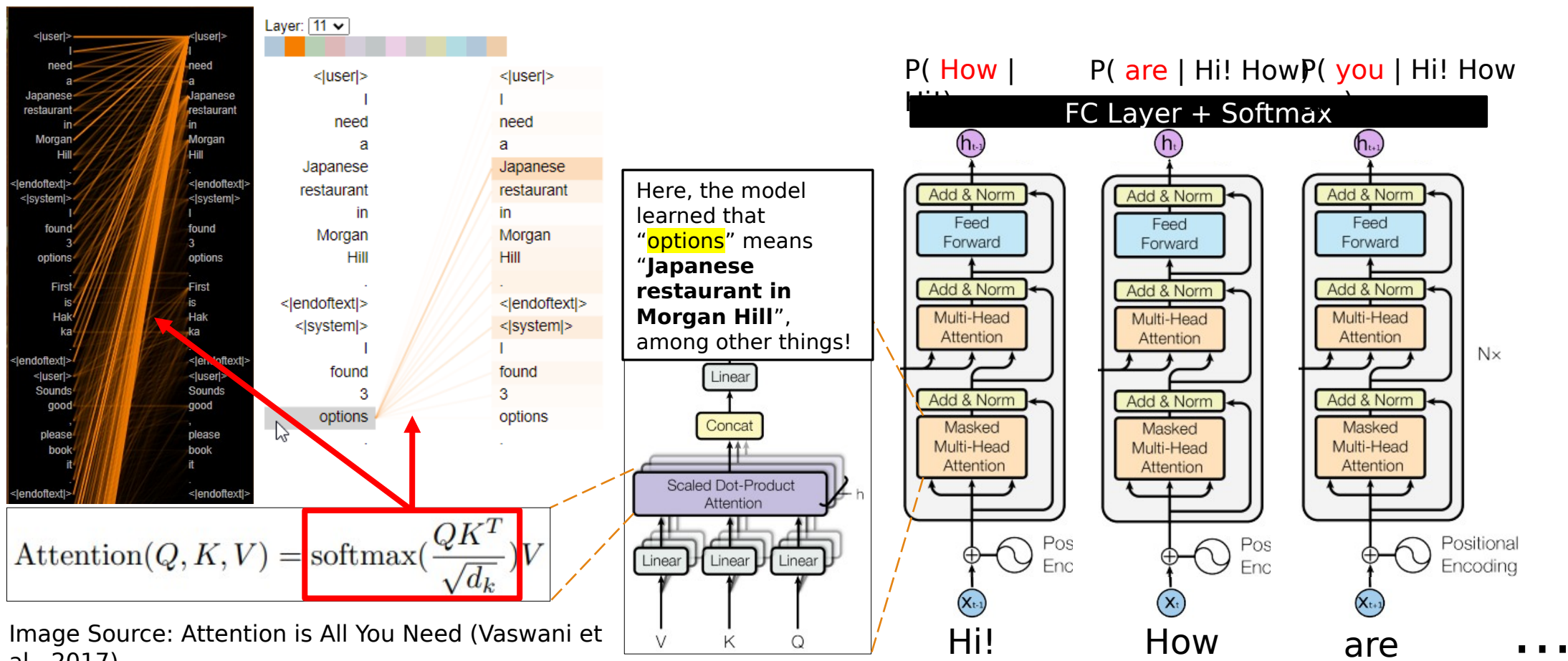


Image Source: Attention is All You Need (Vaswani et al., 2017)

Agenda

Introduction

Recap: How do LLMs Work?

Instruction Tuning

Chain-of-Thought (CoT)

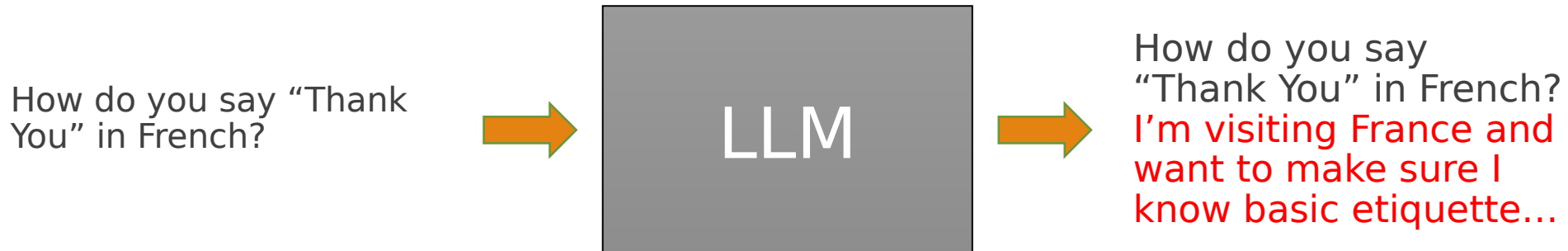
Retrieval Augmented Generation
(RAG)

LLM Agents

Discussion

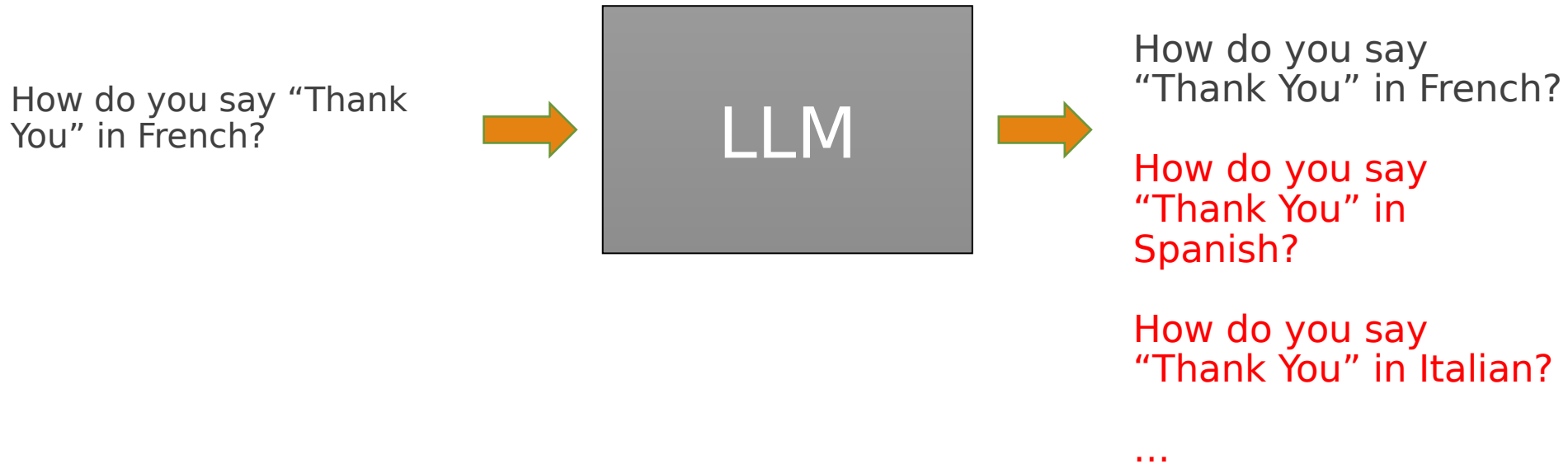
Instruction Tuning

So, LLMs are just overparameterized autocomplete models.



Instruction Tuning

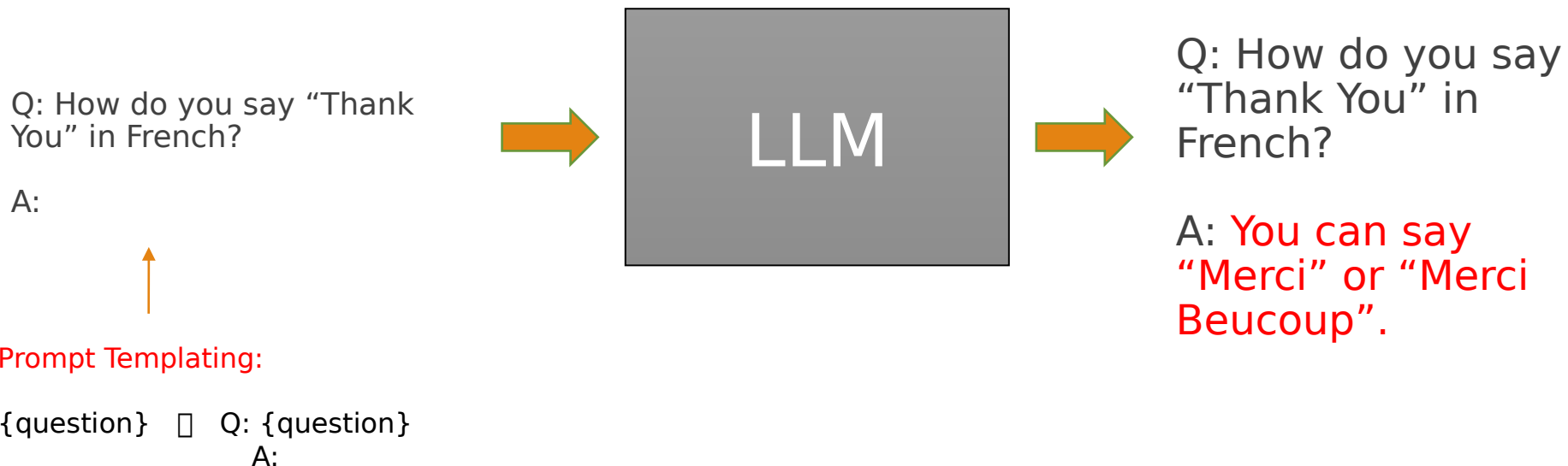
So, LLMs are just overparameterized autocomplete models.



Instruction Tuning

So, LLMs are just overparameterized autocomplete models.

Prompt engineering is needed to get desired results:



Instruction Tuning

So, LLMs are just overparameterized autocomplete models.

Few-shot learning is often needed to “teach” the LLM a new task in context:

Q: How do you say “Thank You” in Spanish?

A: You can say “Gracias” or “Muchas Gracias”.

Q: How do you say “Thank You” in German?

A: You can say “Danke” or “Danke Schon”.

Q: How do you say “Thank You” in French?

A:



Q: How do you say “Thank You” in Spanish?

A: You can say “Gracias” or “Muchas Gracias”.

Q: How do you say “Thank You” in German?

A: You can say “Danke” or “Danke Schon”.

Q: How do you say “Thank You” in French?

A: You can say “Merci” or “Merci Beaucoup”.

Few-Shot Prompt Templating:

```
{example Q1}, {example A1},  
{example Q2}, {example A2},  
...  
{question} □ Q: {example Q1}  
A: {example A1}  
...  
Q: {question}  
A:
```

Instruction Tuning

So, LLMs are just overparameterized autocomplete models.

If you have a couple thousand examples you can also fine-tune the weights directly for the desired template, for example to just treat every input as a question and try to answer it:

How do you say "Thank You" in French?



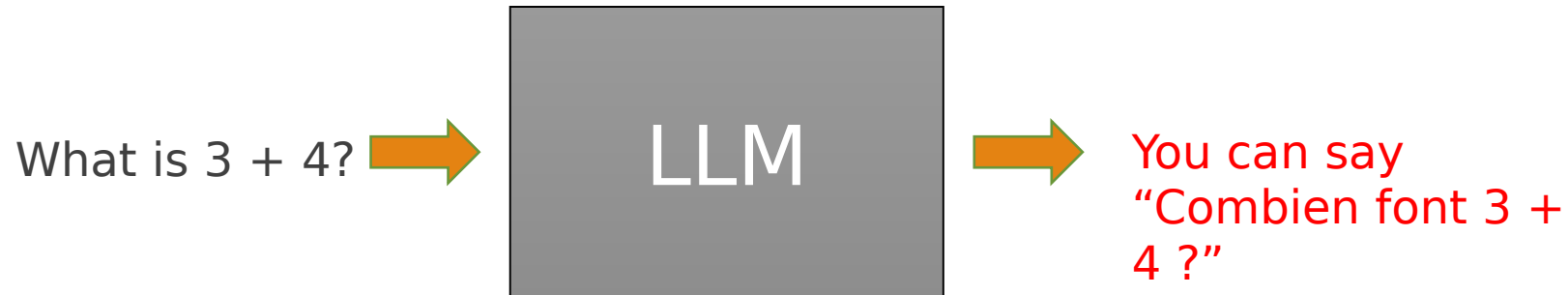
You can say "Merci" or "Merci Beucoup".

Critically: Fine-tuning can make the most likely autocomplete become a natural response!

Instruction Tuning

Critically: Fine-tuning can make the most likely autocompletion become a natural response!

BUT: a fine-tuned LLM would become specialized to that one task and be incapable of others.



Instruction Tuning

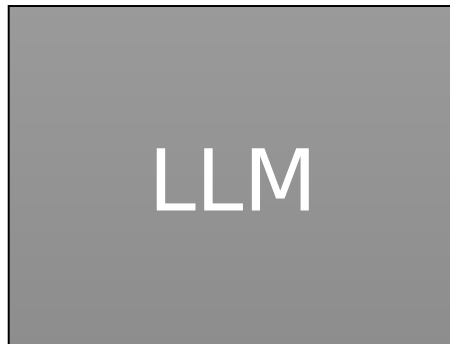
Critically: Fine-tuning can make the most likely autocompletion become a natural response!

The solution? **Instruction Tuning!**

- Fine-tune on a mixture of tasks prefixed with natural language instructions:

Answer the following question:

What is $3 + 4$?



The answer is 7.

Translate the following sentence into French:

"Thank You"



"Merci" or "Merci Beucoup"

Instruction Tuning

Two traditional methods to get a LLM to do a task:

- Fine-tuning
- N-shot prompting

A third is introduced here to get the best of both worlds:

- Instruction tuning

Key insight:

- Fine-tuning a LLM on a very large set of downstream tasks with instruction-following prompts teaches the LLM to *follow general instructions*, enabling superior zero-shot performance!

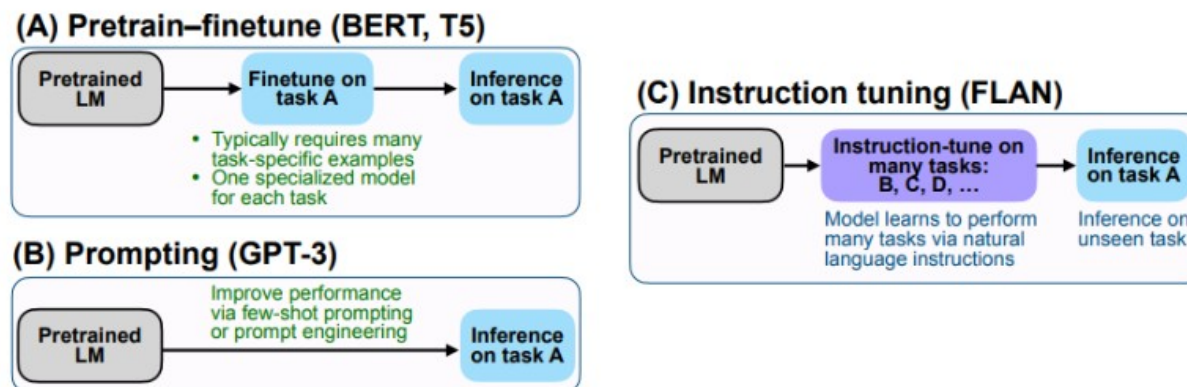
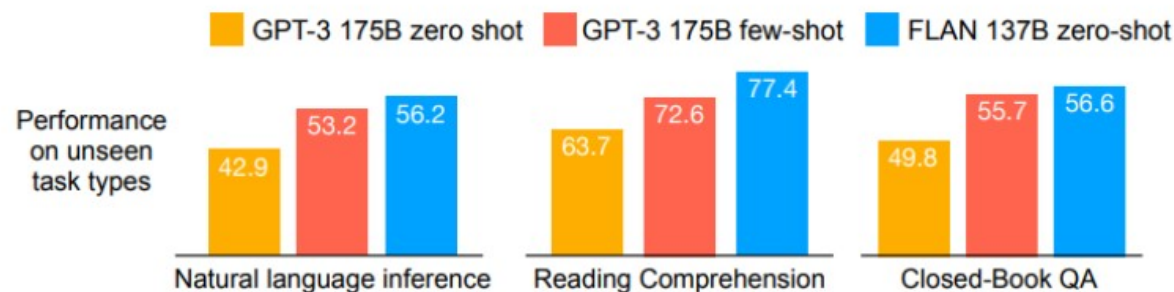


Figure 2: Comparing instruction tuning with pretrain–finetune and prompting.

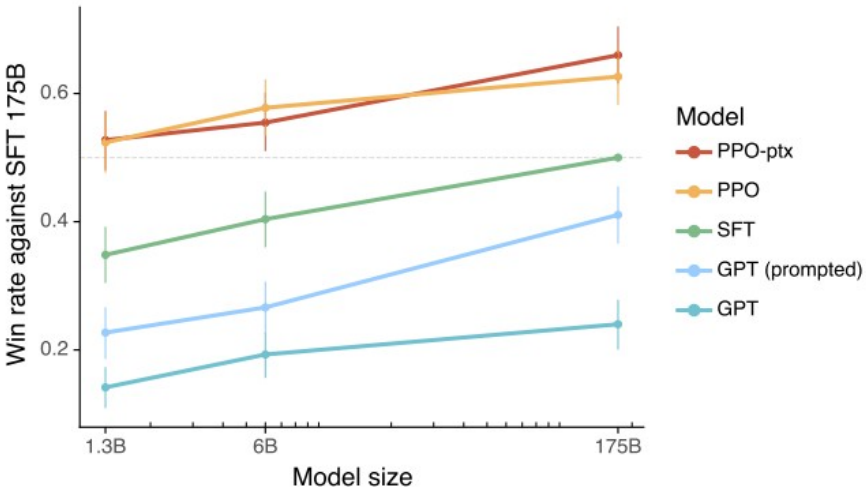


Wei et al., ICLR 2022; Google Research.

<https://arxiv.org/pdf/2109.01652.pdf>

Instruction Tuning

- Instruction tuning is now the de-facto standard for LLMs used as Assistants or Agents. Some very influential works:
 - **FLAN + FlanT5** (Wei et al., 2022; Chung et al, 2022)
 - **InstructGPT + ChatGPT** (Ouyang et al., 2022; OpenAI blog, 2022)



classification	<p>You are a very serious professor, and you check papers to see if they contain missing citations. Given the text, say whether it is missing an important citation (YES/NO) and which sentence(s) require citing.</p> <p>{text of paper}</p>									
extract	<p>Extract all course titles from the table below:</p> <table border="1"> <tr> <td>Title</td> <td>Lecturer</td> <td>Room</td> </tr> <tr> <td>Calculus 101</td> <td>Smith</td> <td>Hall B</td> </tr> <tr> <td>Art History</td> <td>Paz</td> <td>Hall A</td> </tr> </table>	Title	Lecturer	Room	Calculus 101	Smith	Hall B	Art History	Paz	Hall A
Title	Lecturer	Room								
Calculus 101	Smith	Hall B								
Art History	Paz	Hall A								
chat	<p>The following is a conversation with an AI assistant. The assistant is helpful, creative, clever, and very friendly.</p> <p>Human: Hello, who are you? AI: I am an AI created by OpenAI. How can I help you today? Human: I'd like to cancel my subscription. AI:</p>									

Ouyang et al., 2022; <https://arxiv.org/pdf/2203.02155.pdf>

Instruction Tuning Examples (FLAN)

Input:

John is reading the paper
"Finetuned Language Models are
Zero-Shot Learners."
Change to past tense.

FLAN output:

John was reading the paper
"Finetuned Language Models are
Zero-Shot Learners."

Input:

John is reading the paper
"Finetuned Language Models are
Zero-Shot Learners."
Change to passive voice.

FLAN output:

The paper "Finetuned Language
Models are Zero-Shot Learners" is
being read by John.

Input:

John is reading the paper
"Finetuned Language Models are
Zero-Shot Learners."
Change to the verb to eat.

FLAN output:

John is eating the paper
"Finetuned Language Models are
Zero-Shot Learners."

Input:

John is reading the paper
"Finetuned Language Models are
Zero-Shot Learners."
Change the verb to eat and the
subject to Noam Chomsky.

FLAN output:

Noam Chomsky is eating the paper
"Finetuned Language Models are
Zero-Shot Learners."

Figure 15: FLAN can perform zero-shot rule-based manipulations.

Input:

Movie review: This movie is the
best RomCom since Pretty Woman.

Did this critic like the movie?

OPTIONS

-yes
-no

FLAN output:

yes

Input:

Movie review: This movie is the
best RomCom since Pretty Woman.

Did this critic dislike the movie?

OPTIONS

-yes
-no

FLAN output:

no

Figure 13: For sentiment analysis, FLAN changes the answer appropriately when the question is flipped.

Input:

What is Barack Obama's
profession?

FLAN output:

- politician
- lawyer
- senator
- us president

Input:

What is Barack Obama's
profession?
Answer in Chinese

FLAN output:

- 总统
- 美国总统

Figure 14: For question answering, FLAN can answer a question in another language when instructed to do so. Multiple FLAN outputs are generated via random sampling with a temperature of 0.9 and top k of 40.

Instruction Tuning Examples (FLAN)

Input:
Recommend fun activities to do on a rainy night in Seattle.

FLAN output:

- Go to the pacific science center
- Go out to eat
- Go to a comedy club
- Build a blanket fort
- Go to the Space Needle
- Go to underground tours
- Find a karaoke bar

Input:
Recommend activities to do on a sunny weekend in Copenhagen.

FLAN output:

- Visit Christiania
- Copenhagen Zoo
- Visit the Tivoli Gardens
- Take a canal tour
- Museums
- Visit The Little Mermaid Statue
- Visit Copenhagen Zoo
- Visit the Danish Maritime Museum

Figure 16: FLAN can make zero-shot recommendations. Multiple FLAN outputs are generated via random sampling with a temperature of 0.9 and top k of 40.

Input:
Generate utterances with the intent "get COVID vaccine"

FLAN output:

- I would like to get the covid vaccine.
- I would like a COVID vaccine as soon as possible
- Can I get a COVID vaccine as a preventative measure?
- Get COVID vaccine for a five year old
- Where is a center for COVID vaccine testing?

Input:
Generate utterances with the intent "where to vote in the state Senate election"

FLAN output:

- Tell me where to vote in the state senate election
- Find out where to vote in the state senate election
- Where am i supposed to vote in the state senate election
- Where can i get information on where to vote in the state Senate election?

Figure 17: FLAN can be used in a zero-shot fashion to generate data, such as utterances that are consistent with a given intent. Multiple FLAN outputs are generated via random sampling with a temperature of 0.9 and top k of 40.

Input:
Suggest a related search term to "watermelon"

FLAN output:

- Cantaloupe
- Cucumber
- Melon
- Fruit
- Watermelon juice
- Watermelons
- Lemon
- Apple

Input:
Suggest a related search term to "Novak Djokovic"

FLAN output:

- novak djokovic girlfriend
- serena williams
- rafael nadal
- novak djokovic wife
- roger federer
- andy murray
- novak djokovic news
- novak djokovic serbia
- novak djokovic wimbledon

Figure 18: FLAN can be used for zero-shot query expansion. Multiple FLAN outputs are generated via random sampling with a temperature of 0.9 and top k of 40.

Input:
On the way to Yosemite, our front tire blew.
Finish writing the story, and include a happy ending.

FLAN output (cherrypicked):

- We got our tire replaced. The mechanic was honest, kind and helpful. Then we had a smooth ride for the rest of the trip. Our trip was filled with fun and excitement.
- We waited three hours for the tow truck, but had a great picnic. The tow truck driver helped us put the spare on, and we arrived in Yosemite for our reservations dinner, with time to spare.

Input:
"I guess I need to do laundry."
Rewrite the above sentence in a Shakespearean style.

FLAN output:

- To my mind, verily, I suppose 'tis time, I wash my clothes.
- Mine eyes and ears and tongue declare me idle; and yet it is true I guess, By my garments, that I need to do laundry
- Lo my duty now calls To thee, to wash this body's clothes.

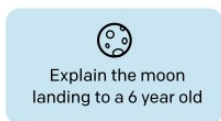
Figure 19: FLAN can perform zero-shot tasks relevant to assisted-writing applications. Multiple FLAN outputs are generated via random sampling with a temperature of 0.9 and top k of 40.

Aligned Instruction Tuning Process (InstructGPT)

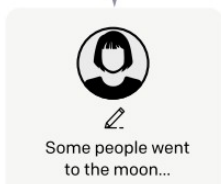
Step 1

Collect demonstration data, and train a supervised policy.

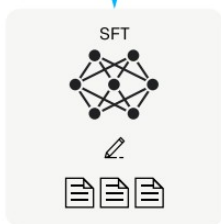
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



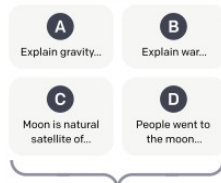
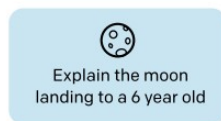
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

Collect comparison data, and train a reward model.

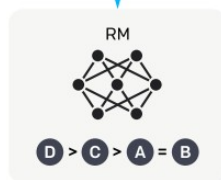
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



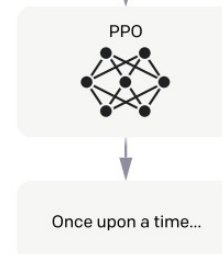
Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



Also known as Reinforcement Learning from Human Feedback: **RLHF**

Ouyang et al., 2022;


<https://arxiv.org/pdf/2203.02155.pdf>

Instruction Tuning

Takeaway: Almost any off-the-shelf LLM you use today will be instruction tuned and also likely aligned with RLHF.

If using open-source models, be careful:

The same model often comes as “base”, “instruct”, and/or “chat” variants:

 Hugging Face

Models 207

llama-2-70b

[meta-llama/Llama-2-70b-chat-hf](#)
Text Generation • Updated 1 day ago • ↓ 1.7M • ♥ 1.6k

← Instruction tuned, RLHF'd, ready to serve you as your personal assistant.

[meta-llama/Llama-2-70b-hf](#)
Text Generation • Updated 1 day ago • ↓ 146k • ♥ 658

← Wild, untamed autocomplete engine!

Agenda

Introduction

Recap: How do LLMs Work?

Instruction Tuning

Chain-of-Thought (CoT)

Retrieval Augmented Generation
(RAG)

LLM Agents

Discussion

Chain-of-Thought (CoT)

Ok so, LLMs are *not* just overparameterized autocomplete models. LLMs can follow instructions.

But, they are pretty awful at math:

Chain-of-thought Reasoning (Wei et al., 2022)

Standard Prompting

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. ❌

Source: <https://arxiv.org/pdf/2201.11903.pdf>

Chain-of-Thought (CoT)

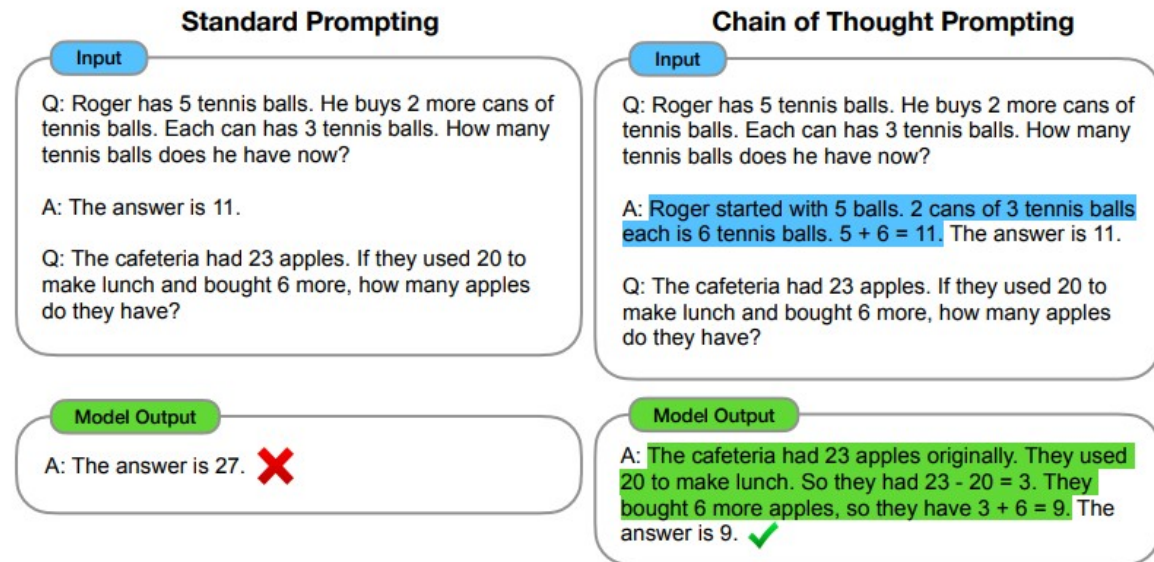
Ok so, LLMs are *not* just overparameterized autocomplete models. LLMs can follow instructions.

But, they are pretty awful at math: al., 2022)

... Or not, if the problem can be broken down into simple steps!

Chain-of-Thought (CoT) is a prompting technique to elicit step-by-step reasoning in LLMs.

Chain-of-thought Reasoning (Wei et



Source: <https://arxiv.org/pdf/2201.11903.pdf>

Chain-of-Thought (CoT)

In 2022 it was discovered that the largest LLMs at the time (e.g., GPT-3) could be prompted to do zero-shot CoT using a simple “incantation”:

(a) Few-shot

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: 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:

(Output) *The answer is 8. ❌*

(c) Zero-shot

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: The answer (arabic numerals) is

(Output) *8 ❌*

(b) Few-shot-CoT

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: 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:

(Output) *The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✅*

(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. ✅*

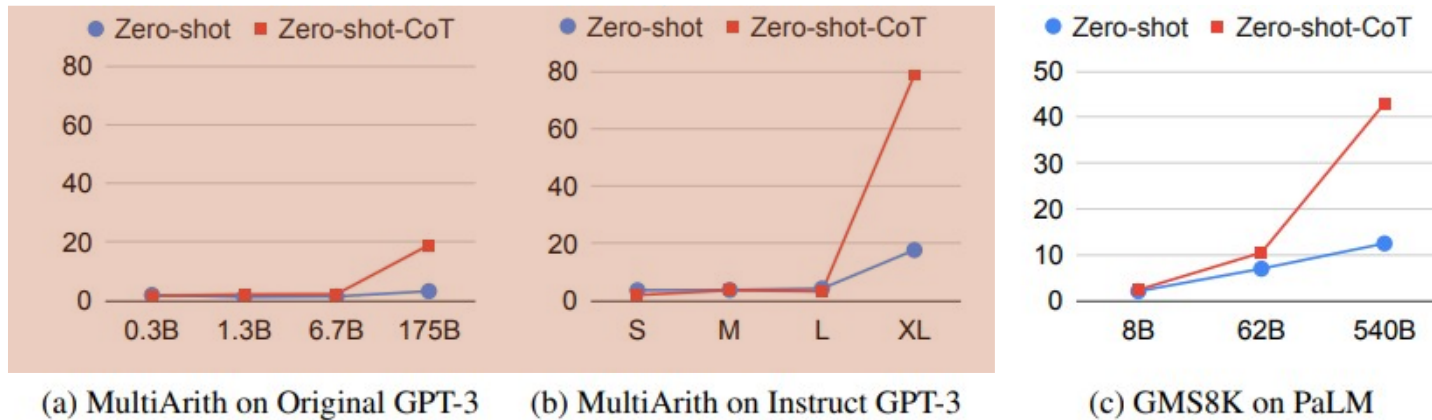
The “magic” incantation:
“Let’s think step by step”

<https://arxiv.org/pdf/2205.11916.pdf>

Large Language Models are Zero-Shot Reasoners (Kojima et al., NeurIPS 2022)

Chain-of-Thought (CoT)

In 2022 it was discovered that the largest LLMs at the time (e.g., GPT-3) could be prompted to do zero-shot CoT using a simple “incantation”:



The “magic” incantation:
“Let’s think step by step”

Figure 3: Model scale study with various types of models. S: text-ada-001, M: text-babbage-001, L: text-curie-001, XL: text-davinci-002. See Appendix A.3 and E for the detail.

Critically, the technique is dramatically more effective on InstructGPT than base GPT-3!

<https://arxiv.org/pdf/2205.11916.pdf>

Large Language Models are Zero-Shot Reasoners (Kojima et al., NeurIPS 2022)

Chain-of-Thought (CoT)

Other similar phrases yielded similar results:

Table 4: Robustness study against template measured on the MultiArith dataset with text-davinci-002. (*1) This template is used in Ahn et al. [2022] where a language model is prompted to generate step-by-step actions given a high-level instruction for controlling robotic actions. (*2) This template is used in Reynolds and McDonell [2021] but is not quantitatively evaluated.

No.	Category	Template	Accuracy
1	instructive	Let's think step by step.	78.7
2		First, (*1)	77.3
3		Let's think about this logically.	74.5
4		Let's solve this problem by splitting it into steps. (*2)	72.2
5		Let's be realistic and think step by step.	70.8
6		Let's think like a detective step by step.	70.3
7		Let's think	57.5
8		Before we dive into the answer,	55.7
9		The answer is after the proof.	45.7
10	misleading	Don't think. Just feel.	18.8
11		Let's think step by step but reach an incorrect answer.	18.7
12		Let's count the number of "a" in the question.	16.7
13		By using the fact that the earth is round,	9.3
14	irrelevant	By the way, I found a good restaurant nearby.	17.5
15		AbraKadabra!	15.5
16		It's a beautiful day.	13.1
-		(Zero-shot)	17.7

<https://arxiv.org/pdf/2205.11916.pdf>

Large Language Models are Zero-Shot Reasoners (Kojima et al., NeurIPS 2022)

Chain-of-Thought (CoT)

Works for a variety of reasoning tasks, not just math:

Table 1: Accuracy comparison of Zero-shot-CoT with Zero-shot on each tasks. The values on the left side of each task are the results of using answer extraction prompts depending on answer format as described at § 3. The values on the right side are the result of additional experiment where standard answer prompt "The answer is" is used for answer extraction. See Appendix A.5 for detail setups.

	Arithmetic					
	SingleEq	AddSub	MultiArith	GSM8K	AQUA	SVAMP
zero-shot	74.6/ 78.7	72.2/77.0	17.7/22.7	10.4/12.5	22.4/22.4	58.8/58.7
zero-shot-cot	78.0/78.7	69.6/74.7	78.7/79.3	40.7/40.5	33.5/31.9	62.1/63.7
	Common Sense		Other Reasoning Tasks		Symbolic Reasoning	
	Common SenseQA	Strategy QA	Date Understand	Shuffled Objects	Last Letter (4 words)	Coin Flip (4 times)
zero-shot	68.8/72.6	12.7/ 54.3	49.3/33.6	31.3/29.7	0.2/-	12.8/53.8
zero-shot-cot	64.6/64.0	54.8/52.3	67.5/61.8	52.4/52.9	57.6/-	91.4/87.8

<https://arxiv.org/pdf/2205.11916.pdf>

Large Language Models are Zero-Shot Reasoners (Kojima et al., NeurIPS 2022)

Chain-of-Thought (CoT)

Common-senseQA

Q: If there is a pond with trees around it, where is it likely located? Answer Choices: (A) ground (B) bathroom (C) forest (D) countryside (E) rural area

A: Let's think step by step. A pond is likely to be located near trees because trees need water to survive. Therefore, the most likely location for a pond with trees around it is in a forest. Therefore, among A through E, the answer is C. ✓

StrategyQA

Q: Would JPEG be a good format for saving an image of Da Vinci's Vitruvian Man?

A: Let's think step by step. The JPEG format is a lossy compression format, which means that some of the image data will be lost when the image is compressed. The Vitruvian Man is a very detailed image, with lots of small details. So, when we compress the image using JPEG, some of those small details will be lost, and the image will not be as sharp and clear as the original. So, in conclusion, JPEG is not the best format for saving an image of the Vitruvian Man. Therefore, the answer (Yes or No) is No. ✓

<https://arxiv.org/pdf/2205.11916.pdf>

Large Language Models are Zero-Shot Reasoners (Kojima et al., NeurIPS 2022)

Chain-of-Thought (CoT)

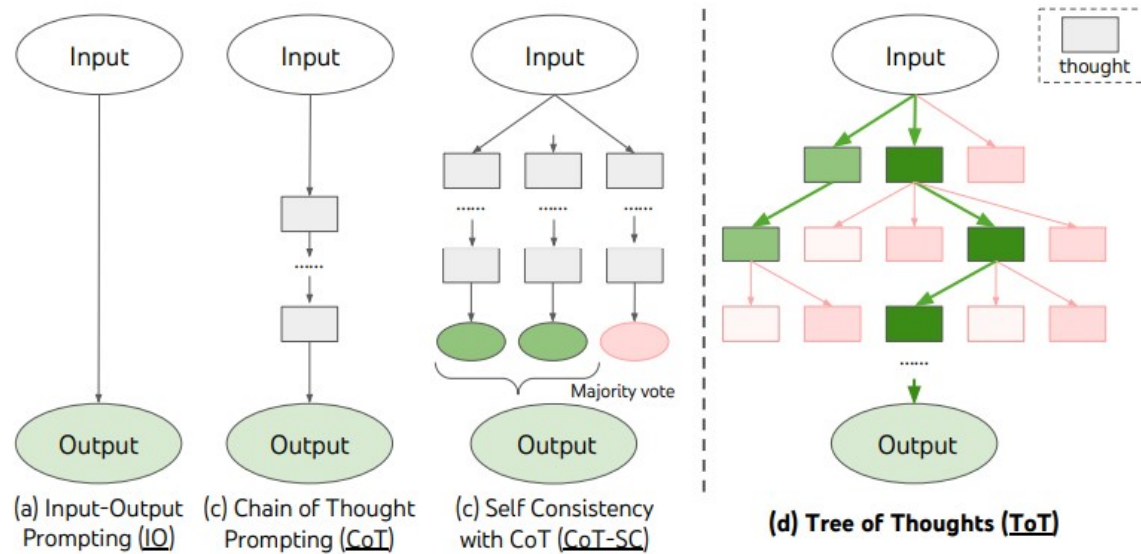
Question	Q: For Halloween Megan received 11 pieces of candy from neighbors and 5 pieces from her older sister. If she only ate 8 pieces a day, how long would the candy last her?
Zero-Shot	A: The answer (arabic numerals) is 3 days. ✗
Zero-shot-CoT (1)	A: Let's think step by step. Megan received 11 pieces of candy from neighbors and 5 pieces from her older sister. That means she has a total of 16 pieces of candy. If she only ate 8 pieces a day, she would have to divide her candy into two days. That means she would have 8 pieces of candy for Day 1 and 8 pieces of candy for Day 2. Therefore, the answer (arabic numerals) is 2. ✓

<https://arxiv.org/pdf/2205.11916.pdf>

Large Language Models are Zero-Shot Reasoners (Kojima et al., NeurIPS 2022)

Chain-of-Thought (CoT)

Recently, authors have proposed improvements to CoT:

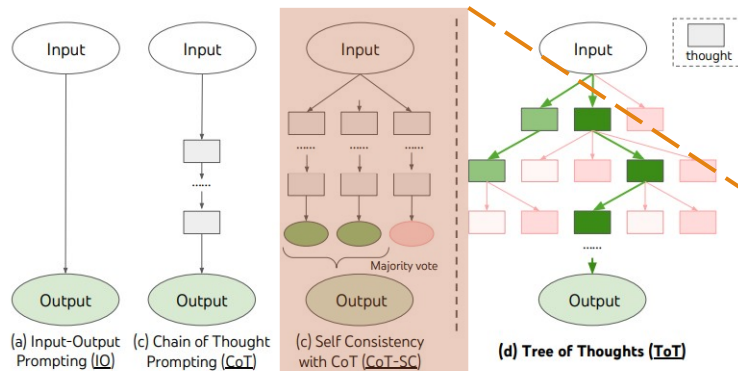


Tree of Thoughts: Deliberate Problem Solving with Large Language Models (Yao et al., 2023)
<https://arxiv.org/pdf/2305.10601.pdf>

Figure 1: Schematic illustrating various approaches to problem solving with LLMs. Each rectangle box represents a *thought*, which is a coherent language sequence that serves as an intermediate step toward problem solving. See concrete examples of how thoughts are generated, evaluated, and searched in Figures 2,4,6.

Chain-of-Thought (CoT)

Recently, authors have proposed improvements to CoT:

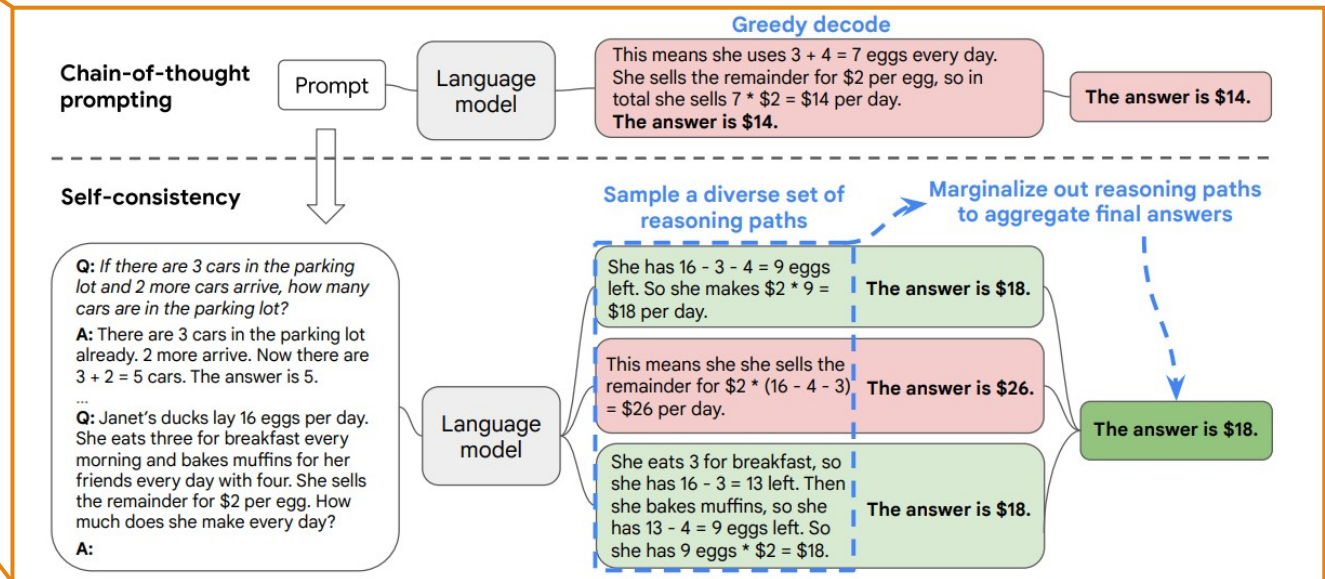


SELF-CONSISTENCY IMPROVES CHAIN OF THOUGHT REASONING IN LANGUAGE MODELS (Wang et al., ICLR 2023)

<https://arxiv.org/pdf/2203.11171.pdf>

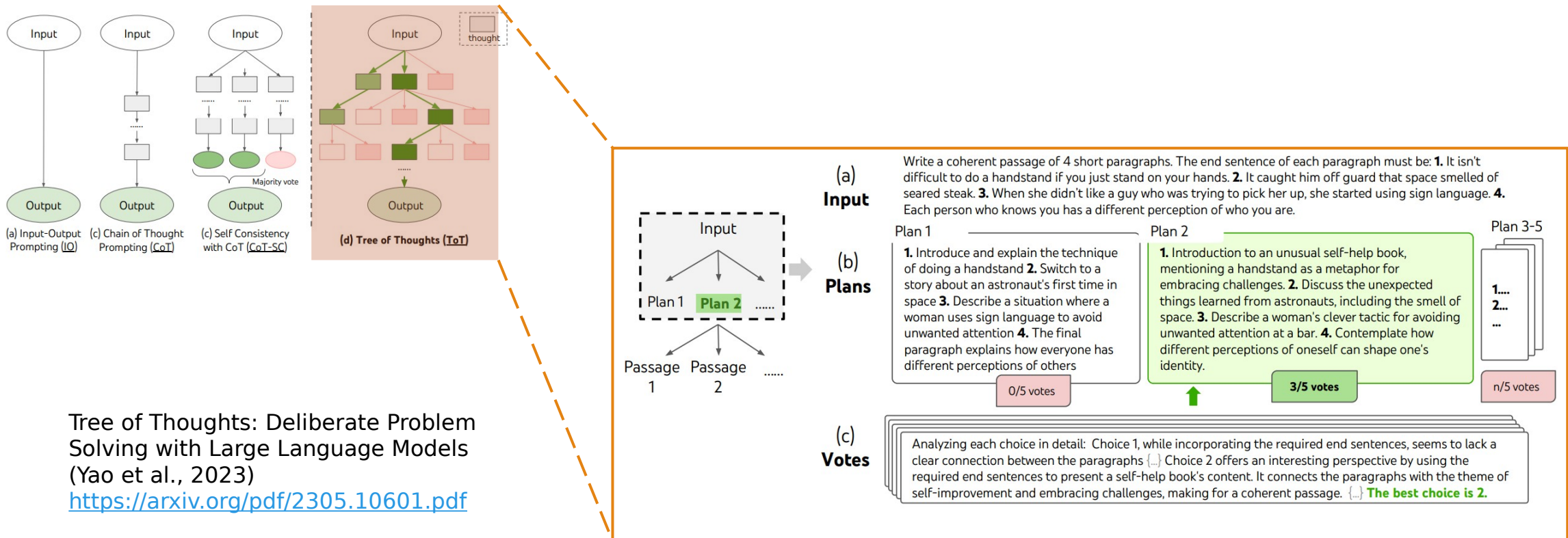
	GSM8K	MultiArith	AQuA	SVAMP	CSQA	ARC-c
Greedy decode	56.5	94.7	35.8	79.0	79.0	85.2
Weighted avg (unnormalized)	56.3 ± 0.0	90.5 ± 0.0	35.8 ± 0.0	73.0 ± 0.0	74.8 ± 0.0	82.3 ± 0.0
Weighted avg (normalized)	22.1 ± 0.0	59.7 ± 0.0	15.7 ± 0.0	40.5 ± 0.0	52.1 ± 0.0	51.7 ± 0.0
Weighted sum (unnormalized)	59.9 ± 0.0	92.2 ± 0.0	38.2 ± 0.0	76.2 ± 0.0	76.2 ± 0.0	83.5 ± 0.0
Weighted sum (normalized)	74.1 ± 0.0	99.3 ± 0.0	48.0 ± 0.0	86.8 ± 0.0	80.7 ± 0.0	88.7 ± 0.0
Unweighted sum (majority vote)	74.4 ± 0.1	99.3 ± 0.0	48.3 ± 0.5	86.6 ± 0.1	80.7 ± 0.1	88.7 ± 0.1

Table 1: Accuracy comparison of different answer aggregation strategies on PaLM-540B.



Chain-of-Thought (CoT)

Recently, authors have proposed improvements to CoT:



Agenda

Introduction

Recap: How do LLMs Work?

Instruction Tuning

Chain-of-Thought (CoT)

Retrieval Augmented Generation
(RAG)

LLM Agents

Discussion

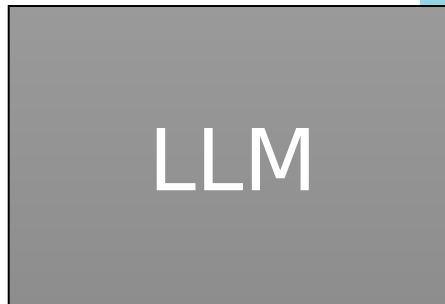
Retrieval Augmented Generation (RAG)

Ok so, LLMs can follow instructions & use step-by-step reasoning to plan the best response.

But, they still hallucinate facts:



I'm planning to visit Palm Island in the Grenadines. Can I drive there?



Yes, you can drive over the Palm Island Bridge from Clifton to Palm Island, St. Vincent & Grenadines.

Retrieval Augmented Generation (RAG)

Ok so, LLMs can follow instructions & use step-by-step reasoning to plan the best response.

I'm planning to visit Palm Island in the Grenadines. Can I drive there?

Palm Island in the Grenadines is a small island one mile from Union Island, only accessible by boat.



3. Tokens of correct information become more likely in the autocomplete process!
No, Palm Island in the Grenadines is only accessible by boat.

1. Query Knowledge Source

Palm Island, Grenadines

From Wikipedia, the free encyclopedia



Palm Island in the **Grenadines** is a small island one mile from Union Island, **only accessible by boat**. Originally known as *Prune Island*, Palm Island got its current name when the former owners, the late ... of **coconut palms** (*Cocos nucifera*), transforming the deserted, **swampy**, and **mosquito** infested island of more palms following the construction of a 752-metre (2,467 ft) concrete airstrip on Union Island. It is a nursery for young palms.

2. Append to context

This process is called Retrieval-Augmented Generation (RAG)!

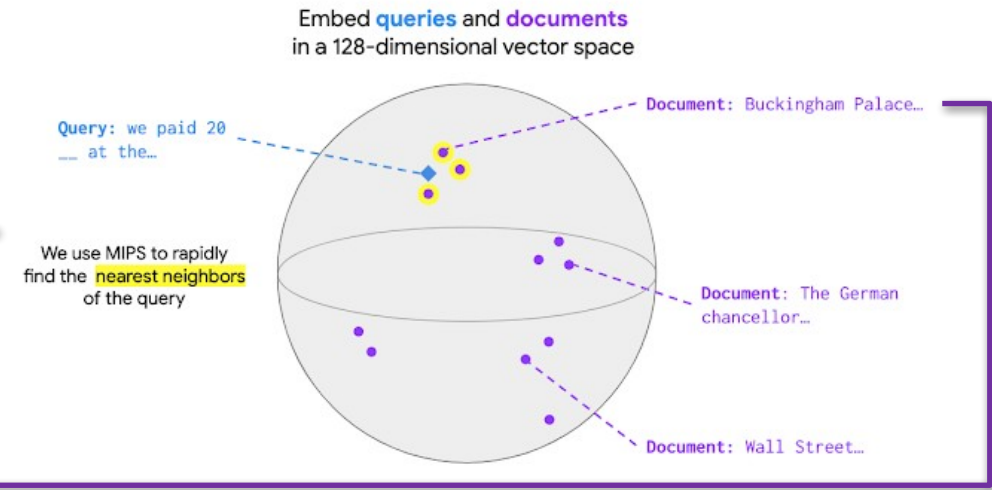
Retrieval Augmented Generation (RAG)

Approaches RAG include:

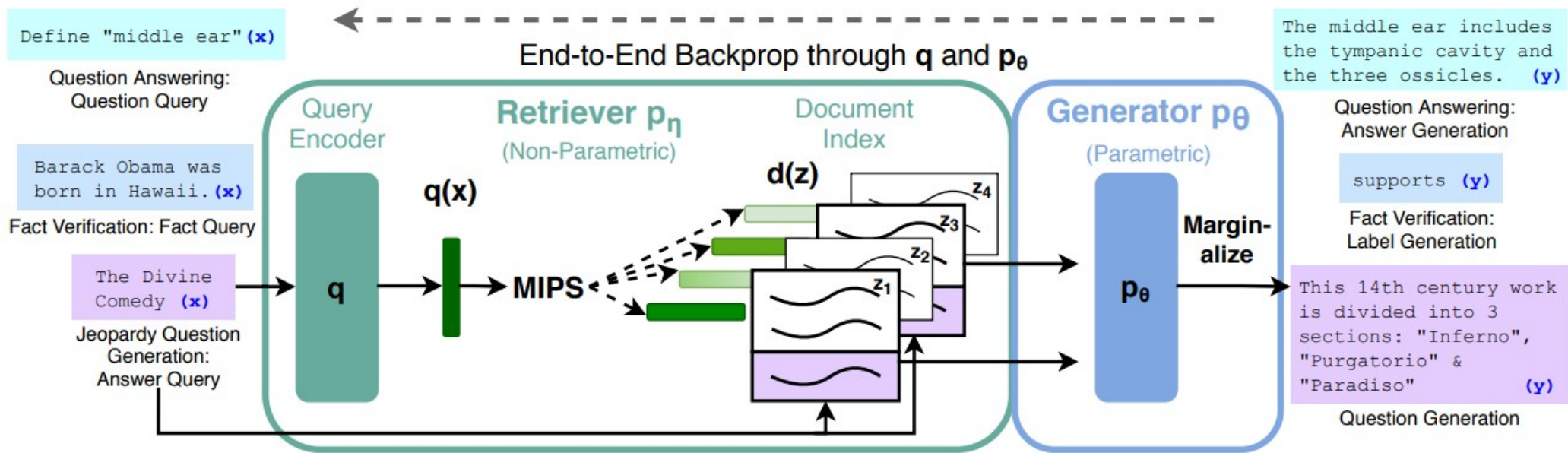
1. Generating queries against a vector-space IR system

- a. **REALM:** Retrieval-Augmented LM Pre-training (Guu et al., 2020)
- b. **RAG:** Retrieval-Augmented Generation (Lewis et al., 2020) ← (origin of term “RAG”)
- c. **DPR:** Dense Passage Retrieval (Karpukl
- d. **FiD:** Fusion-in-Decoder (Izacard & Grav
- e. **RETRO:** Retrieval-Enhanced Transformer (Borgeaud et al., 2021) →

We paid twenty __ at the Buckingham Palace gift shop.
[sep] Buckingham Palace is the London residence of the British monarchy.



Retrieval Augmented Generation (RAG)



$$p_{\text{RAG-Sequence}}(y|x) \approx \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x, z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i p_\theta(y_i|x, z, y_{1:i-1})$$

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (Lewis et al., NeurIPS 2020)

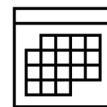
<https://arxiv.org/abs/2005.11401>

Retrieval Augmented Generation (RAG)

Recent, more capable instruction-tuned LLMs have allowed a simpler form of RAG to take hold: **Tool Use**.

For example:

- a) LaMDA (Thoppilan et al., 2022)
- b) Toolformer (Schick et al., 2023)
Toolformer has only 6.7b parameters but **outperforms GPT-3** (175b params) on Q&A and tasks requiring mathematical or temporal reasoning.



The New England Journal of Medicine is a registered trademark of [QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society] the MMS.

Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

The name derives from "la tortuga", the Spanish word for [MT("tortuga") → turtle] turtle.

The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

Tool Use is a generalization of RAG, going beyond traditional information retrieval and including all kinds of external functions!

Source:

<https://arxiv.org/pdf/2302.04761.pdf>

Agenda

Introduction

Recap: How do LLMs Work?

Instruction Tuning

Chain-of-Thought (CoT)

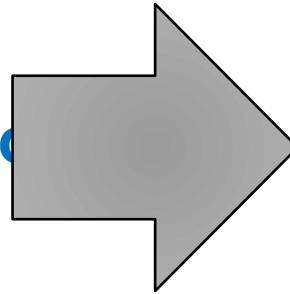
Retrieval Augmented Generation
(RAG)

LLM Agents

Discussion

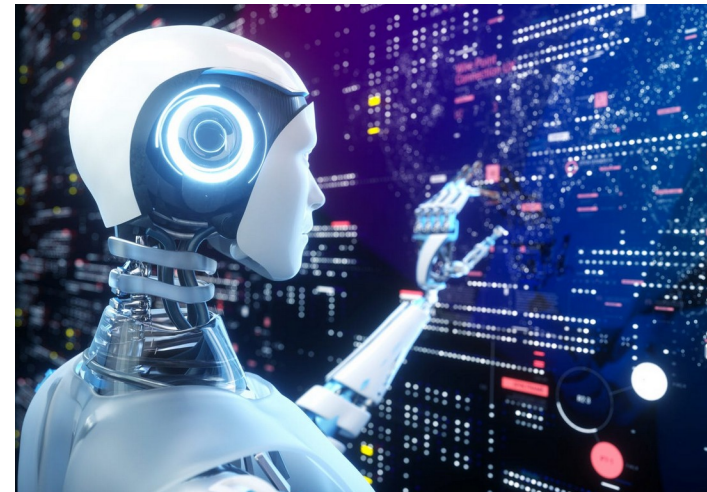
LLM Agents

- ✓ ○ **Scale (# parameters, # tokens)**
- ✓ ○ **Instruction Tuning**
- ✓ ○ **Alignment (RLHF)**
- ✓ ○ **Advanced Prompting (CoT)**
- ✓ ○ **Retrieval Augmentation (RAG)**
- ✓ ○ **Tool Use**
- **Perception-Action loop**



Perception-Action loop brings it all together!

Autonomous LLM Agents!



[Image Source](#)
[e](#)

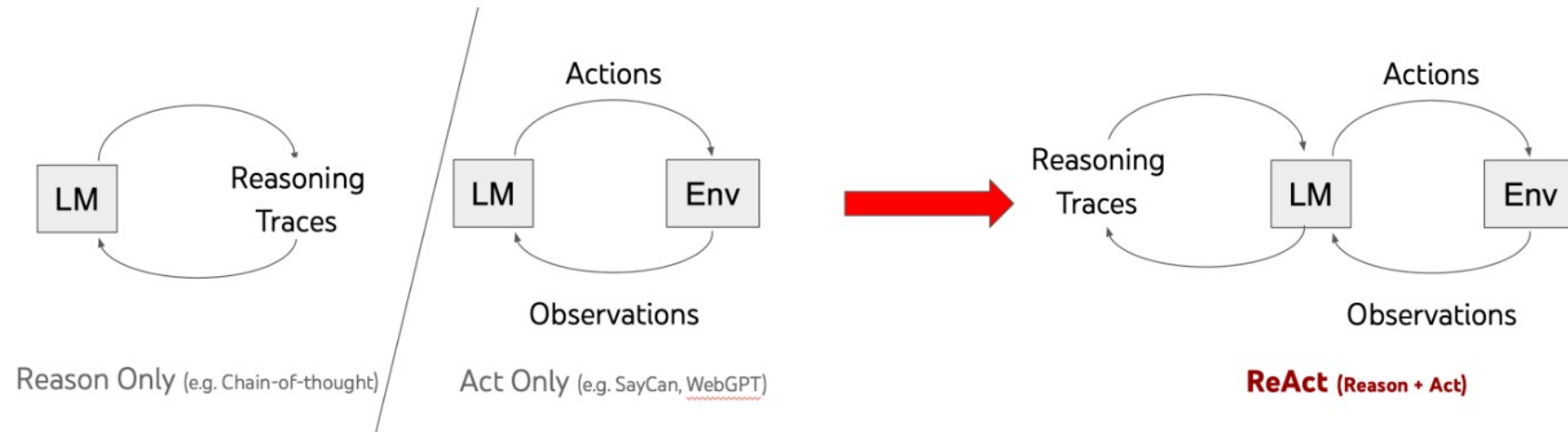
LLM Agents

<https://react-lm.github.io/>

ReAct: Synergizing Reasoning and Acting in Language Models

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, Yuan Cao

[\[Paper\]](#) [\[Code\]](#) [\[Blogpost\]](#) [\[BibTex\]](#)



Language models are getting better at reasoning (e.g. chain-of-thought prompting) and acting (e.g. WebGPT, SayCan, ACT-1), but these two directions have remained separate.

ReAct asks, what if these two fundamental capabilities are combined?

LLM Agents ⁿ Following Instruction

Tool Use for RAG!

CoT!

(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

(1a) Standard

Answer: iPod



(1b) CoT (Reason Only)

Thought: Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

Answer: iPhone, iPad, iPod Touch



(1c) Act-Only

Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control ...

Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ...

Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software ...

Act 4: Finish[yes]



(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

Thought 3: Front Row is not found. I need to search Front Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software ...

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
Act 4: Finish[keyboard function keys]



REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS (Yao et al., ICLR 2023)

<https://arxiv.org/pdf/2210.03629.pdf>

LLM Agents

ALFWorld: <https://alfworld.github.io/>

(2) AlfWorld

You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 1, a coffee machine 1, a countertop 3, a stove burner 1, and a toaster 1.
Your task is to: Put some pepper shaker on a drawer.

(2a) Act-Only

Act 1: Go to drawer 1
Obs 1: The drawer 1 is closed.

Act 2: Open drawer 1
Obs 2: You open the drawer 1. The drawer 1 is open. In it, you see a dish sponge 2, and a spoon 1.

Act 3: Go to sinkbasin 1
Obs 3: On the sinkbasin 1, you see a dish sponge 3, a spatula 1, and a spoon 2.

Act 4: Take peppershaker 1 from sinkbasin 1
Obs 4: Nothing happens.

Act 5: Take peppershaker 1 from sinkbasin 1
Obs 5: Nothing happens.



(2b) ReAct (Reason + Act)

Act 1: **Think**[First I need to find a pepper shaker...more likely to appear in cabinets (1-6), countertops (1-3), ...]
Act 2: Go to cabinet 1
Obs 2: On the cabinet 1, you see a vase 2.

(Here the agent go to cabinet 1, then cabinet 2, 3, then countertop 1 and 2)

Act 6: Go to countertop 3
Obs 6: On the countertop 3, you see a apple 1, a bread 1, a pepper shaker 1, and a vase 3.

Act 7: Take pepper shaker 1 from countertop 3
Obs 7: You pick up the pepper shaker 1 from the countertop 3.

Act 8: **Think**[Now I find a pepper shaker 1. Next, I need to put it in/on drawer 1.]
Act 9: Go to drawer 1
Obs 9: Drawer 1 is closed.

Act 10: Open drawer 1
Obs 10: You open Drawer 1 ...

Act 11: Put pepper shaker 1 in/on drawer 1
Obs 11: You put pepper shaker 1 in/on the drawer 1.



REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS (Yao et al., ICLR 2023)

<https://arxiv.org/pdf/2210.03629.pdf>

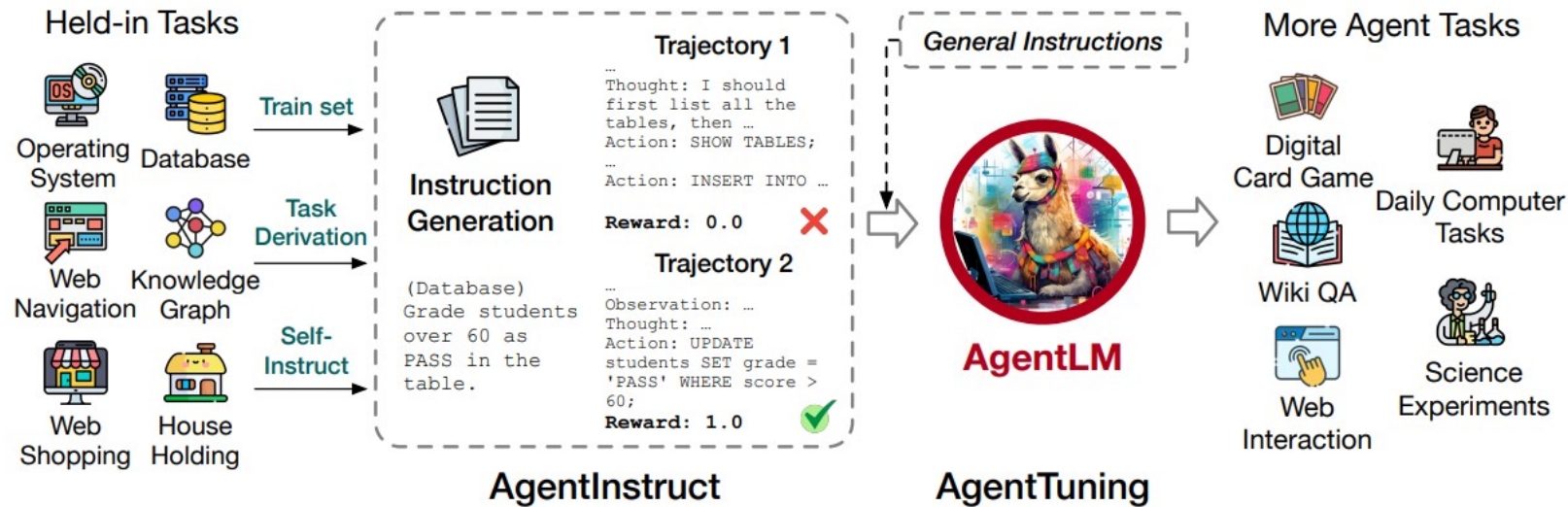
LLM Agents

Prompting Method	HotpotQA (question answering, exact match, 6-shot)	FEVER (fact verification, accuracy, 3-shot)	ALFWorld (text-based game, succ rate, 2-shot)	WebShop (web interaction, succ rate, 1-shot)
Standard (IO)	28.7	57.1	N/A (cannot act)	N/A (cannot act)
Reason-only (CoT)	29.4	56.3	N/A (cannot act)	N/A (cannot act)
Act-only	25.7	58.9	45	30.1
Best ReAct method	35.1	64.6	71	40
Supervised/Imitation Learning SoTA	67.5 (140k samples)	89.5 (90k samples)	37 (100k samples)	29.1 (90k samples)

<https://react-lm.github.io/>

LLM Agents

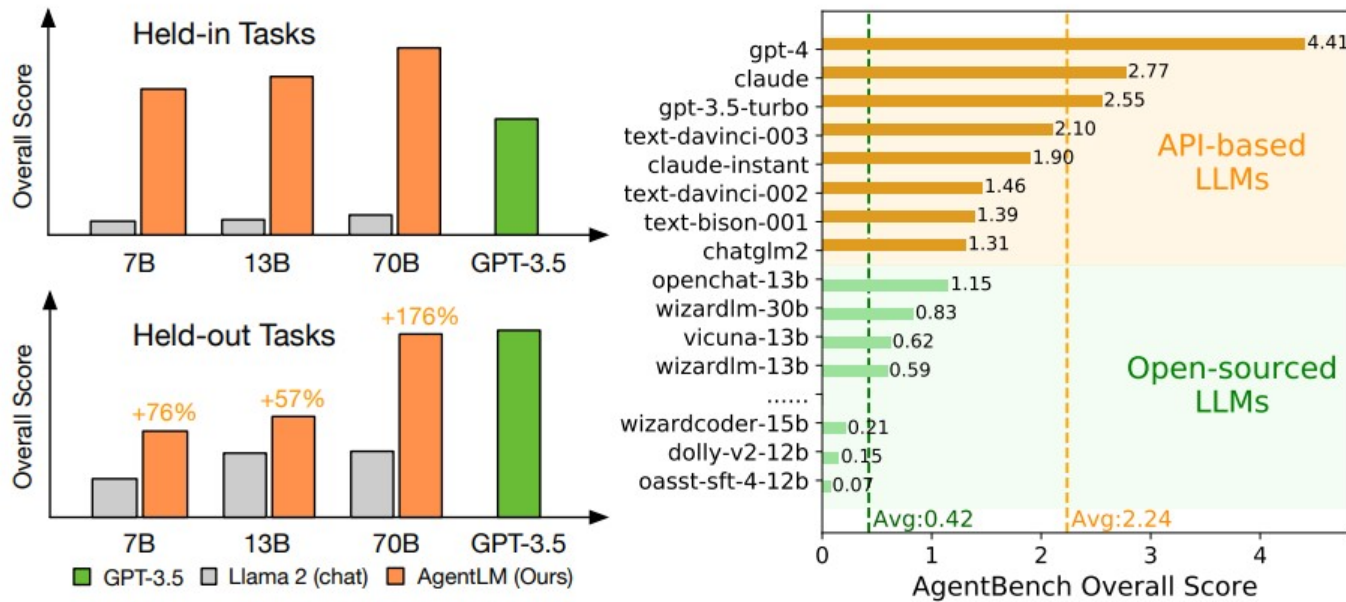
AgentInstruct is an instruction-tuning dataset containing multi-step ReAct trajectories for a variety of tasks that require CoT and tool use!



<https://arxiv.org/pdf/2310.12823.pdf>
AGENTTUNING: ENABLING GENERALIZED AGENT ABILITIES FOR LLMs (Zeng et al., 2023)

Figure 2: **An overview of AgentInstruct and AgentTuning.** The construction of AgentInstruct, consisting of instruction generation, trajectory interaction, and trajectory filter. AgentLM is fine-tuned using a mixture of AgentInstruct and general-domain instructions.

LLM Agents



(a) Overall score in our held-in and held-out tasks. (b) Closed & open LLMs on agent tasks (Liu et al., 2023)

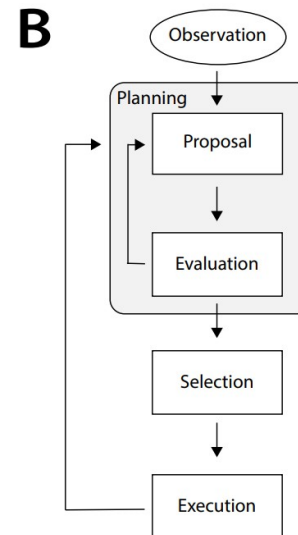
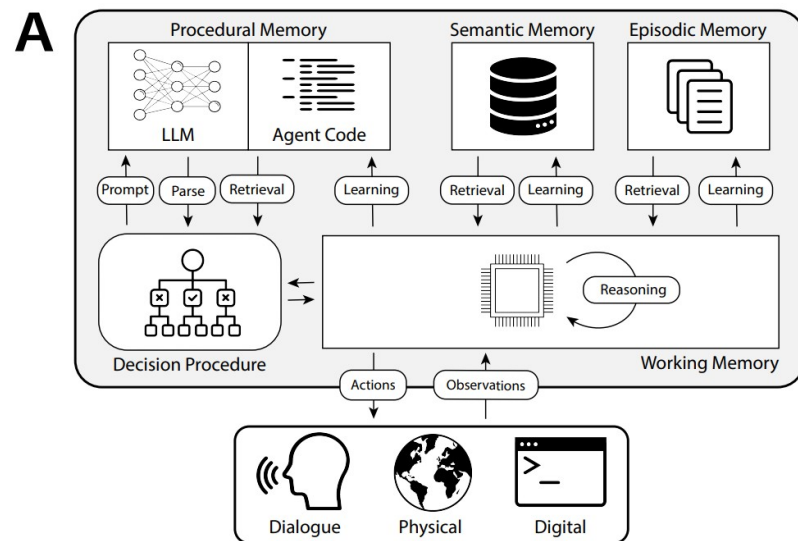
Figure 1: (a) **AgentLM exhibits superior performance.** AgentLM is a series of models fine-tuned on the foundation of Llama 2 chat. Moreover, its generalization capability on held-out tasks is on par with GPT-3.5; (b) This figure is directly re-printed from AgentBench (Liu et al., 2023) with permission. **Open LLMs significantly underperforms API-based LLMs.**

<https://arxiv.org/pdf/2310.12823.pdf>
AGENTTUNING: ENABLING
GENERALIZED AGENT ABILITIES FOR
LLMS (Zeng et al., 2023)

LLM Agents

Agent Takeaways:

- LLM Agents tackle far more complex problems than previously possible.
- They combine instruction following, reasoning, and tool use in a step-by-step loop.
- The agent’s “short-term memory” is its context;
- The agent’s “long-term memory” is its knowledge retrieved via RAG (tools)



There are already efforts to formalize LLM agent components into cognitive architectures...

Cognitive Architectures for Language Agents (Sumers et al., 2023)

<https://arxiv.org/pdf/2309.02427.pdf>

Agenda

Introduction

Recap: How do LLMs Work?

Instruction Tuning

Chain-of-Thought (CoT)

Retrieval Augmented Generation
(RAG)

LLM Agents

Discussion

Discussion

So, you want to build your own Agents? Here are some resources:

◦ <https://www.langchain.com/use-case/agents>



◦ <https://huggingface.co/docs/transformers/main/en/trans>



◦ <https://platform.openai.com/docs/assistants/overview>



◦ <https://microsoft.github.io/autogen/>



Build with OpenAI or open-source models (e.g., Llama-2)!

<https://www.promptingguide.ai/>

Thank You!

