Stanford | ENGINEERING

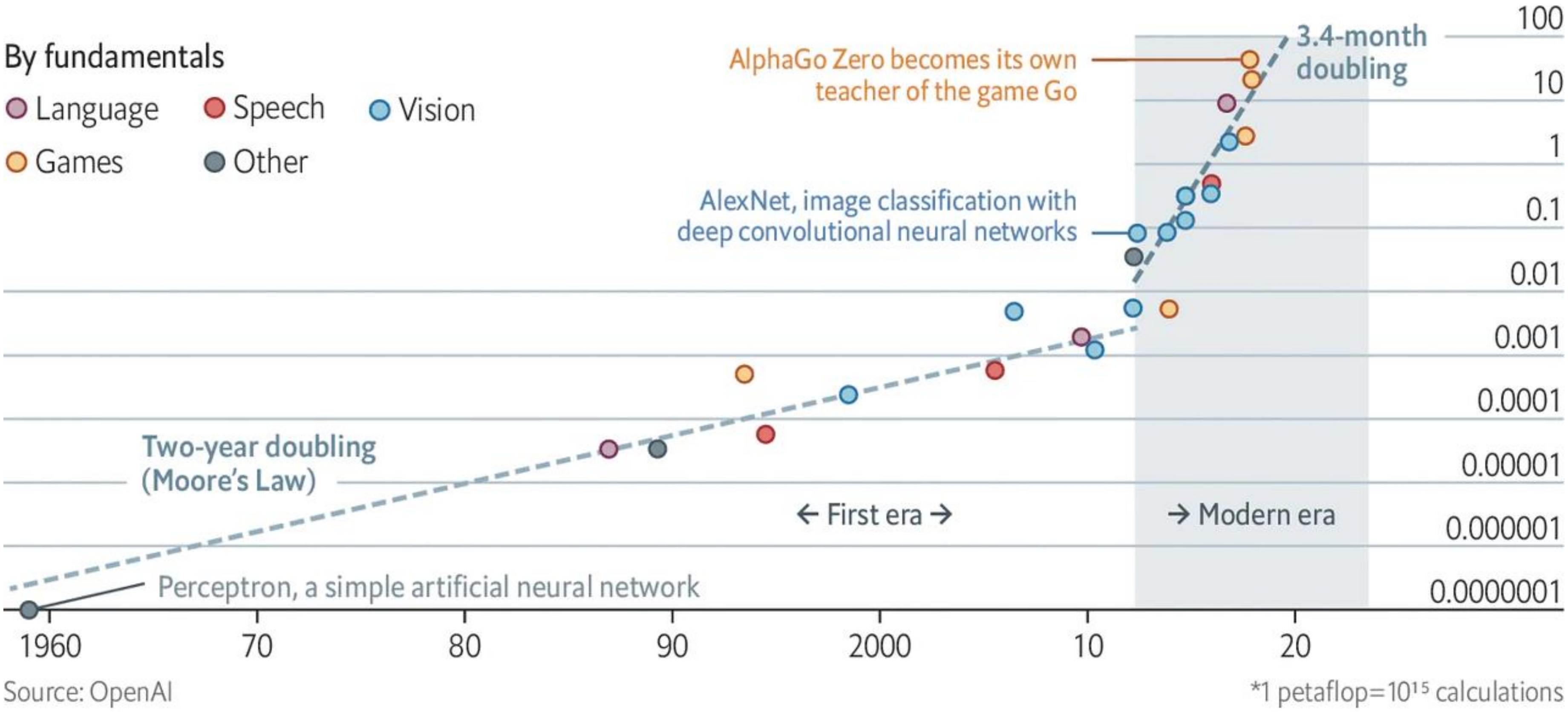
How to build Intelligent and Collaborative Agents

Shirley Wu
https://cs.stanford.edu/~shirwu/

CS224W Lecture

(The first part is based on slides in CS224N and CS224R)

Larger and larger models



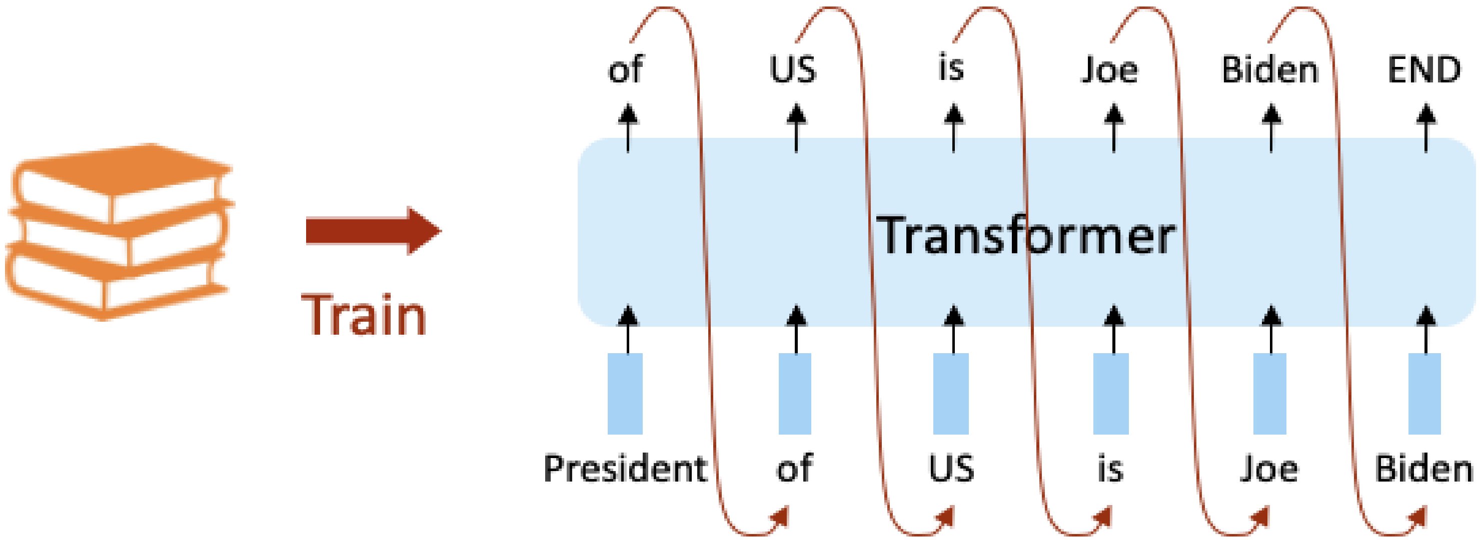
Training Large Language Models

Reinforcement Learning

Finetuning

Pretraining

Pretraining: LLMs are trained to predict the next token



Stanford University is located in ___________

Recap: What kinds of things does pretraining learn?

- *Stanford University is located in __________, California.* [Trivia]
- *I put ___ fork down on the table.* [syntax]
- *The woman walked across the street, checking for traffic over ___ shoulder.* [coreference]
- *I went to the ocean to see the fish, turtles, seals, and _____.* [lexical semantics/topic]
- *Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was ___.* [sentiment]

Shirley Wu (@ShirleyYXWu)

Recap: What kinds of things does pretraining learn?

- *Stanford University is located in __________, California.* [Trivia]
- *I put ___ fork down on the table.* [syntax]
- *The wom* … reference]
- *I went to* … topic]
- *Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was ___.* [sentiment]

Pretraining: Lots of text; learn general things!

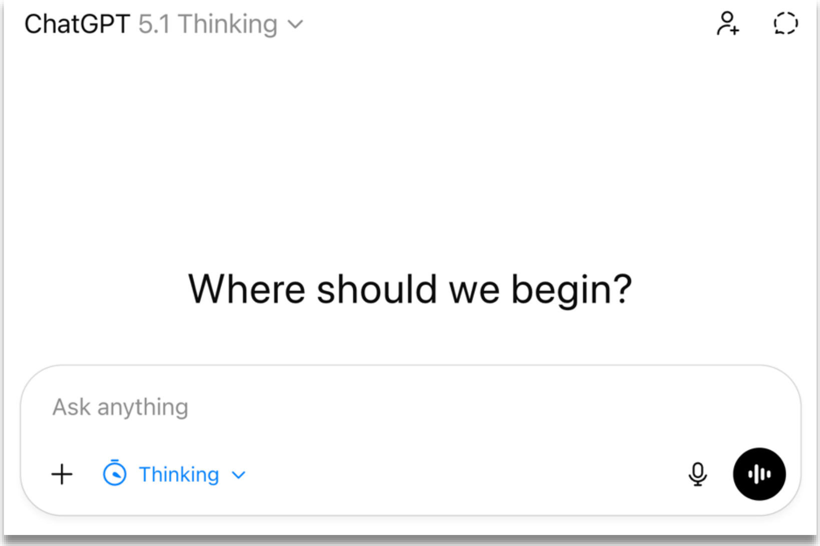
Make sure your model can process large-scale, diverse datasets

Language models as assistants?

How do we get from *this*

Stanford University is located in ___________

to *this*?



Language modeling ≠ assisting users

PROMPT	*Explain the moon landing to a 6 year old in a few sentences.*
COMPLETION	GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

Language models are not *aligned* with user intent [Ouyang et al., 2022].

Language modeling ≠ assisting users

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION **Human**

A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

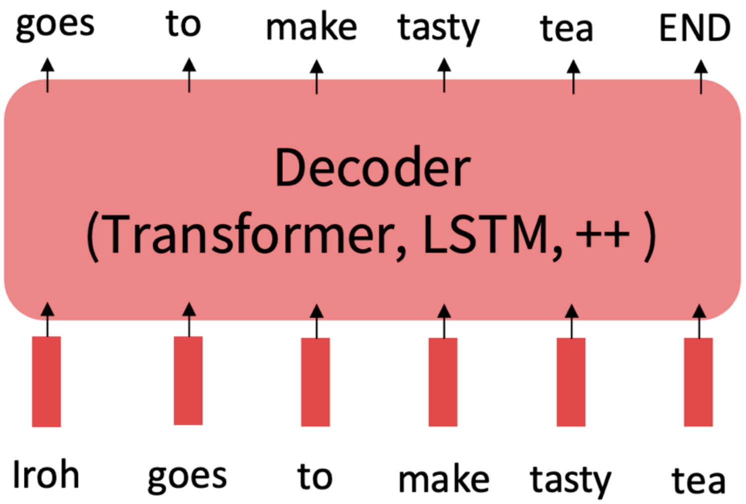
Language models are not *aligned* with user intent [Ouyang et al., 2022].

Finetuning to the rescue!

The Pretraining / Finetuning Paradigm

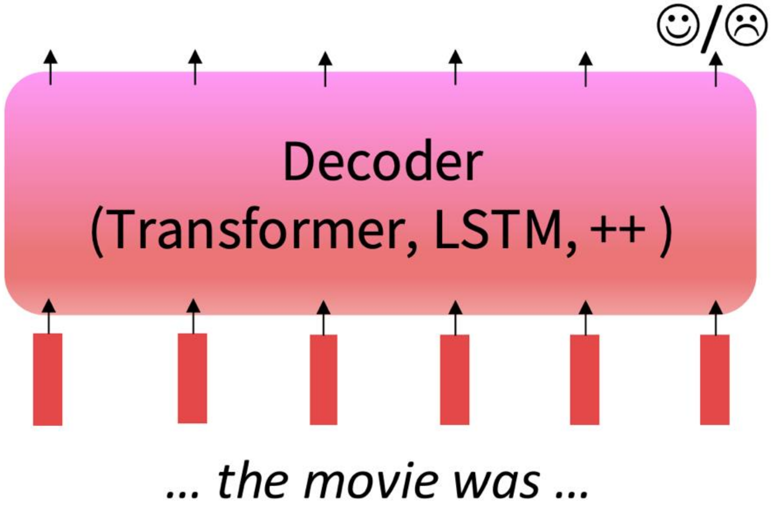
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



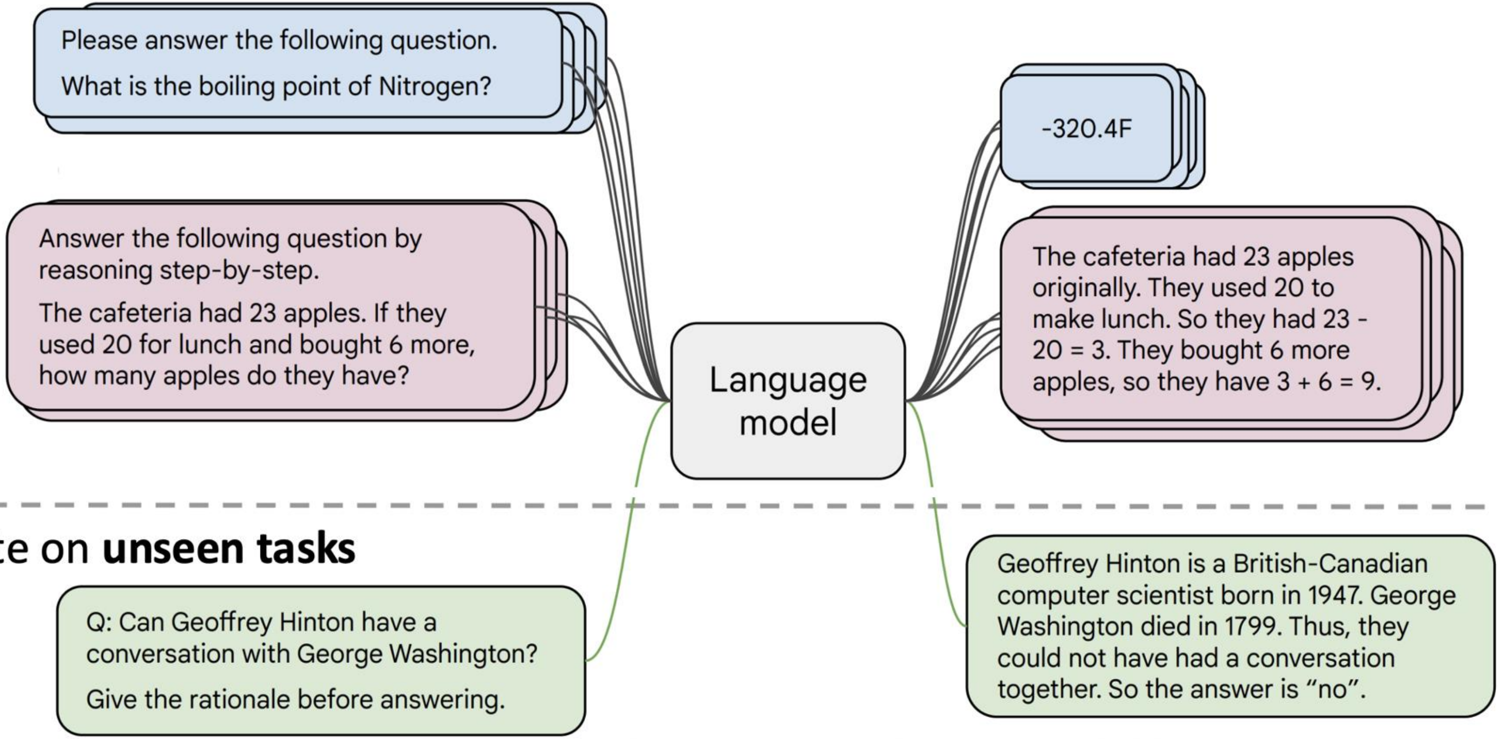
Step 2: Finetune (on your task)

Not many labels; adapt to the task!



Instruction finetuning

- **Collect examples** of (instruction, output) pairs across many tasks and finetune an LM



- Evaluate on **unseen tasks**

Instruction finetuning

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

Before instruction finetuning

The reporter and the chef will discuss their favorite dishes.
The reporter and the chef will discuss the reporter's favorite dishes.
The reporter and the chef will discuss the chef's favorite dishes.
The reporter and the chef will discuss the reporter's and the chef's favorite dishes.

✖ **(doesn't answer question)**

Instruction finetuning

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

After instruction finetuning

The reporter and the chef will discuss their favorite dishes does not indicate whose favorite dishes they will discuss. So, the answer is (C). ✅

Instruction finetuning: Improvements & Limitations

Instruction finetuning: Follows user instructions

Limitation 1:
Tasks like open-ended creative generation have no right answer.
E.g., Write me a story about a dog and her pet grasshopper.

Limitation 2:
Language modeling penalizes all token-level mistakes equally,
but some errors are worse than others.

Optimizing for human preferences

- Let's say we were training a language model on some task (e.g. summarization).
- For an instruction x and a LM sample y, imagine we had a way to obtain a *human reward* of that summary: $R(x, y) \in \mathbb{R}$, higher is better.

SAN FRANCISCO, California (CNN) -- A magnitude 4.2 earthquake shook the San Francisco ... overturn unstable objects.	An earthquake hit San Francisco. There was minor property damage, but no injuries.	The Bay Area has good weather but is prone to earthquakes and wildfires.
x	y_1	y_2
	$R(x, y_1) = 8.0$	$R(x, y_2) = 1.2$

- Now we want to maximize the expected reward of samples from our LM:

$$\mathbb{E}_{\hat{y} \sim p_\theta(y \mid x)}[R(x, \hat{y})]$$

[Schulman et al, 2017]

Optimizing for human preferences

- Let's say we were training a language model on some task (e.g. summarization).
- For an instruction x and a LM sample y, imagine we had a way to obtain a *human reward* of that summary: $R(x, y) \in \mathbb{R}$, higher is better.

```
SAN FRANCISCO,
...
overturn unstable
objects.
```

x

```
An earthquake hit
...
but no injuries.
```

y_1

$R(x, y_1) = 8.0$

```
The Bay Area has
...
wildfires.
```

y_2

$R(x, y_2) = 1.2$

But this requires a reward model!

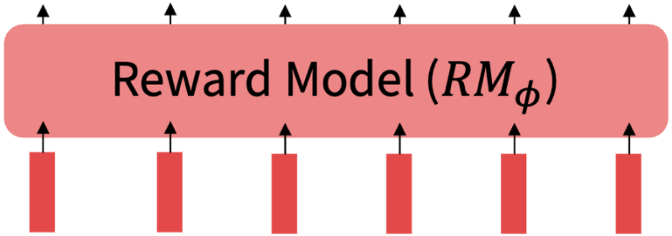
- Now we want to maximize the expected reward of samples from our LM:

$$\mathbb{E}_{\hat{y} \sim p_\theta(y \mid x)}[R(x, \hat{y})]$$

[Schulman et al, 2017]

Reward Model

$$R(x, y_2) = 1.2$$



SAN FRANCISCO,
California (CNN) --
A magnitude 4.2
earthquake shook the
San Francisco
...
overturn unstable
objects.

x

The Bay Area has
good weather but is
prone to
earthquakes and
wildfires.

y_2

How do we model human preferences?

Get a ranking based on human preference:

```
An earthquake hit
San Francisco.
There was minor
property damage,
but no injuries.
```

$>$

```
The Bay Area has
good weather but is
prone to
earthquakes and
wildfires.
```

s^w s^l

$$J_{RM}(\phi) = -\mathbb{E}_{(s^w, s^l) \sim D}\left[\log \sigma(RM_\phi(s^w) - RM_\phi(s^l))\right]$$

"winning" sample

"losing" sample

s^w should score higher than s^l

[Rafailov et al. 2023]

Reinforcement Learning from Human Preferences
: Optimizing the learned reward model

- We have the following:
  - A pretrained (possibly instruction-finetuned) LM $p^{PT}(y \mid x)$
  - A reward model $RM_\phi(x, y)$ that produces scalar rewards for LM outputs, trained on a dataset of human comparisons

- Now to do RLHF:
  - Copy the model $p_\theta^{RL}(y \mid x)$, with parameters θ we would like to optimize
  - We want to optimize:

$$\mathbb{E}_{\hat{y} \sim p_\theta^{RL}(\hat{y}|x)} \left[RM_\phi(x, \hat{y})\right]$$

Reinforcement Learning from Human Preferences

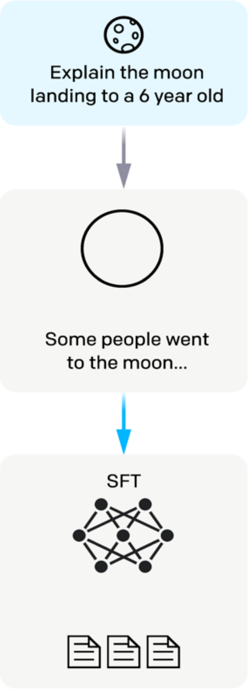
Step 1

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.



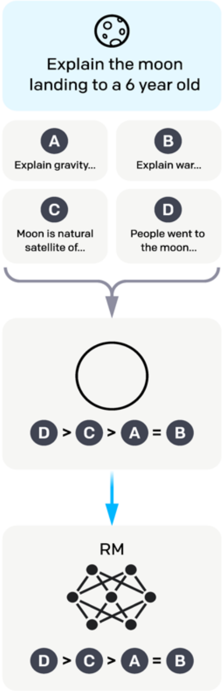
Step 2

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

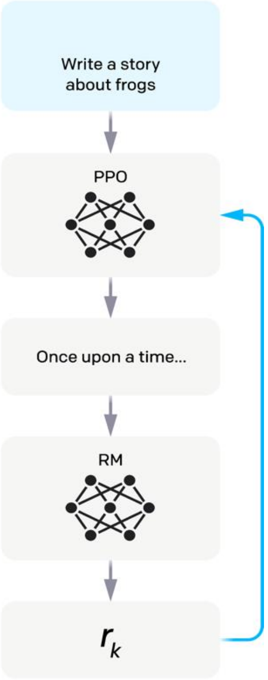
Optimize a policy against the reward model using reinforcement learning.

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.

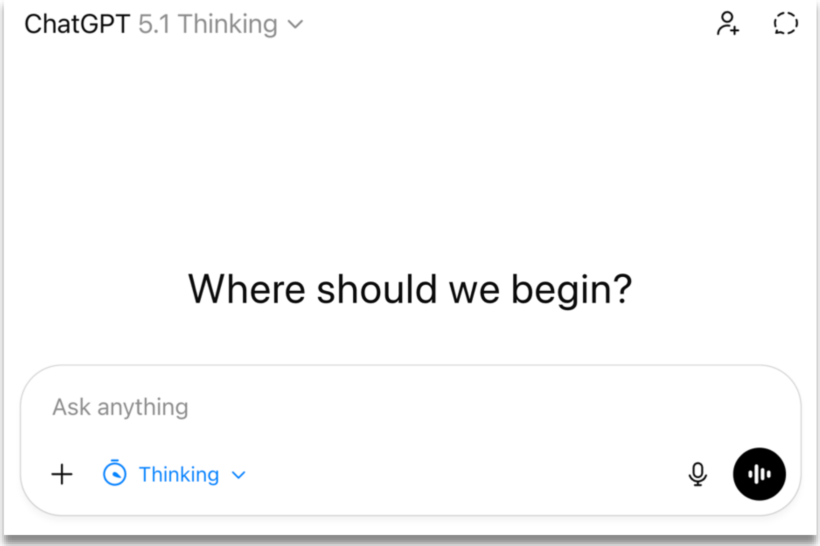


Language models as assistants?

How do we get from *this*

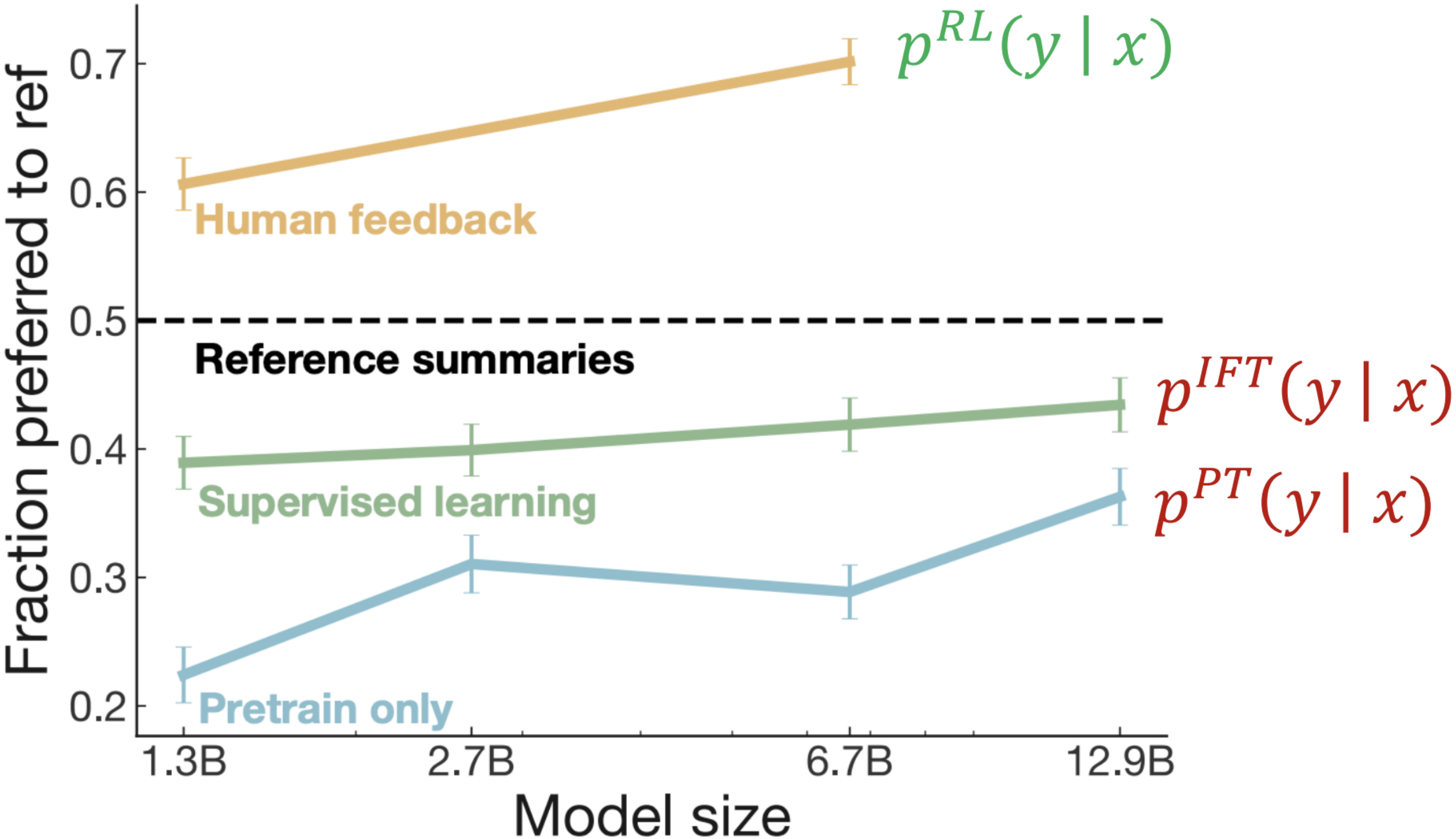
Stanford University is located in ____________

to *this*?

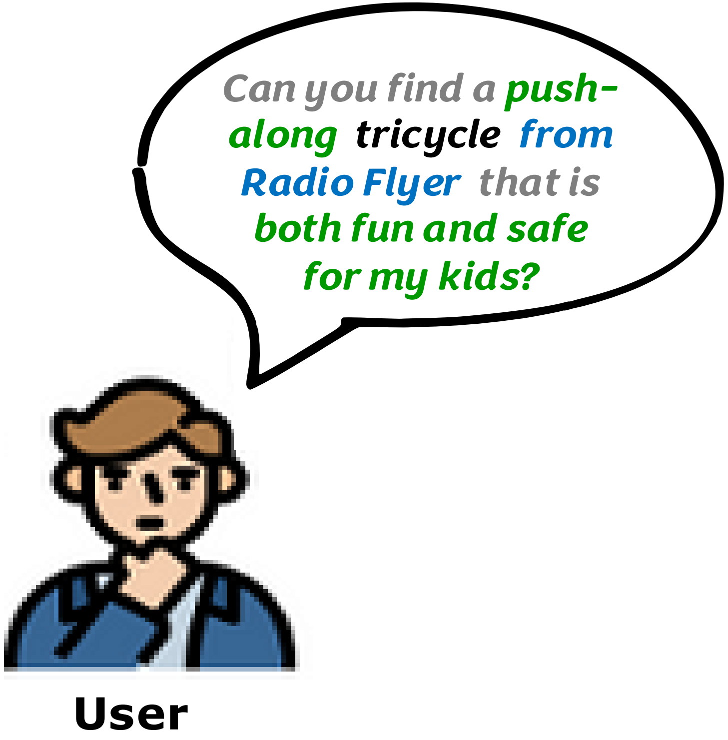


Shirley Wu (@ShirleyYXWu)

RLHF provides gains over pretraining + finetuning

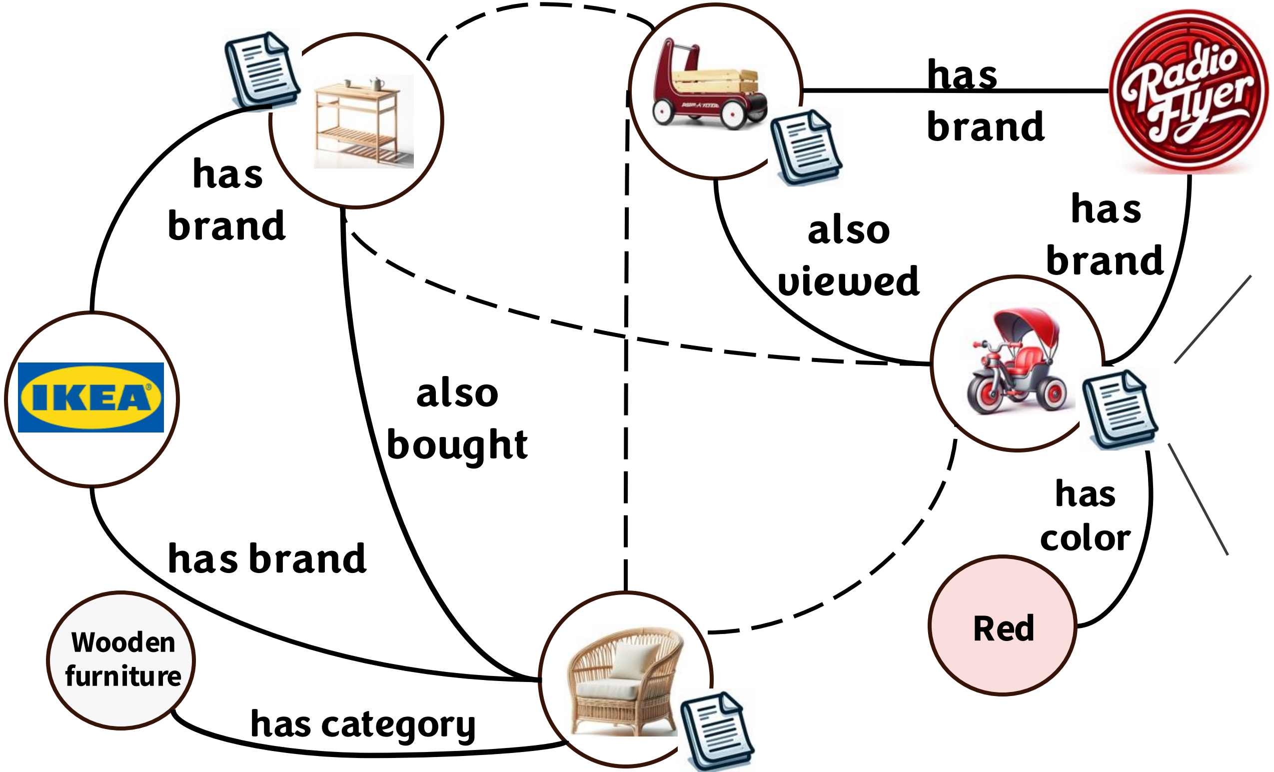


Queries are often Knowledge-Intensive



Navigate Complex Knowledge

—— Ecommerce: Amazon knowledge base

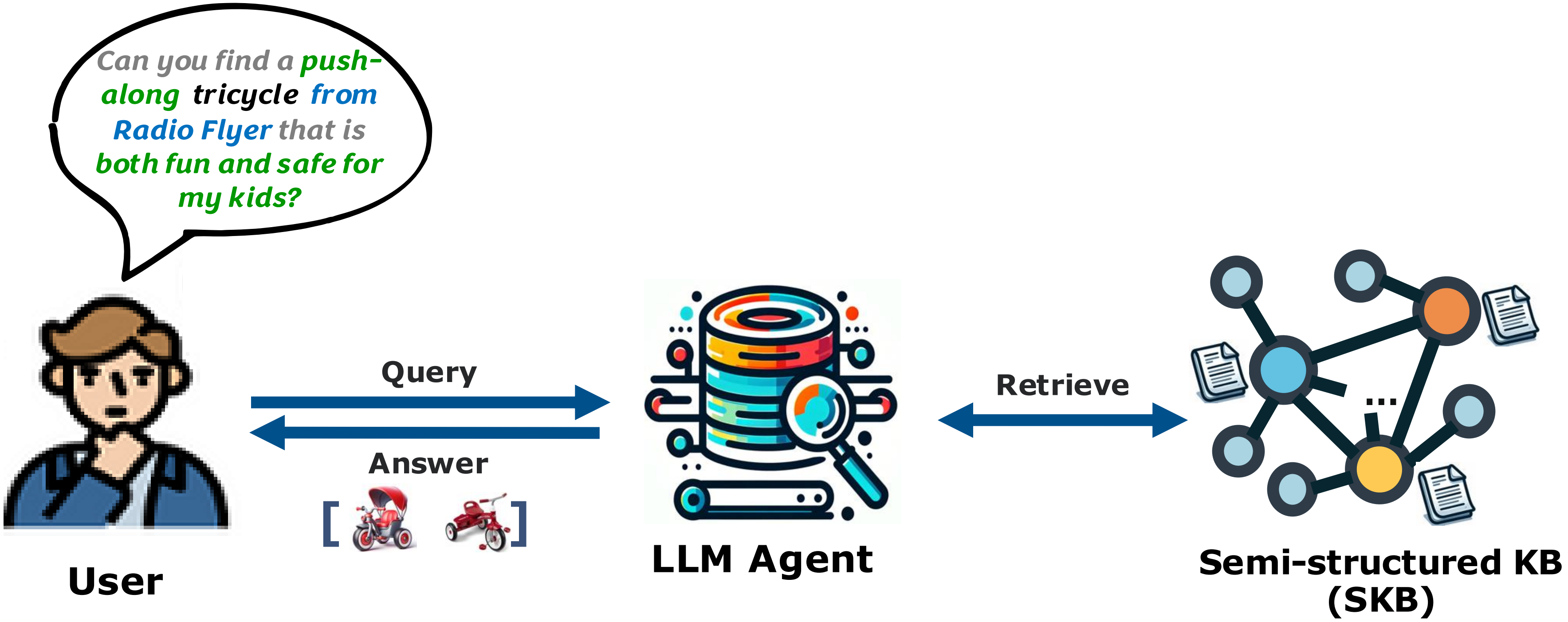


Title: Radio Flyer Ultimate All-Terrain Stroll 'N Tricycle
Price: $84.99
Feature:
- AGES 1 TO 5 YEARS: ..
- REMOVABLE ACCESSORIES: ..

Dimensions: 37.2”x34.3”x22”.

Description: This tricycle grows with your toddler through different riding stages.

Query Semi-structured Knowledge Bases



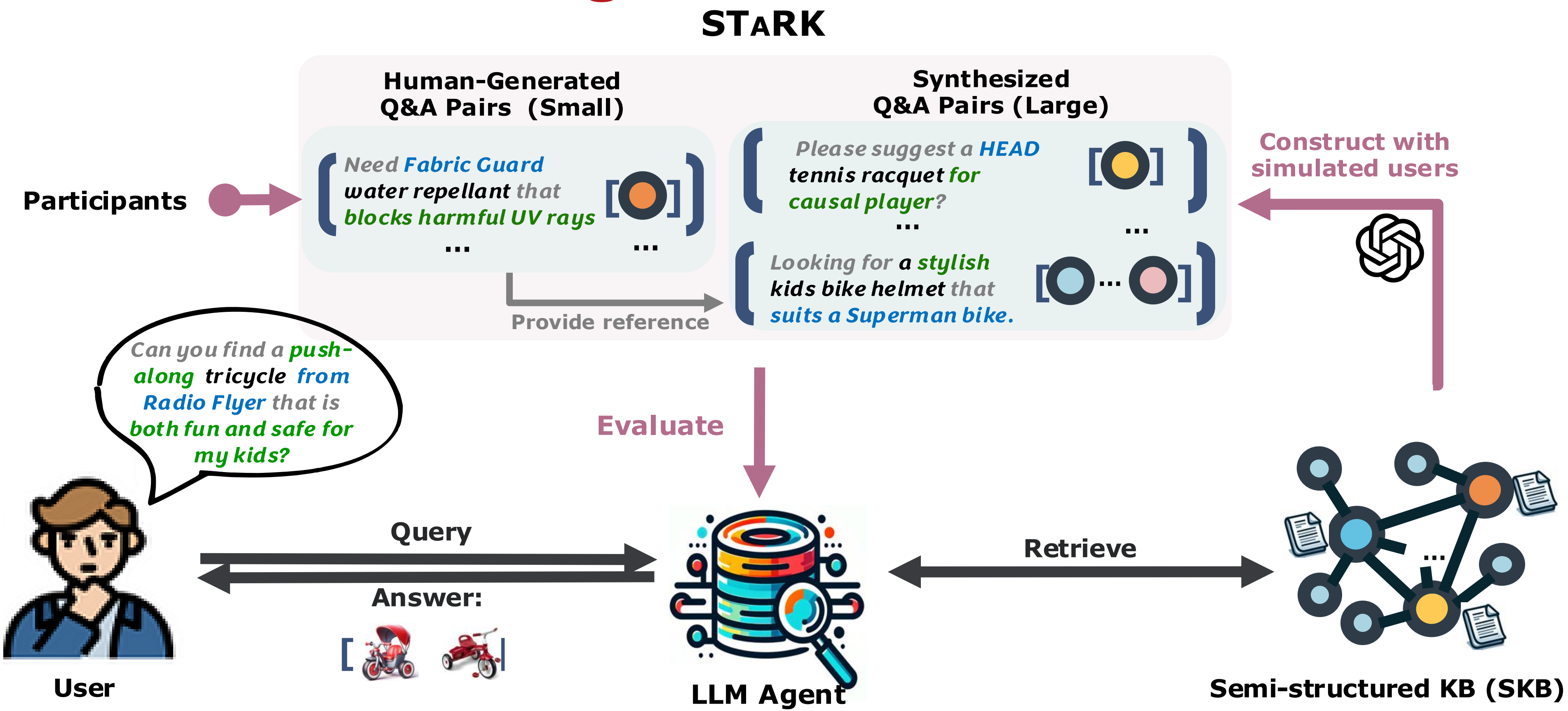
Why is it hard for LLMs?

Real-world queries require
multi-hop reasoning, filtering, and synthesis.

LLMs need to
① navigate large semi-structured knowledge bases,
② find useful information,
③ reason and aggregate answers.

Shirley Wu (@ShirleyYXWu)

Benchmarking semi-structure retrieval



Key Results: Retrieval-augmented methods and LLMs are not good

	STaRK-Prime	
	Hit@1	Hit@5
QAGNN (roberta)	7.14	17.14
ada-002	15.36	31.07
Claude3 Reranker	17.79	36.90

For all methods, Hit@1 is below 18%.

Takeaway: STaRK

STaRK establishes the challenge

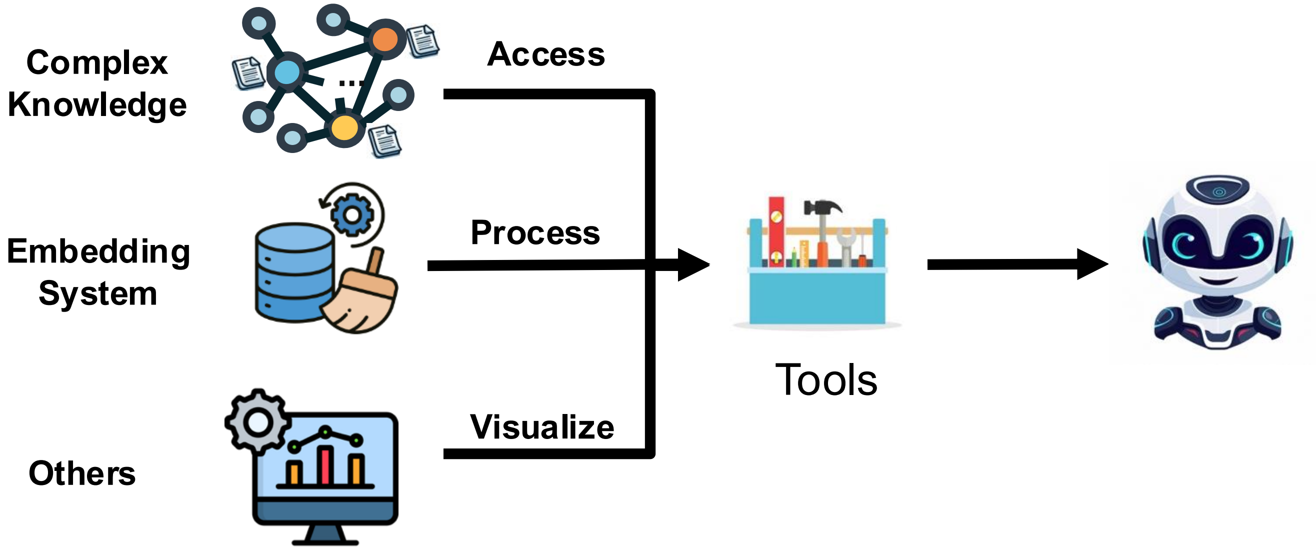
Current systems need significant improvement.

How do we make LLMs better at tackling these tasks?

Our insight

LLMs need to to effectively use external tools

(e.g. retrieval systems over KGs and text).



BUT LLMs use tools poorly

(a) ReAct: Unoptimized Agent

```
Input: Q1: "Can you suggest any TUSA swim fins that has a split fin design for better
propulsion?", Action space (GetEntityTypes, GetEntityDocuments, StringMatching, …)
Output: Answer(s)

Thought 1: Compute similarity scores based on the product description
Action 1: ComputeEmbeddingSimilarity["swim fins", GetEntityDocuments()]
Result/Obs 1: s1 ← similarity scores ❌ Ignore the brand information "TUSA"

Thought 2: Check the functionality requirement
Action 2: StringMatching["split fin design for better propulsion", GetEntityDocuments()]
Result/Obs 2: s2 ← string matching scores ❌ All zeros due to no perfect match

Thought 3: Synthesize the final result
Action 3: Multiply[s1, s2]
Result/Obs 3: s ← final scores ❌ All zeros which lead to a trivial solution
...
Final Result: answers ← GetTopkEntities[s, k=5] ❌ Poor task performance
```

LLM agents easily use wrong tools or use tools in a wrong way!

AvaTaR uses proper tools to solve the task!

(b) AVATAR: Optimized Agent

Input: Any query (demonstration example: Q1); Action space (GetEntityTypes, …)
Output: Answer(s)

✅ Accurately <u>decompose</u> the query into multiple aspects

Action 1: ParseAttributeFromQuery[query, (brand, type, material, features)]
Result 1: subquery ← { brand: "TUSA", type: "swim fins", material: NA, features: "split fin design for better propulsion" }

✅ Use <u>embedding tool</u> to filter entities

Action 2: ComputeEmbeddingSimilarity[subquery.type, **GetEntityTypes**()]
Result 2: s1 ← type similarity scores
Action 3: GetTopk[s1, k=20]
Result 3: candidates ← top-20 entities with the highest type similarity

✅ Use <u>token matching tool</u> for flexible brand matching

Action 4: GetEntityBrand[candidates]
Result 4: brands ← brands of the top-20 entities
Action 5: TokenMatching[subquery.brand, brands]
Result 5: s2 ← brand matching scores

✅ Use <u>LLM reasoning API</u> to validate the required functionality

Action 6: GetSatisfictionScoreByLLM[subquery.features, **GetEntityDocuments**()]
Result 6: s3 ← feature scores by LLM reasoning

...

✅ Synthesize final scores with optimized parameters

Action 7: WeightedSum[s1, s2, s3, coefficients=(0.43, 0.37, 0.20)]
Result 7: s ← combined scores
Final Result: answers ← **GetTopkEntities**[s, k=5] ✅ Excellent task performance

Our Idea in AVATAR

We need **more effective instructions** to improve the agent's ability in using tools!

We use **contrastive reasoning** to construct **better instructions.**

Contrastive Reasoning: Analogy

Think about teaching a student to do calculation:

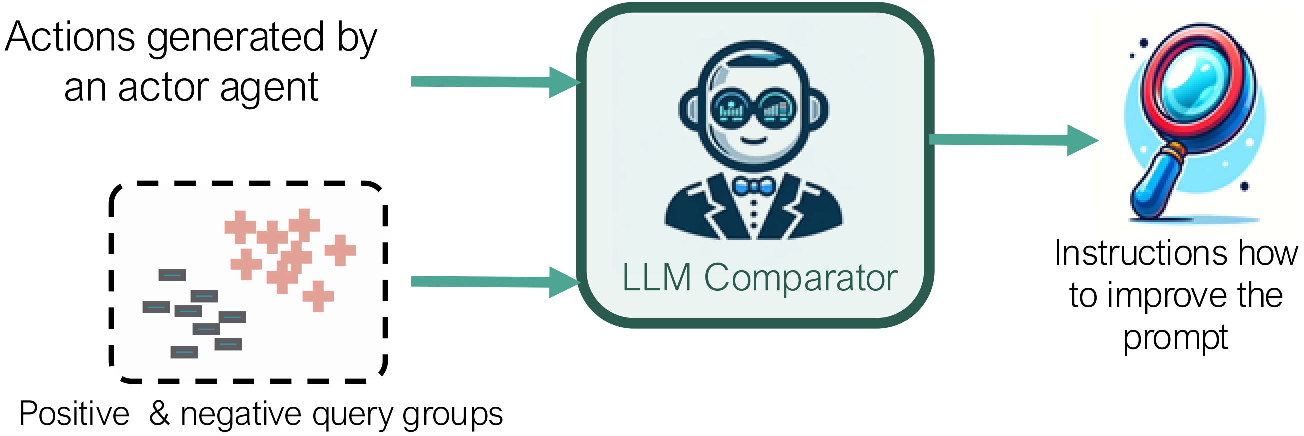
The problems that the student solves **correctly**:	The problems that the student solves **incorrectly**:
1 + 1 = 2 ✓	2 * 5 = 12 ✗
10 + 20 = 30 ✓	10 * 22 = 240 ✗
45 + 112 = 157 ✓	45 * 12 = 545 ✗
3 + (45 - 8) = 40 ✓	3 * (45 - 8) = 113 ✗

What does it tell us?

The student should practice multiplication!

We prompt an LLM Comparator to do contrastive reasoning!

The LLM Comparator gives insightful instructions by **understanding the gap** between positive and negative caused by the agent's actions!



Contrastive Reasoning by LLMs

We prompt the LLM Comparator:

"Here are two groups of queries that an agent perform poorly and well on, understand their differences:"

Queries answered correctly		Queries answered incorrectly	
"Need a pair of basketball NIKE shoes"	✓	"Find me visually stunning castle card modelling kits"	✗
"Recommend a scooter for under $100"	✓	"I want a nice mug for my cousin who is very into spiderman"	✗

LLM Comparator's output:

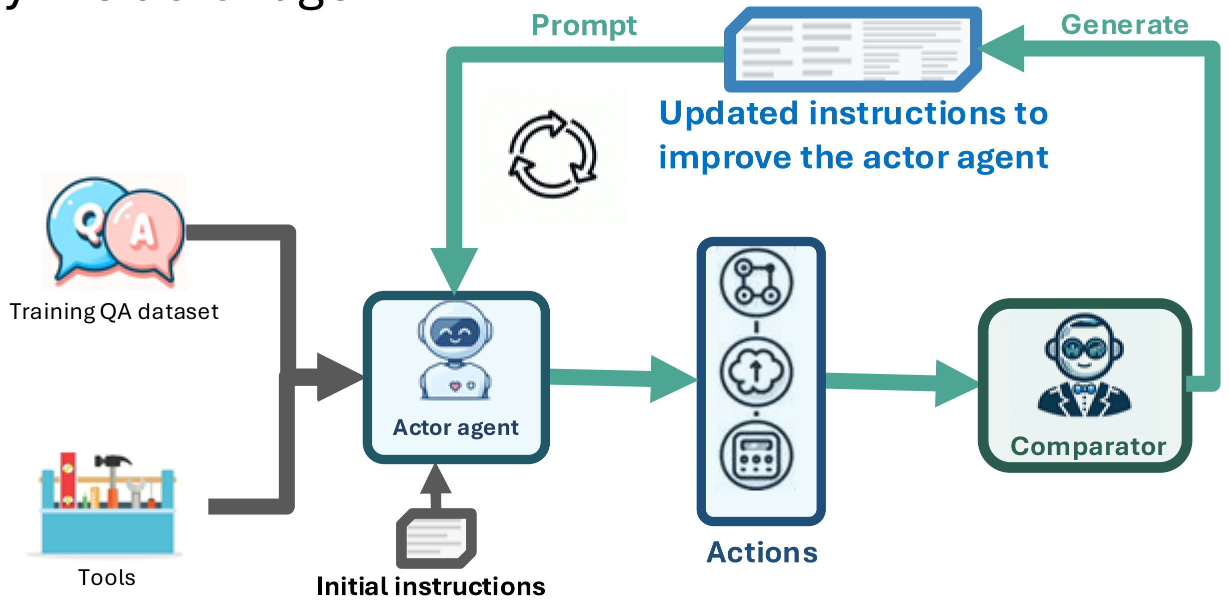
"You do well on queries with simple product features, while fail on specific and nuanced product descriptions.

I suggest to better parse and utilize query attributes. Use tools to compute F1 score for string matching."

Shirley Wu (@ShirleyYXWu)

AvaTaR: Contrastive Reasoning for Optimizing Tool Usage

Comparator's instruction improve actions generated by the actor agent



Results: AvaTaR on STaRK

Hit@1 Retrieval Score

	STaRK-Amazon	STaRK-MAG	STaRK-Prime
VSS (ada-002)	39.02	28.20	15.36
ReAct	42.14	31.07	15.28
Reflexion	42.79	40.71	14.28
AvaTaR	**49.87**	**44.36**	**18.44**
Relative Improvement	+28%	+57%	+20%

VSS: Vector similarity search (RAG)

ReAct (Yao et al. 2022): An unoptimized agent that generates actions for each query

Reflexion (Shinn et al. 2022): An agent optimized via self-reflection

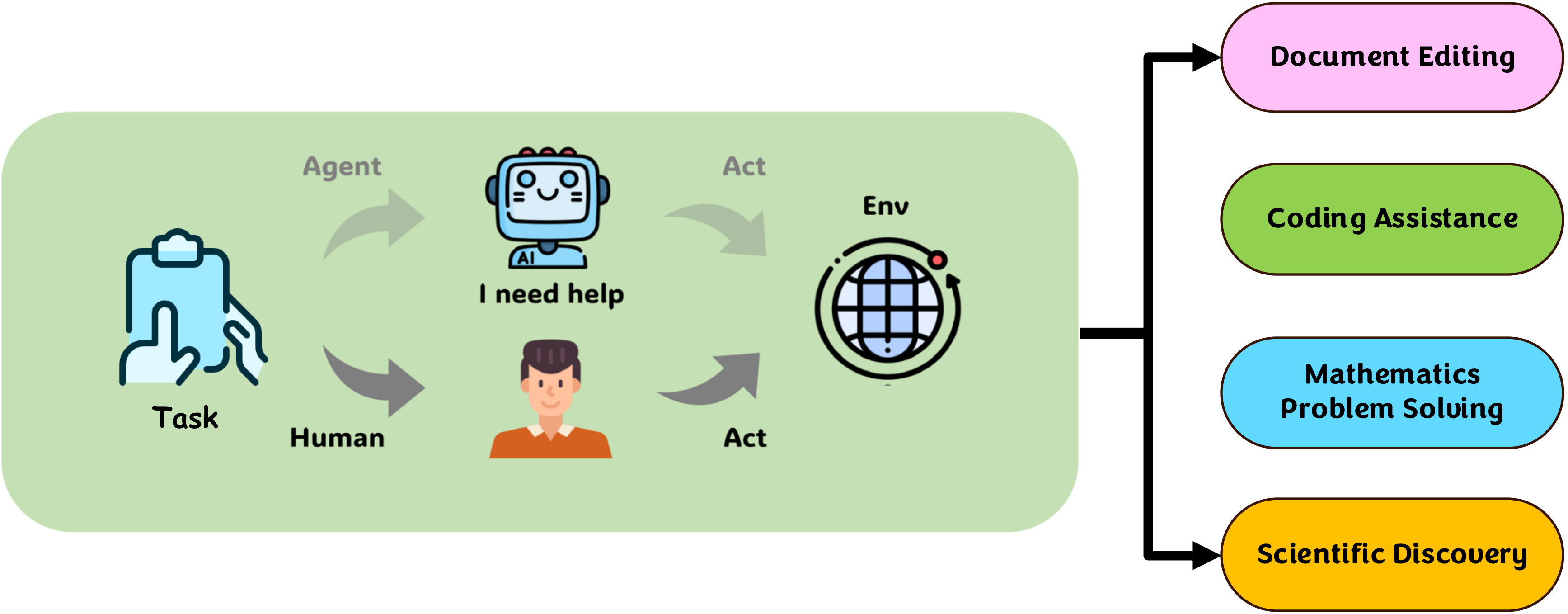
Takeaway: AvaTaR

AvaTaR helps LLMs better tackle complex Q&A tasks by improving their tool-use ability.

AvaTaR offers a principled, automated way to optimize LLM agents for tool use.

But complex tasks often involve <u>interaction</u> and <u>evolving goals</u>, not just one-shot Q&A.

Human-LLM interactions are everywhere



LLMs jump to (wrong) conclusions

Inefficient

What's a good pasta recipe?

Cook pasta, add chicken broth... [wasted tokens]

I am vegetarian 😠 !

Here is a vegetarian … [relevant tokens]

Useless

My muscles have been feeling really weak.

It could be
(1) Dehydration, ...
(6) Chronic conditions

I don't think these make sense 😕

Other reasons can be …

Examples from STaR-GATE (Andukuri et al. 2024), UnknowBench (Liu et al., 2024), STaRK (Wu et al., 2024)

Problems with LLMs

- LLMs don’t naturally help users clarify needs or explore options

- LLMs act as passive responders, especially when faced with ambiguity

Why do today's LLMs fail to actively understand users?

LLMs are usually tuned based on single-turn human preferences

I need to write an article about optimism

User query

Model response 1:

<article>

More useful in single turn
→ Higher reward

Model response 2:

<question>

No answer provided in single turn
→ Lower reward

Single-turn rewards encourage model responses that may NOT be useful in the long term.

Our work: CollabLLM

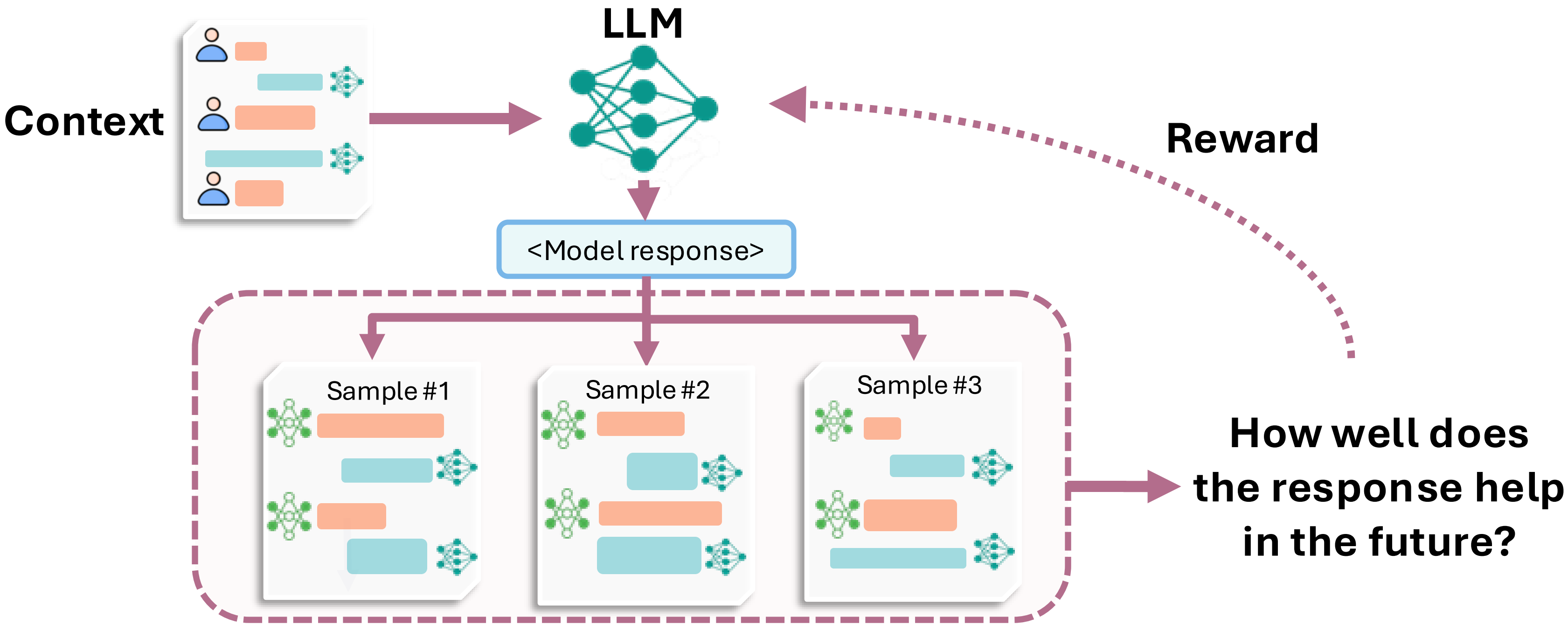
CollabLLM empowers LLMs to actively seek information from users and collaborate more effectively with humans!

Key Insight:

Rewards responses based on their long-term impact on the conversation.

→ **Multiturn-aware Reward**

Key question: How do we estimate a response's long-term impact?



But this requires sampling future trajectories

Our Idea: Using LLMs to simulate users

Inputs to User Simulator

Document Generation

Task description: "You should write a document"
Persona: User characteristics
Context: Current conversation

Synthetic future conversation

User Simulator

LLM

Can you help make it more concise?

...

Good start! Can we add more about ...?

...

Estimate long-term impact with synthetic conversations

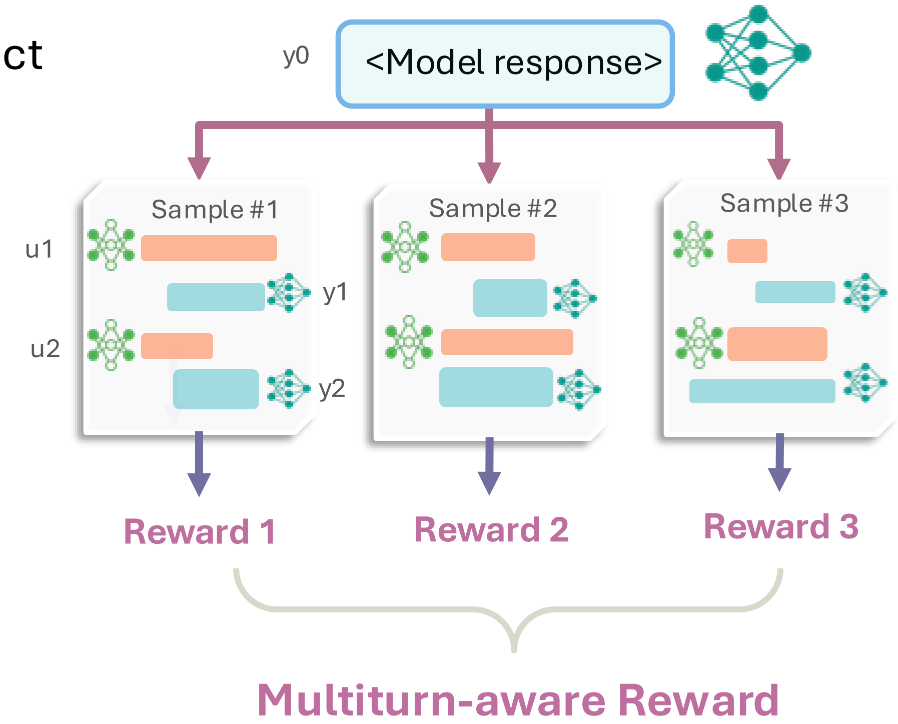
Goal: Evaluate the long-term impact of model response:

Approach:

① Sample synthetic conversations w/ User simulators

② Compute reward (e.g., accuracy, efficiency, interactivity) for each synthetic conversation

③ Average the rewards



Example 1: Estimate long-term impact

I need to write about how optimism can improve our well-being.

We want to estimate long-term impact for this response:

Here's a piece for you:
The Power of Optimism: Unlocking a Brighter You
……

Synthetic conversation example:

The tone is too formal and examples are too old-school.

The Optimism Revolution: Unleashing Your Inner Power
Hey there, friend! Are you ready to join the optimism revolution?
…….

Still not what I want, I want to talk about how it helps with relationships.

……

Efficiency: Low (user need to read 1.39k tokens) **Document quality: Low** **Interaction experience: Bad**

Example 2: Estimate long-term impact

I need to write about how optimism can improve our well-being.

We want to estimate long-term impact for this response:

To get us started, what kind of tone are you aiming for? Also, what do you want to highlight?

Synthetic conversation example:

I'm aiming for an inspirational tone. Touching on how it helps in relationships would be great!

I'd like to propose an opening paragraph to set the tone: <…>
Perhaps we can further add personal experience to make it more impactful.

Nice! Help me add some personal experience…

.....

Efficiency: High (only read 1.12k tokens) | Document quality: High | Interaction experience: Good

CollabLLM in a nutshell

Collaborative Simulation: Simulate multi-turn interactions.

Multiturn-aware Rewards: Causal Effect Estimation – how current response affects long-term conversation outcome

Reinforcement Finetuning: DPO/PPO finetune the LLM using these long-term, interaction-level rewards.

How do we evaluate models in multiturn environments?

BIGCODEBENCH

Instruct Prompt

Write a function `def task_func(script='backup.sh', log='/tmp/log.json')` to:
Description

The function should raise exception for:
Raises

The function should output with:
Returns

You should start with:

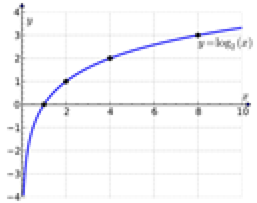
```
import os
import json

def task_func(
    script='backup.sh', log='/tmp/log.json'
):
```

HotpotQA

A Dataset for Diverse, Explainable Multi-hop Question Answering

Q: What was the former band of the member of Mother Love Bone who died just before the release of "Apple"?
A: Malfunkshun

MATHVISTA



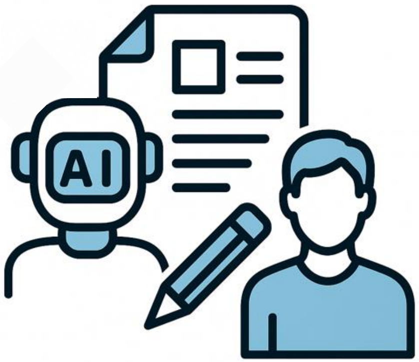
Question: The derivative of y at $x = 6$ is ____ that at $x = 8$.
Choices: (A) larger than (B) equal to (C) smaller than
Answer: (A) larger than

Question: How many zeros does this function have?
Answer: 1

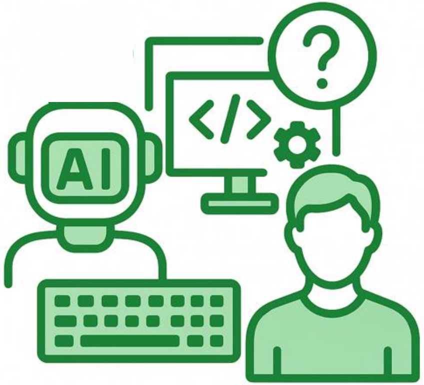
Question: What is the value of y at $x = 1$?
Answer: 0

Popular benchmarks are single-turn!

Our multiturn benchmarks



MediumDocEdit-Chat



BigCodeBench-Chat

Built on (Zhuo et al. 2024)

Math-Chat

Built on (Hendrycks et al. 2021)

Metrics:

Task Performance	**BLEU (doc. similarity)**	**Pass Rate (PR)**	**Accuracy**
User experience	**# Tokens:** Efficiency of LLM during the conversation **Interactivity (ITR):** How engaging the conversation is		

Methods

CollabLLM Trained on the benchmarks' training sets

Baselines:

"You should ask questions and reduce user efforts..."

Llama-3.1-8b-Instruct

Base

Llama-3.1-8b-Instruct

Proactive Base

Results on simulated environments

	BigCodeBench-Chat		
	PR ↑	#Tokens(k) ↓	ITR ↑
Base	9.3	1.59	22.0
Proactive Base	11.0	1.51	33.7
CollabLLM	13.0	1.31	52.0
Rel. Improv.	18.2%	13.2%	54.3%

CollabLLM obtains average improvements of 18%, 13%, 46% on task performance, efficiency, and interactivity, compared to Base and Proactive Base!

Results on simulated environments

	BigCodeBench-Chat			MediumDocEdit-Chat			MATH-Chat		
	PR ↑	#Tokens(k) ↓	ITR ↑	BLEU ↑	#Tokens(k) ↓	ITR ↑	ACC ↑	#Tokens(k) ↓	ITR ↑
Base	9.3	1.59	22.0	32.2	2.49	46.0	11.0	3.40	44.0
Proactive Base	11.0	1.51	33.7	35.0	2.18	62.0	12.5	2.90	46.0
CollabLLM	13.0	1.31	52.0	36.8	2.00	92.0	16.5	2.37	60.0
Rel. Improv.	18.2%	13.2%	54.3%	5.14%	8.25%	48.3%	32.0%	18.3%	36.4%

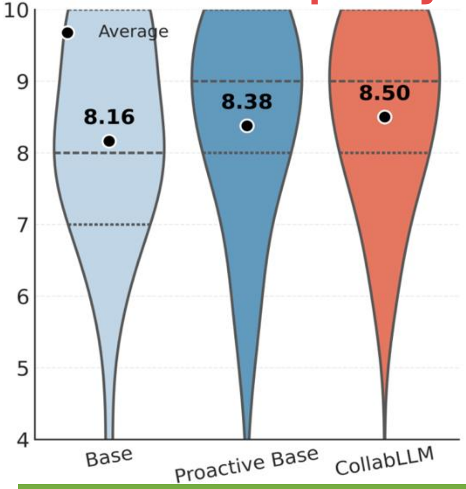
CollabLLM obtains average improvements of 18%, 13%, 46% on task performance, efficiency, and interactivity, compared to Base and Proactive Base!

Results on real-world environments

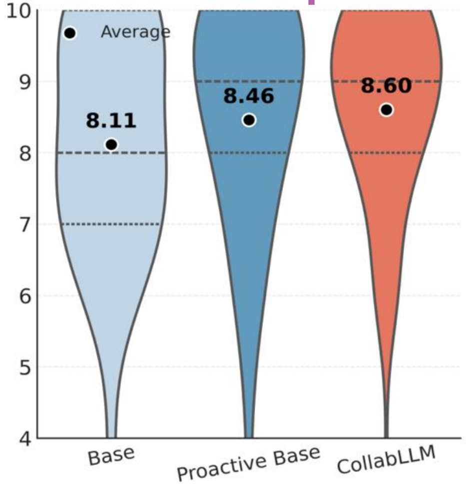
201 people were asked to complete writing tasks with LLMs:

- Give ratings (1-10) on the **document quality** and **interaction experience**.
- **Time spent** to finish the task is recorded

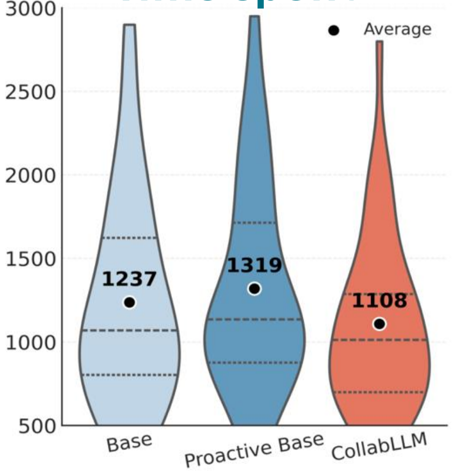
Document quality



Interaction experience



Time spent



CollabLLM yields high-quality documents, better user experience, and saves time by >10%!

CollabLLM improves collaboration

Representative feedback from participants:

About Base (Llama-3-1-8b):

"the AI just agreed with me on pretty much everything.
There was no debate or discussion."

About Proactive Base:

"The AI seemed to be very redundant and asked me the same questions over and over"

About CollabLLM:

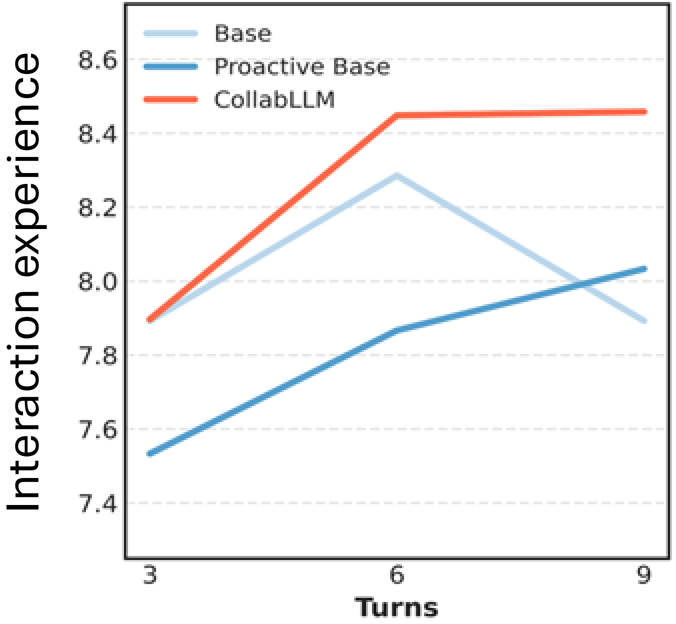
"It helped really well to navigate what to say and what information is needed"

"The AI really helped focusing on one part of the story at a time."

"Asking questions and making you think of things you never thought of"

CollabLLM improves user experience

Every 3 turns, we asked participants to rate their interaction experience (1-10).



Base model's performance degrades after multiple turns!

CollabLLM improves user experience along conversations.

CollabLLM generalizes

CollabLLM on Abg-CoQA benchmark:

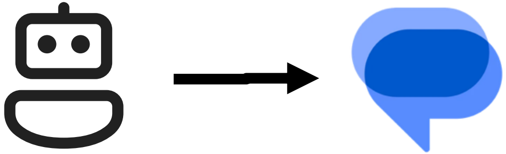
1) For ambiguous queries, model should ask questions

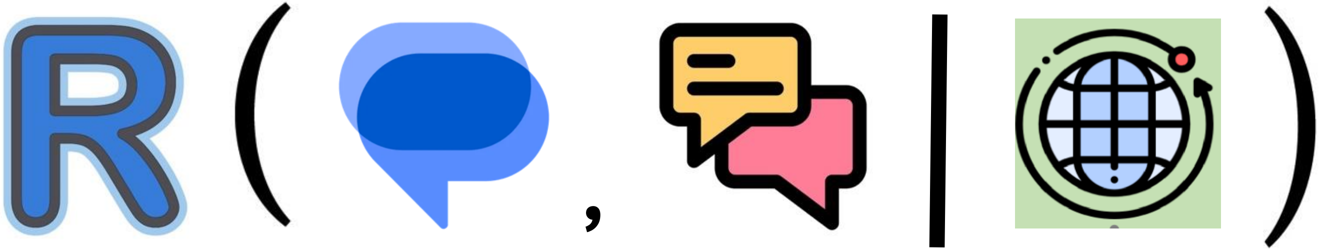
2) For unambiguous queries, model should provide direct answer

	Action-level Accuracy	
	Ambiguous	Non-Ambiguous
GPT-4o	15.44%	95.60%
Base (Llama-3.1-8B)	16.26%	90.40%
COLLABLLM	52.84%	72.32%

CollabLLM asks **~3x more questions when queries are ambiguous**. When queries are unambiguous, it only asks questions 18% more often than Base model.

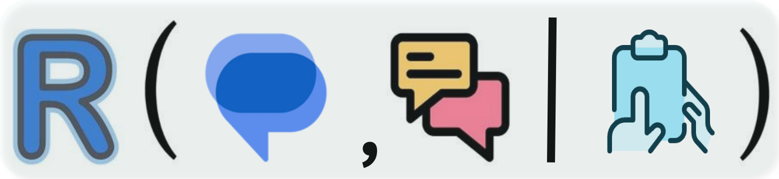
High-level takeaway





a_t s_t e

Task-centric objective



Task

Human-centric objective

Human

Building Agents that are Intelligent and Collaborative

STaRK (NeurIPS 2024) and **AvaTaR (NeurIPS 2024)** enable more intelligent AI agents that retrieve and use tools well.

Beyond that, CollabLLM (Outstanding Paper @ ICML 2025, 6 out of all papers) leads a new way to define what matters in human-AI collaboration.

STaRK

AvaTaR

CollabLLM

Stanford


Shirley Wu


Shiyu Zhao (OpenAI)


Qian Huang (MSL)


Michi Yasunaga (MSL)


Kaidi Cao (Anthropic)


Kexin Huang (Biomni)


James Zou


Jure Leskovec

Microsoft Research


Michel Galley


Yao Dou (GT)


Baolin Peng


Hao Cheng


Weixin Liang


Jianfeng Gao

Amazon


Vassilis Ioannidis


Karthik Subbian

Accenture


Cyril Weerasooriya


Wei Wei

END. Questions?