



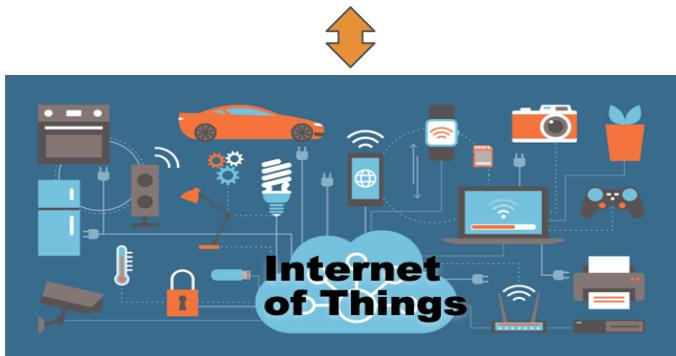
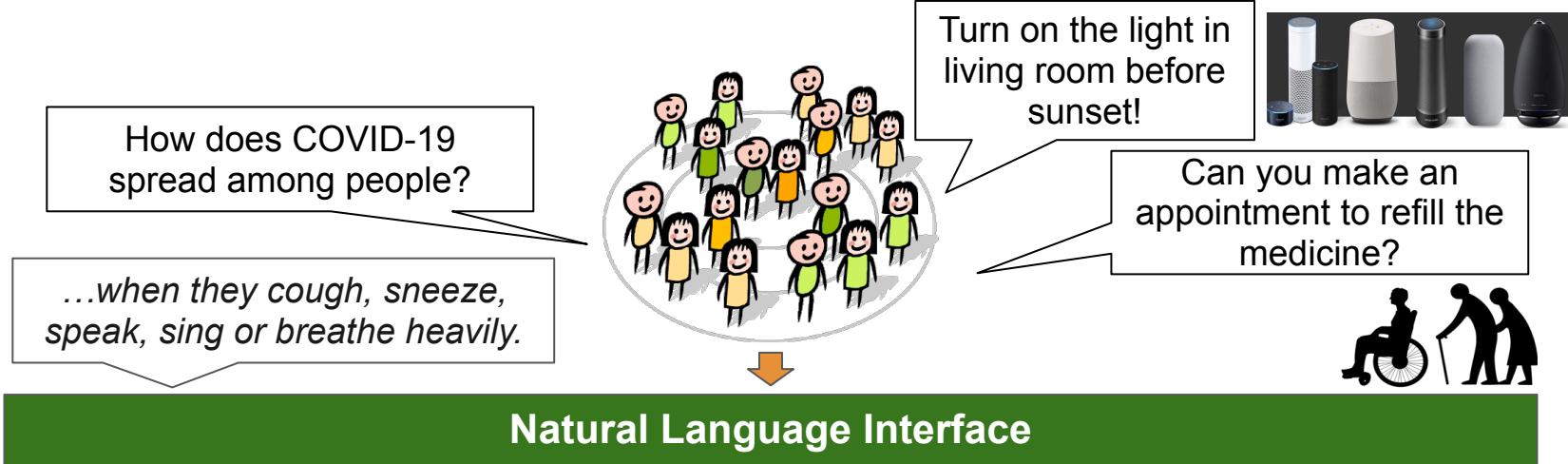
Towards Enhancing the Utilization of Large Language Models for Humans

Ziyu Yao

Assistant Professor, Computer Science
George Mason University
<https://ziyuyao.org/>

Virginia Tech (NVC Campus)
March 22, 2024

Natural Language Interfaces (NLIs)

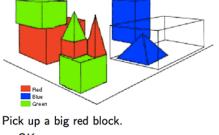


NLIs from Past to Present

- Early NLI systems



ELIZA (1966)



Person: Pick up a big red block.
Computer: OK.
Person: Grasp the pyramid.
Computer: I don't understand which pyramid you mean.

SHRDLU (1971)



Ask Jeeves (1997)

LLM Benchmark About Submit here!

Evaluation Queue for the 😊 Open LLM Leaderboard

These models will be automatically evaluated on the 😊 cluster.

- Finished Evaluations (5402)
- Running Evaluation Queue (13)
- Pending Evaluation Queue (1)

5000+ LLMs!
https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

- to learning-based systems
- then neural networks
- then pre-trained contextualized embedding (e.g., BERT)
- then large language models (LLMs)

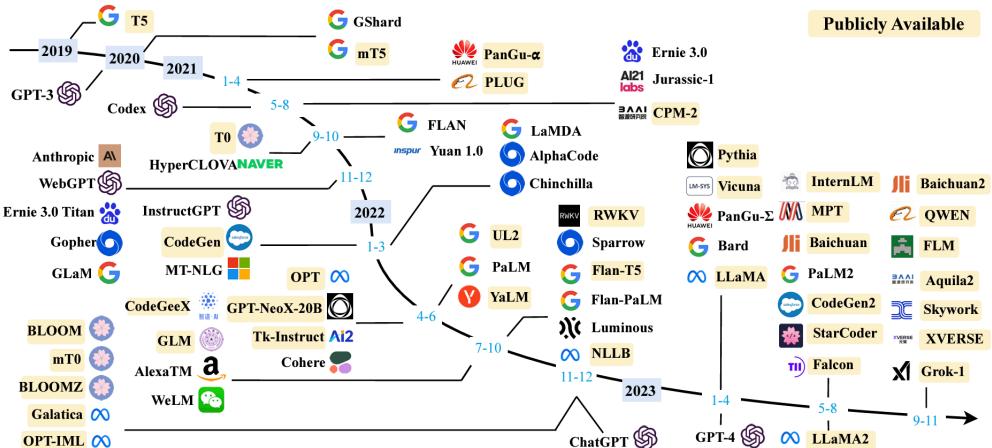


Image source: Zhao et al. "A survey of large language models." arXiv preprint arXiv:2303.18223.

NLIs in the Age of Large Language Models

Paradigm shift: unified architecture, task generalization, instruction following

Semantic Parsing

Which players did win the Australian Open?

Question Answering

Greece held its last Summer Olympics in which year?

Data-to-Text Generation

Describe the table result.

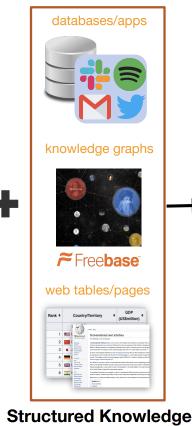
Fact Verification

Canada obtained 3 more gold medals than Mexico.

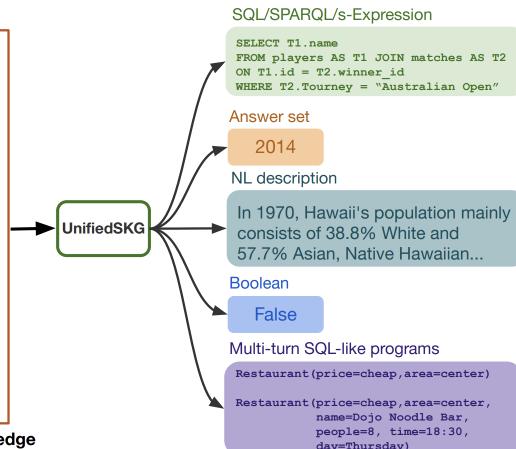
Dialogs

I am looking for a cheap restaurant in the city center.

Book a table for 8 at 18:30 on Thursday.



Structured Knowledge



UnifiedSKG (Xie...Yao et al., 2022)



LLM (e.g., ChatGPT) as a unified NLI

- + Plugins and tools
- + Program/symbolic representation (e.g., Amazon's Dialog2API)

NLIs in the Age of Large Language Models

...and the Challenges

Challenge 1: The expensive use of LLMs through API calls

Model	Input	Output
gpt-4	\$30.00 / 1M tokens	\$60.00 / 1M tokens
gpt-4-32k	\$60.00 / 1M tokens	\$120.00 / 1M tokens

GPT-4

<https://openai.com/pricing>



(Drawn with
DALLE)

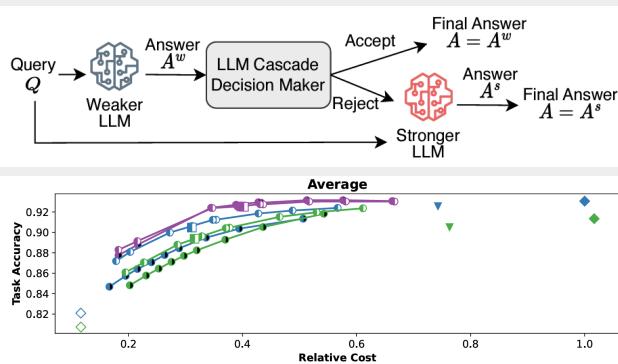
Challenge 2: The limited accessibility of LLMs from non-LLM-tech-savvy people



Teachers designing homework assignments, physicians querying medical knowledge, policy makers confirming regulation details, etc.

This Talk: Towards Enhancing the Utilization of LLMs for Humans

Topic 1: LLM Cascade with Mixture-of-Thought for Cost-Efficient Reasoning



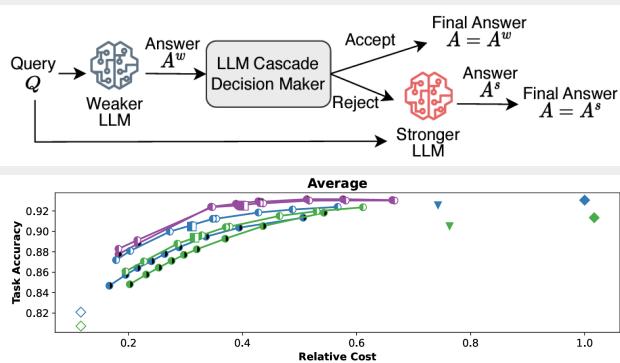
Topic 2: Instance-level Prompt Optimization with LLMs in the Loop



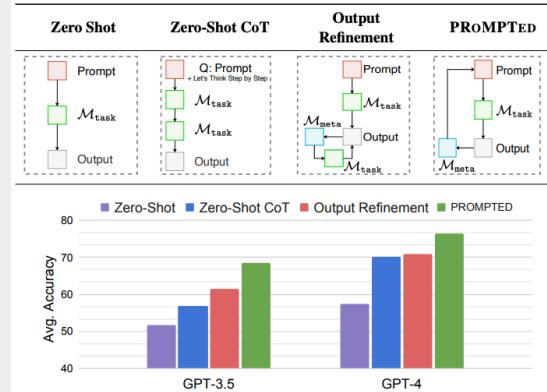
Other ongoing effort: LLM interpretability + Alignment, LLMs for education (supported by Microsoft AFMR), etc.

This Talk: Towards Enhancing the Utilization of LLMs for Humans

Topic 1: LLM Cascade with Mixture-of-Thought for Cost-Efficient Reasoning



Topic 2: Instance-level Prompt Optimization with LLMs in the Loop



Other ongoing effort: LLM interpretability + Alignment, LLMs for education (supported by Microsoft AFMR), etc.

The Trade-Off between LLMs' Cost(\$) and Performance

- More powerful, but also more expensive, LLMs
 - E.g., GPT-4 vs. GPT-3.5-turbo

Model	Input	Output
8K context	\$0.03 / 1K tokens	\$0.06 / 1K tokens
32K context	\$0.06 / 1K tokens	\$0.12 / 1K tokens

Model	Input	Output
4K context	\$0.0015 / 1K tokens	\$0.002 / 1K tokens
16K context	\$0.003 / 1K tokens	\$0.004 / 1K tokens

<https://openai.com/pricing>

GPT-4
GPT-3.5-turbo

20x \$ for input
30x \$ for output
More powerful
but less
affordable!



How can we save \$ without sacrificing task performance?

LLM Cascades with Mixture of Thought Representations for Cost-Efficient Reasoning



Murong Yue 



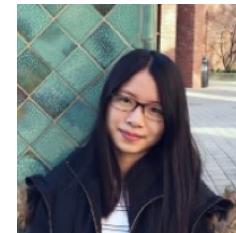
Jie Zhao 



Min Zhang 



Liang Du 



Ziyu Yao 

ICLR 2024

LLM Cascades for Cost Saving

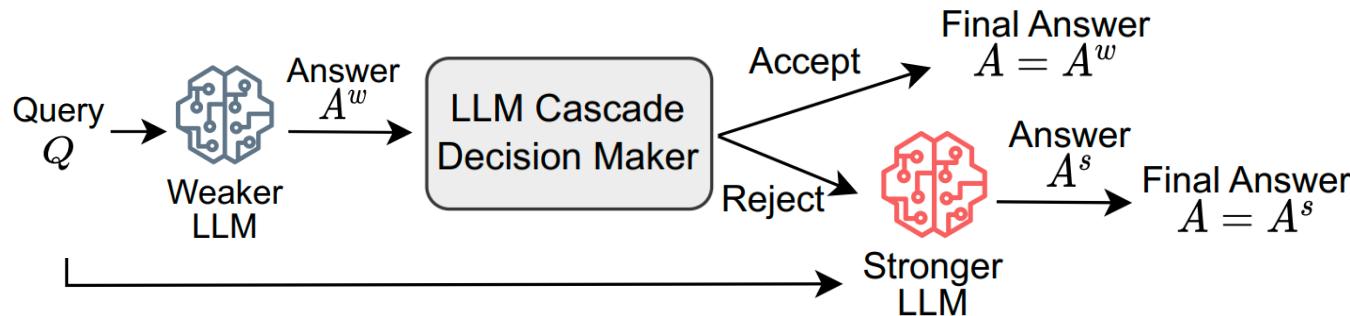
Intuition: easy questions can be handled by relatively weaker (and cheaper) LLMs to save \$.

FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance

Lingjiao Chen, Matei Zaharia, James Zou

Stanford University

*Decision making based on textual descriptions of question and answer;
Do not work for Reasoning*



$$\text{Final cost: } C = C^w + C^d + \mathbf{1}_{\text{reject}} C^s$$

Final performance: based on Final Answer A

Extreme cases: only weaker LLM or only stronger LLM

Reasoning with Thought Representations

Chain of Thought (**CoT**; Wei et al., 2022) & Program of Thought (**PoT**; Chen et al., 2022, Gao et al., 2022)

GSM8k (Cobbe et al., 2021)

Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?

A (CoT): It takes $2/2=1$ bolt of white fiber. So the total amount of fabric is $2+1=3$ bolts of fabric.
ans=3

A (PoT):
Python code, return ans
bolts_of_blue_fiber = 2
bolts_of_white_fiber = num_of_blue_fiber / 2
ans = bolts_of_blue_fiber + bolts_of_white_fiber

DATE (BIG-Bench Collaboration, 2021)

Q: Today is Christmas Eve of 1937. What is the date tomorrow in MM/DD/YYYY?

(CoT) Explain: Today is the Christmas Eve of 1937, so today is 12/24/1937.

Today is 12/24/1937, the date tomorrow is 12/25/1937.

A: 12/25/1937

(PoT) # Write Python Code to solve the following questions.
from datetime import date, timedelta
from dateutil.relativedelta import relativedelta

Q: Today is Christmas Eve of 1937. What is the date tomorrow in MM/DD/YYYY?

today is Christmas Eve of 1937, then today is 12/24/1937
today = date(1937, 12, 24)

tomorrow

date_tomorrow = today + relativedelta(days=1)

The answer formatted with %m/%d/%Y is
ans = date_tomorrow.strftime("%m/%d/%Y")

LLM Cascade Decision-Making: When to Accept the Weaker LLM?

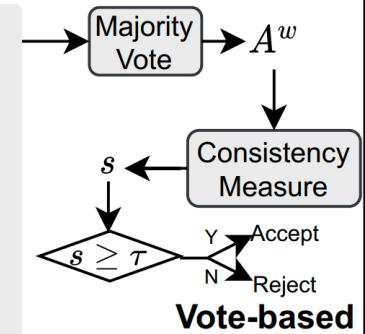
- Idea: if the weaker LLM is uncertain about an answer, the question could be too challenging for it to solve
- How to measure an LLM's (un)certainty on an answer?
 - See how consistently it samples the same answer
 - Same idea as “Self Consistency (SC)” (Wang et al., 2023)
- Questions:
 - Where to sample the answers for better judgment?
 - How to quantify the answer consistency?

**Our Work: Diversified Answer Sampling +
Voting/Verification Consistency Measurement**

Approaches

- Voting-based decision making

Sampled K Answers



$$s = \frac{\sum_{i=1}^K \mathbb{1}_{A_i^w = A^w}}{K}$$

Approaches

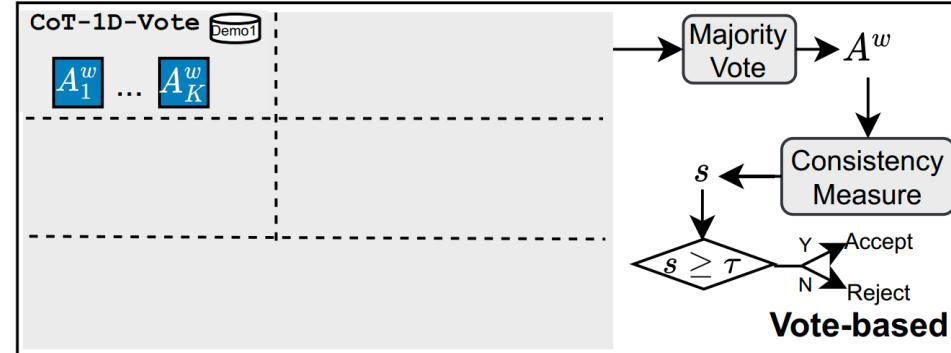
- Voting-based decision making, sampling from
 - a single thought representation
 - a single demonstration set

Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?

A: It takes $2/2=1$ bolt of white fiber. So the total amount of fabric is $2+1=3$ bolts of fabric. ans=3

... (M shots of CoT examples)

Q: Test question
A:



$$s = \frac{\sum_{i=1}^K \mathbb{1}_{A_i^w = A^w}}{K}$$



Method: CoT-1D-Vote

Approaches

- Voting-based decision making, sampling from
 - a single thought representation
 - a single demonstration set

Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?

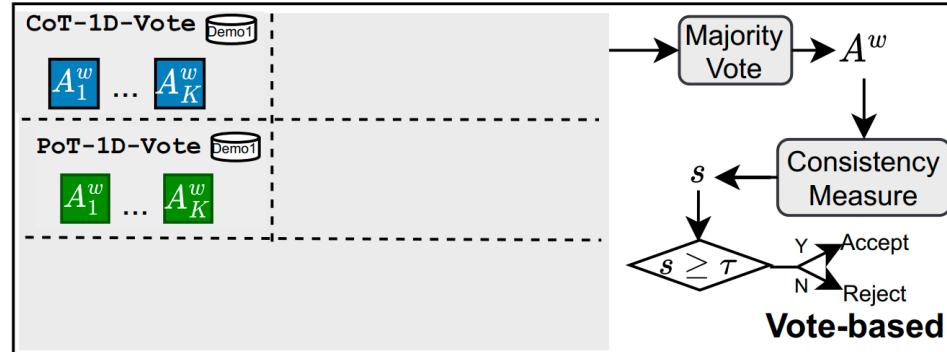
A:

```
# Python code, return ans
bolts_of_blue_fiber = 2
bolts_of_white_fiber = num_of_blue_fiber / 2
ans = bolts_of_blue_fiber + bolts_of_white_fiber
```

... (M shots of PoT examples)

Q: Test question

A:



$$s = \frac{\sum_{i=1}^K \mathbb{1}_{A_i^w = A^w}}{K}$$



Method: PoT-1D-Vote

Approaches

- Voting-based decision making, sampling from
 - a single thought representation
 - Two demonstration sets**



Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?
 A: It takes $2/2=1$ bolt of white fiber. So the total amount of fabric is $2+1=3$ bolts of fabric. ans=3

... (M shots of CoT examples from Set 1)

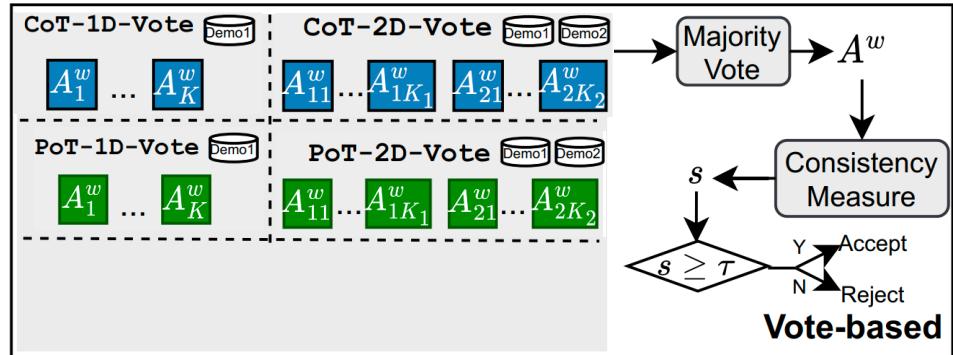
Q: Test question
 A:



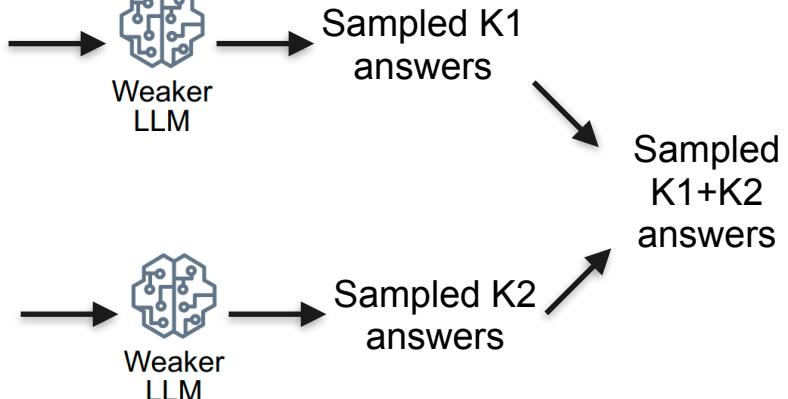
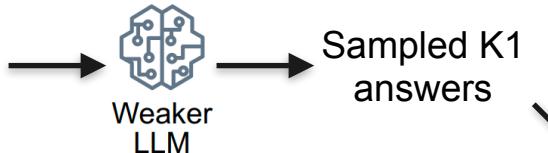
Q: Manny had 3 birthday cookie pies to share with his 24 classmates and his teacher, Mr. Keith. ...
 A: There is a total of $3 \times 10 = 30$ cookie slices... ans = 4

... (M shots of CoT examples from Set 2)

Q: Test question
 A:



$$s = \frac{\sum_{i=1}^{K_1} \mathbb{1}_{A_{1i}^w = A^w} + \sum_{i=1}^{K_2} \mathbb{1}_{A_{2i}^w = A^w}}{K_1 + K_2}$$



Method: CoT-2D-Vote
 (Similarly for PoT-2D-Vote)

Approaches

- Voting-based decision making, sampling from
 - Two thought representations**
 - a single demonstration set



Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?
 A: It takes $2/2=1$ bolt of white fiber. So the total amount of fabric is $2+1=3$ bolts of fabric. ans=3

... (M shots of CoT examples from Set 1)

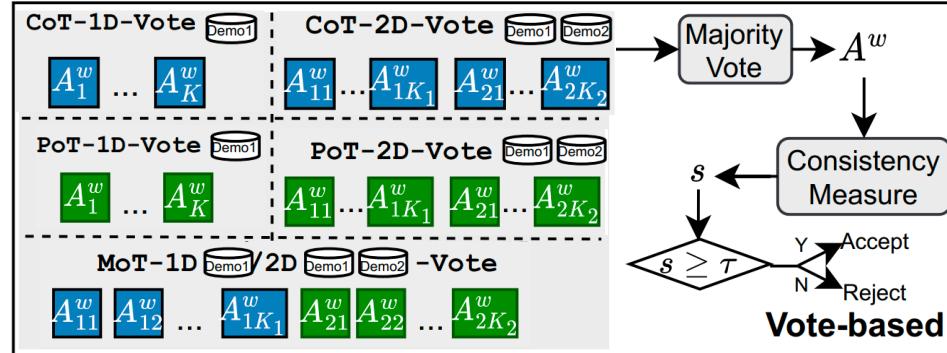
Q: Test question
 A:



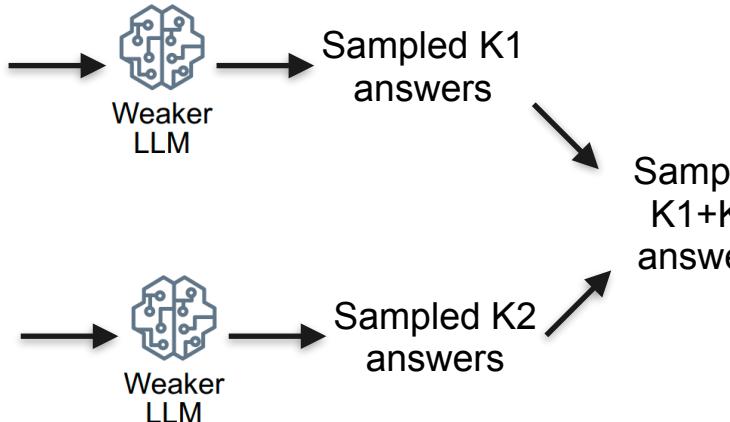
Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?
 A:
 # Python code, return ans
`...ans = bolts_of_blue_fiber + bolts_of_white_fiber`

... (M shots of PoT examples from Set 1)

Q: Test question
 A:



$$s = \frac{\sum_{i=1}^{K_1} \mathbb{1}_{A_{1i}^w = A^w} + \sum_{i=1}^{K_2} \mathbb{1}_{A_{2i}^w = A^w}}{K_1 + K_2}$$



Method: **MoT-1D-Vote**
"Mixture of Thought"

Approaches

- Voting-based decision making, sampling from
 - Two thought representations
 - Two demonstration sets



Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?
 A: It takes $2/2=1$ bolt of white fiber. So the total amount of fabric is $2+1=3$ bolts of fabric. ans=3

... (M shots of CoT examples from Set 1)

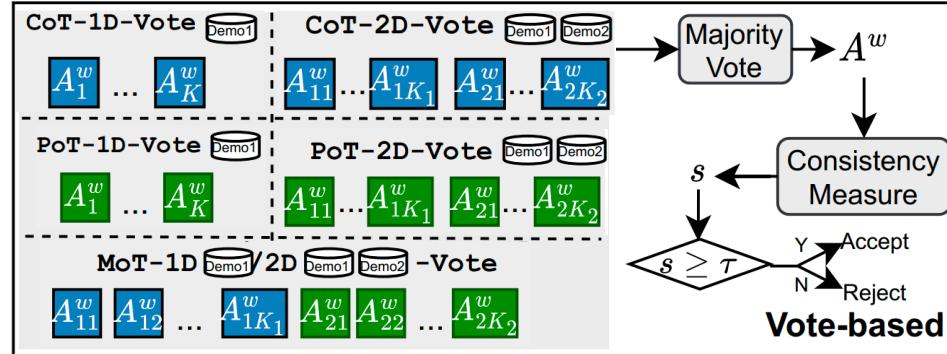
Q: Test question
 A:



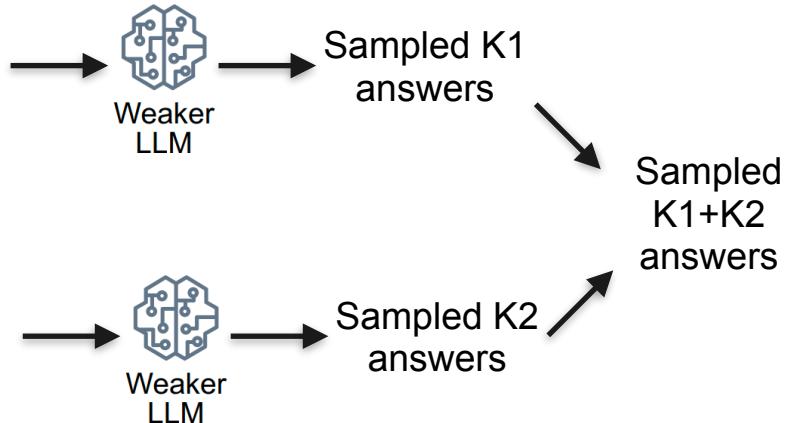
Q: Manny had 3 birthday cookie pies to share with his 24 classmates and his teacher, Mr. Keith. ...
 A:
 # Python code, return ans
`#ans = total_cookie_pies - total_person_count`

... (M shots of PoT examples from Set 2)

Q: Test question
 A:



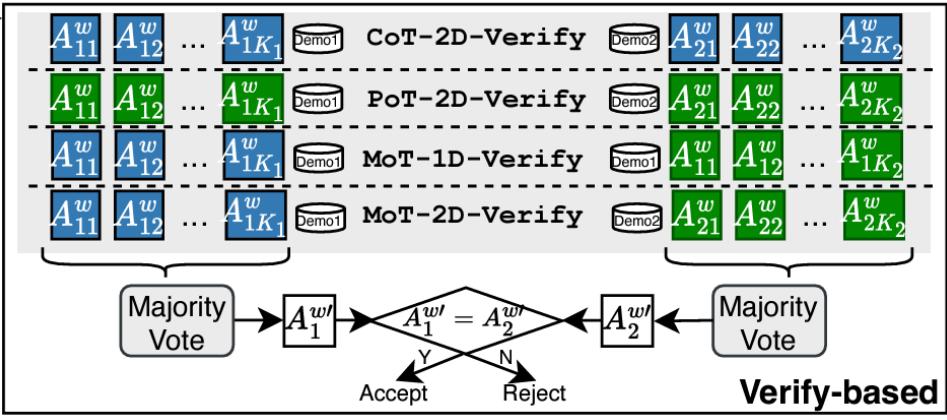
$$s = \frac{\sum_{i=1}^{K_1} \mathbb{1}_{A_{1i}^w = A^w} + \sum_{i=1}^{K_2} \mathbb{1}_{A_{2i}^w = A^w}}{K_1 + K_2}$$



Method: MoT-2D-Vote

Approaches

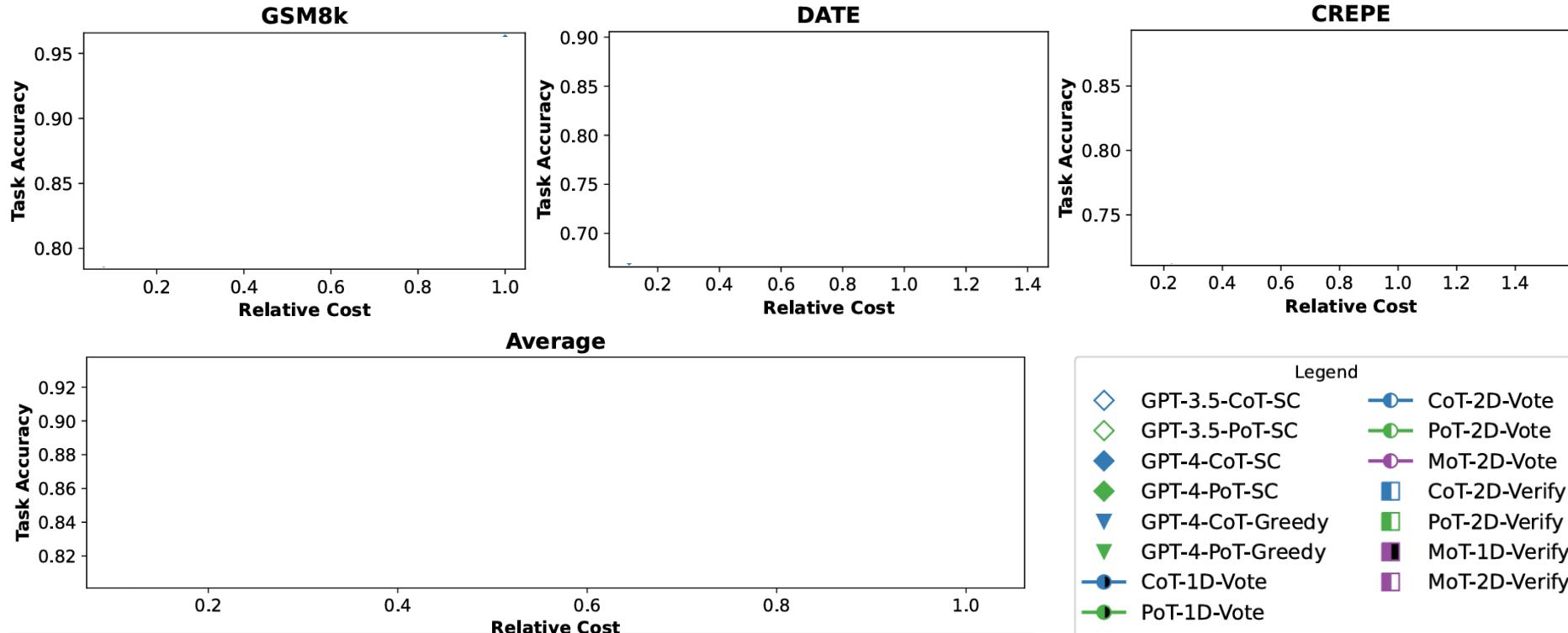
- Verification-based decision making
- Variants depending on
 - The selected thought representations
 - Num of demonstration sets
- Comparison of two kinds of measure
 - **Vote-based:** threshold tuning to meet budget constraint
 - **Verification-based:** relatively optimal solution without threshold engineering



$$s = \mathbb{1}_{A_1^{w'} = A_2^{w'}}$$

Experimental Results

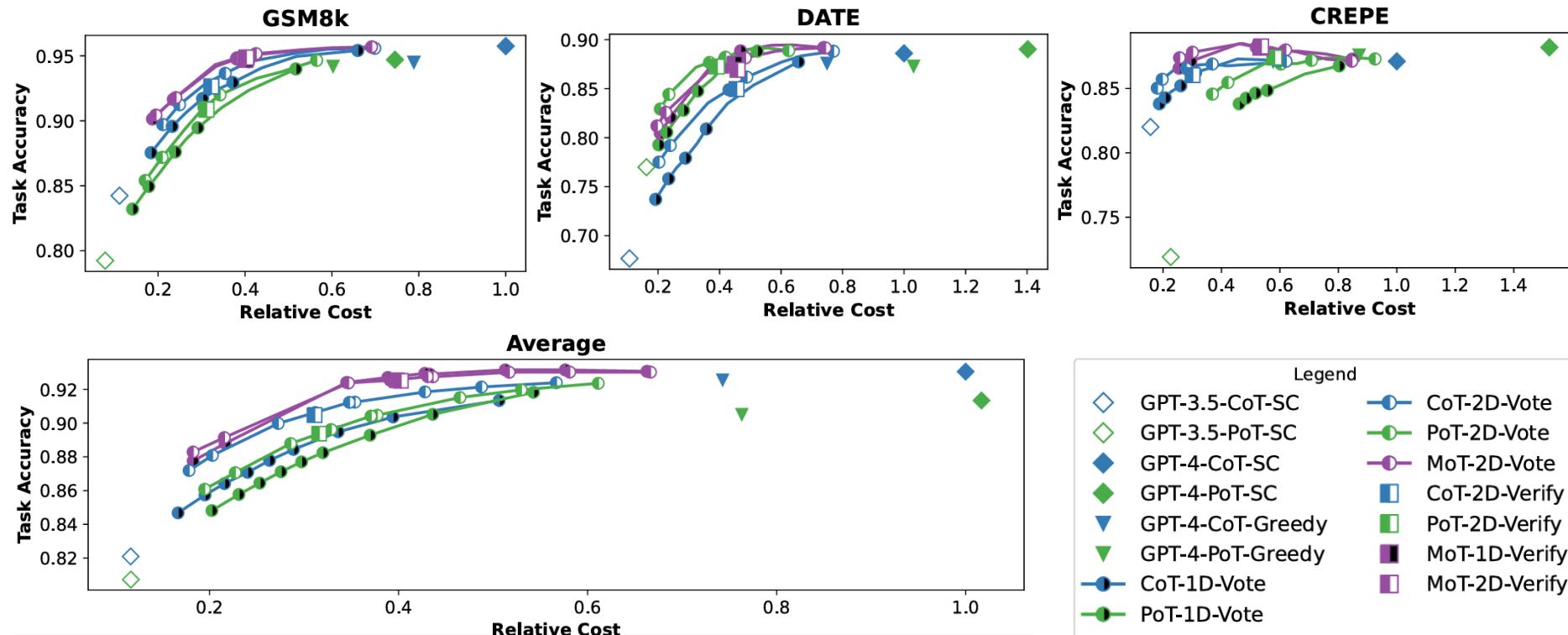
Weaker LLM: GPT-3.5-turbo
Stronger LLM: GPT-4



(Average over GSM8k, ASDIV, TabMWP, DATE, Navigate, CREPE)

Experimental Results

Weaker LLM: GPT-3.5-turbo
Stronger LLM: GPT-4



(Average over GSM8k, ASDIV, TabMWP, DATE, Navigate, CREPE)

Diversify Thought Representations for Uncertainty Measurement

- Mixture-of-Thought (MoT) often yields a larger certainty gap between easy/correct vs. hard/incorrect questions

Question: Blake and Kelly are having a contest to see who can run the most in 15 minutes. They decide to do it on a football field that is 100 yards long. Blake runs back and forth 15 times. Kelly runs back and forth once, and then decides that she doesn't want to run next to Blake, so she starts to run to the 40-yard line and back. She does this 34 times. How much farther does the winner run than the loser?

Gold

Black runs 200 yards for each back and forth because...He runs 3000 yards in total because...Kelly runs 200 yards at first because $100 \times 2 = 200$. She runs 80 yards each time she does a back and forth to the 40-yard line because... She runs 2720 from these back and forth. She runs 2920 in total because...The winner ran 80 yards more because $3000 - 2920 = 80$.

CoT 1

Answer: [...(Ignored)] Kelly runs to the 40-yard line and back 34 times, so she runs $34 \times 80 = 2720$ yards. Thus, Blake runs $3000 - 2720 = 280$ yards farther than Kelly.

Cot 2

Answer: [...(Ignored)] Kelly runs $34 \times (40 \times 2) = 2720$ yards. Blake runs $3000 - 2720 = 280$ yards farther than Kelly.

PoT 1

```
# Python code, return ans  
[ ... (Ignored) ]  
distance_covered_by_kelly =  
(length_of_football_field *  
num_of_laps_for_kelly * 2) + (40 *  
num_of_laps_for_kelly * 2)  
ans = abs(distance_covered_by_blake  
- distance_covered_by_kelly)  
(Answer via Python execution: 6520.0)
```

Logic Generation Error

Value Grounding Error

Takeaway: MoT introduces more diverse “opinions” than mixing demonstration sets → help uncertainty measurement

Diversify Thought Representations for Uncertainty Measurement

- Mixture-of-Thought (MoT) often yields a larger certainty gap between easy/correct vs. hard/incorrect questions

Question	Golden
Is a curling iron necessary in curling?	A curling iron is a tool used to make the hair curly using heat. The sport of curling requires curling brooms, stones (rocks), and curling shoes. A curling iron is not necessary in curling.
CoT	PoT
Yes, a curling iron is necessary for curling as it is the tool used to create curls in the hair.	necessary_equipment_for_curling = ["curling stone", "curling broom"] equipment_used_in_curling_with_curling_iron = ["curling iron"] ans = ("curling iron" in necessary_equipment_for_curling)

Dataset: StrategyQA
(Geva et al., 2021)

Takeaway: ...and this applies to factual reasoning tasks as well!

Other Findings

- Compared with text-based cascade decision making (e.g., FrugalGPT)?
 - Very challenging to decide an easy vs. hard question based on textual hints
- How weak can the weaker LLM be?
 - Experiments using LLAMA2 13B
 - Takeaway: if an LLM is too weak, it won't contribute to the cost saving, because all questions will eventually be passed to the stronger LLM
- Can outputs from the weaker LLM be hints to improve the stronger LLM?
 - No, and they actually confuse the stronger LLM

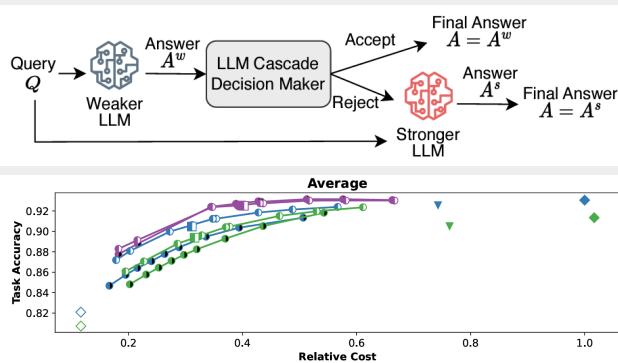
Check out our papers for more details!

Discussion & Future Work

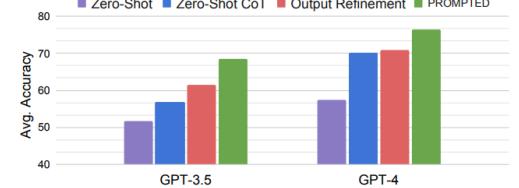
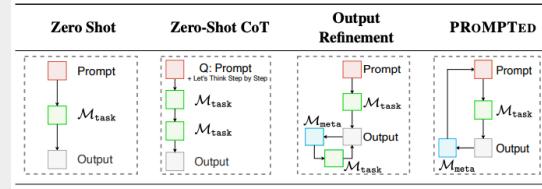
- LLM uncertainty: Does an LLM know when it doesn't know?
 - Many discussions (Kadavath et al., 2022; Xiong et al., 2023; etc.)
 - We showed the promise of mixing thought representations w/ vote-based metric
 - Generalize to tasks where we cannot vote? (e.g., text generation)
- Ensemble of multiple LLMs/LLM-powered agents
 - Concurrent work demonstrating similar synergy between CoT and PoT: e.g., model selection (Zhao et al., 2023), MAmmoTH (Yue et al., 2023)
 - Generally speaking, tasking a cohort of LLMs
 - Weaker vs. stronger, in-house vs. closed API, domain-specific vs. domain-general, etc.

This Talk: Towards Enhancing the Utilization of LLMs for Humans

Topic 1: LLM Cascade with Mixture-of-Thought for Cost-Efficient Reasoning



Topic 2: Instance-level Prompt Optimization with LLMs in the Loop



Other ongoing effort: LLM interpretability + Alignment, LLMs for education (supported by Microsoft AFMR), etc.

Instances Need More Care: Rewriting Prompts for Instances with LLMs in the Loop Yields Better Zero-Shot Performance



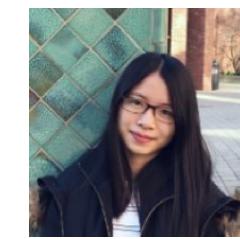
Saurabh
Srivastava



Chengyue
Huang



Weiguo Fan



Ziyu Yao

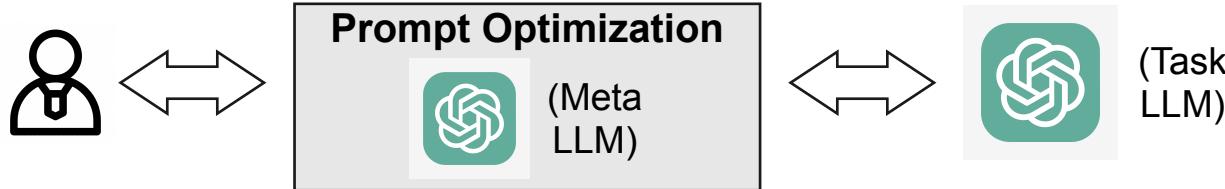
Preprint 2024

The Difficulty in Zero-Shot LLM Prompting

- “Zero shot”: prompting an LLM with only task instruction, *no* demonstrations
- The most typical interaction paradigm between users and ChatGPT, but very challenging
- What to do?
 - “Let’s think step by step” (Kojima et al., 2022) helps a bit
 - But the best practice is still to iteratively and manually revise the prompt design based on the observed effect

This Work: Instance-Level Prompt Optimization

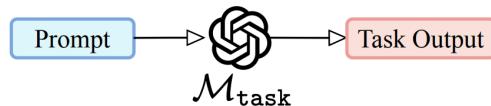
- Optimize prompts automatically for each query instance



- Employ a “Meta LLM” to oversee the “Task LLM” and devise optimal prompts based on its performance in solving the task instance
- Intuition: while the meta LLM does not know about the task (“zero shot”), it
 - knows the common practice of performing this type of task, and
 - bears sufficient knowledge to judge the task LLM’s output.

PROMPTED

Step 1. Initial Task Output Generation



i = 0 (Initial Task Prompt)

Question: Is the Very Large Telescope the most productive telescope in the world?

Choose the correct or best possible option:
(A) Yes
(B) No

(A) Yes. The Very Large Telescope the most productive telescope in the world.

No. Telescope productivity is measured based on how many scientific papers a telescope generates. The Hubble Space Telescope is the most productive telescope in the world.

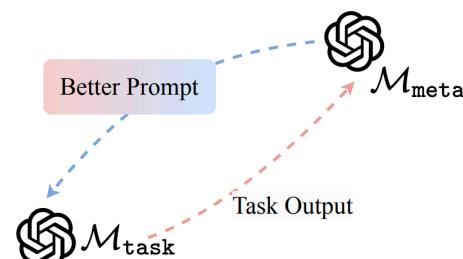
□ Prompts ■ Outputs ■ Ground Truth

Step 2. Iterative Better Prompt Generation

Reason: The output is incorrect. The claim (...) depends on how "productivity" is defined in this context.

Task Type: QUESTION ANSWERING

Better Prompt: Considering the various factors that contribute to a telescope's productivity, such as the amount of data collected, the number of scientific papers produced using its data (...)



The answer is (A) Yes. The Very Large Telescope (...)

Step 3. Final Answer Extraction

The answer is (B) No, and the reason is that (...)

(B) No

i = 1

Considering the various factors that contribute to a telescope's productivity, such as the amount of data collected, the number of scientific papers produced using its data (...)

Provide your answer in the following format: "The answer is [YOUR_ANSWER]"

i = 2

In the world of astronomy, the productivity of a telescope is a multifaceted concept. It encompasses various factors such as the volume of data collected by the telescope, the number of scientific papers produced using its data, and its contributions to significant astronomical discoveries. Considering these parameters, (...) After selecting your answer, please provide a detailed explanation (...) Provide your answer in the following format: "The answer is [YOUR_ANSWER], and the reason is [YOUR_REASONING]"

PROMPTED — Meta LLM

- Learning to rewrite prompts based on the task output



- How? Few-shot in-context learning
 - 16 demonstrations showing how to rewrite a prompt contrastively, manually prepared with ChatGPT in the loop

PROMF

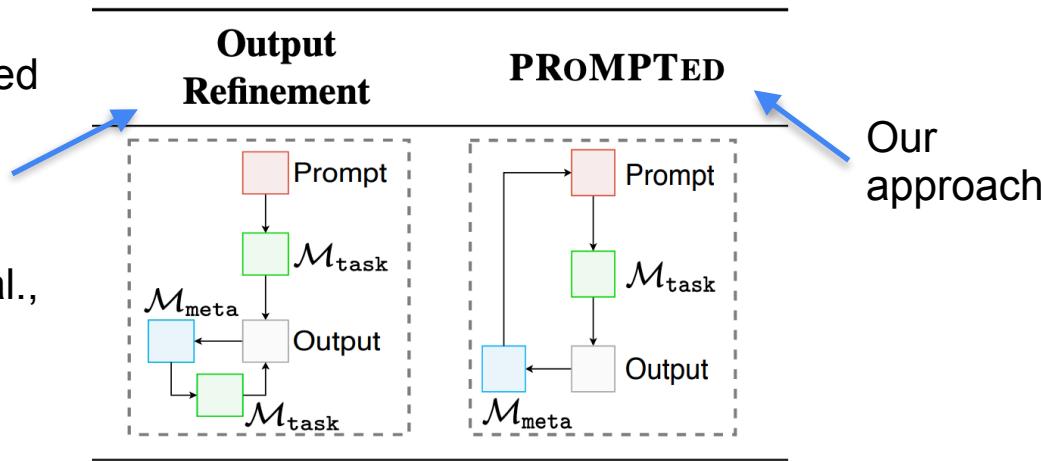
DATASET	EXAMPLE
CYNER (DOMAIN-SPECIFIC IE)	<p>INPUT: Your task is to extract the spans representing Malware, Indicators, System, Organization and Vulnerability names from the input sentence if they are present in the sentence. Remember to extract the names from the sentence not the concepts in a key:value format. The input sentence is: Chinese APK names : Some of FakeSpy 's APK package names contain anglicized Chinese (Mandarin) words that might be related to Chinese songs and lyrics , food , provinces , etc .</p> <p>ZERO-SHOT OUTPUT: {"Malware": "FakeSpy", "Indicators": "APK package names", "System": "Chinese APK", "Organization": "Chinese"} ✗</p> <p>\mathcal{M}_{meta}'s RESPONSE: The output is incorrect. (...) the output incorrectly identifies "Chinese APK" as a System (...) The output also misses out on identifying any Vulnerability names.</p> <p>BETTER PROMPT: You are tasked with extracting specific categories of information from the provided sentence. These categories include: 1) Malware: Look for specific names of malicious software or programs. (...) Remember, you should only extract the names of these elements if they are explicitly mentioned in the sentence. (...) Provide your extracted terms in a dictionary format with keys and values, where the key is the type of information(...) If a certain type of information is not present in the sentence, do not include it in your dictionary.</p> <p>PRoMPTed OUTPUT: The Answer is {"Malware": "FakeSpy"} ✓</p>
MATH (MATHE-MATICAL REASON-ING)	<p>INPUT: How many edges does an octagonal prism have?</p> <p>ZERO-SHOT OUTPUT: An octagonal prism has 18 edges. ✗</p> <p>\mathcal{M}_{meta}'s RESPONSE: Output is incorrect. Here are the reasons (...) (1) (...) (2) The Candidate prompt didn't acknowledge the possibility of conflicting information (...) (3) (...) (4) The Candidate prompt did not encourage admission of uncertainty or lack of knowledge (...)</p> <p>BETTER PROMPT: (...) Can you tell me how many edges does an octagonal prism have? It's important to note that if the information is not fully clear or if there's a possibility of multiple answers due to conflicting sources, I would appreciate it if you could mention this uncertainty. Please avoid stating the answer with absolute certainty unless you are drawing from a verified and definitive source. (...) Provide your answer in the following format: "The answer is [YOUR_ANSWER]"</p> <p>PRoMPTed OUTPUT: The answer is 24. ✓</p>

Table 3: Example output and reason generation with PRoMPTed. PRoMPTed highlights specific phrases (in █) that lead to incorrect or harmful responses. In Better Prompts, it generates a clear task instruction (in █), adds domain knowledge (in █), solution guidance (in █), output structure (in █), and specifies how to handle exceptions (in █). We note that the rewritten prompt on MATH encourages honest responses.

Rewrite the Input Prompt or Refine the LLM Output?

Refine LLM outputs based on Meta LLM-generated feedback

- Special case: self refinement (Madaan et al., 2023; Chen et al., 2023)



- Intuitively, rewriting input prompts allows for rectifying more fundamental (e.g., logic) mistakes in LLM reasoning, while refining outputs is limited to local fixes.

Experimental Results

(Meta LLM= GPT-4, Task LLM = GPT-4)

Task Types	Dataset	Zero-Shot	Zero-Shot CoT	Output Refinement	PRoMPTED
<i>Seen Task Types and Seen/Unseen Datasets</i>					
Mathematical Reasoning	GSM8K (Cobbe et al., 2021)  Seen MATH (Hendrycks et al., 2021)  Unseen				
Code Generation	HumanEval (Chen et al., 2021) 				
Logical Reasoning	Logical Deductions (Suzgun et al., 2022)  Penguins (Suzgun et al., 2022) 				
Domain-Specific Information Tasks	MedQA (Jin et al., 2020)  CyNER (Alam et al., 2022) 				
Fact Verification	FEVER (Aly et al., 2021) 				
Open-Domain Question Answering	StrategyQA (Geva et al., 2021) 				
Content Generation + Harmlessness	ToxicChats (Lin et al., 2023) 				
<i>Unseen Task Types</i>					
Domain-Specific Reading Comprehension	MMLU (PM) (Hendrycks et al., 2021) 				
Visual Reasoning	Geometric Shapes (Suzgun et al., 2022) 				
Symbolic Reasoning	LastLetterConcat (Kojima et al., 2022) 				
Average					

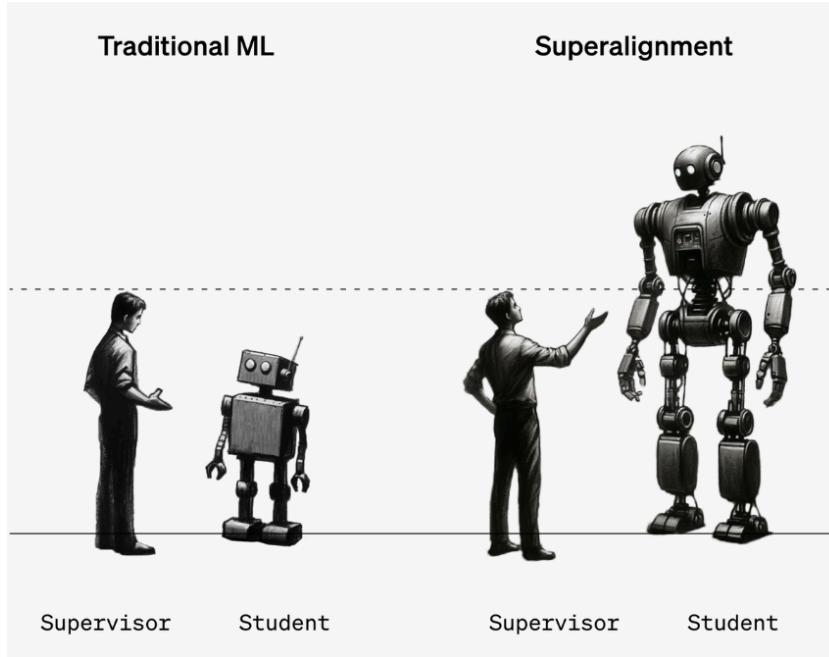
Experimental Results

(Meta LLM= GPT-4, Task LLM = GPT-4)

Task Types	Dataset		Zero-Shot	Zero-Shot CoT	Output Refinement	PRoMPTEd
<i>Seen Task Types and Seen/Unseen Datasets</i>						
Mathematical Reasoning	GSM8K (Cobbe et al., 2021) 	92.400	93.600	94.000	94.400	
	MATH (Hendrycks et al., 2021) 	48.857	56.571	57.143	61.143	
Code Generation	HumanEval (Chen et al., 2021) 	67.000	73.460	74.585	78.659	
Logical Reasoning	Logical Deductions (Suzgun et al., 2022) 	34.500	58.900	66.400	75.600	
	Penguins (Suzgun et al., 2022) 	59.286	62.143	72.734	69.434	
Domain-Specific Information Tasks	MedQA (Jin et al., 2020) 	86.800	88.800	90.400	92.800	
	CyNER (Alam et al., 2022) 	38.910	39.690	63.770	73.070	
Fact Verification	FEVER (Aly et al., 2021) 	78.800	86.800	87.600	89.200	
Open-Domain Question Answering	StrategyQA (Geva et al., 2021) 	72.000	71.600	68.000	74.000	
Content Generation + Harmlessness	ToxicChats (Lin et al., 2023) 	24.000	48.000	68.000	80.000	
<i>Unseen Task Types</i>						
Domain-Specific Reading Comprehension	MMLU (PM) (Hendrycks et al., 2021) 	87.200	88.800	68.800	91.200	
Visual Reasoning	Geometric Shapes (Suzgun et al., 2022) 	54.400	54.400	52.800	55.200	
Symbolic Reasoning	LastLetterConcat (Kojima et al., 2022) 	3.200	90.400	50.800	58.200	
Average		57.489	70.243	70.849	76.424	

Can a *Weaker* Meta LLM Oversee a *Stronger* Task LLM?

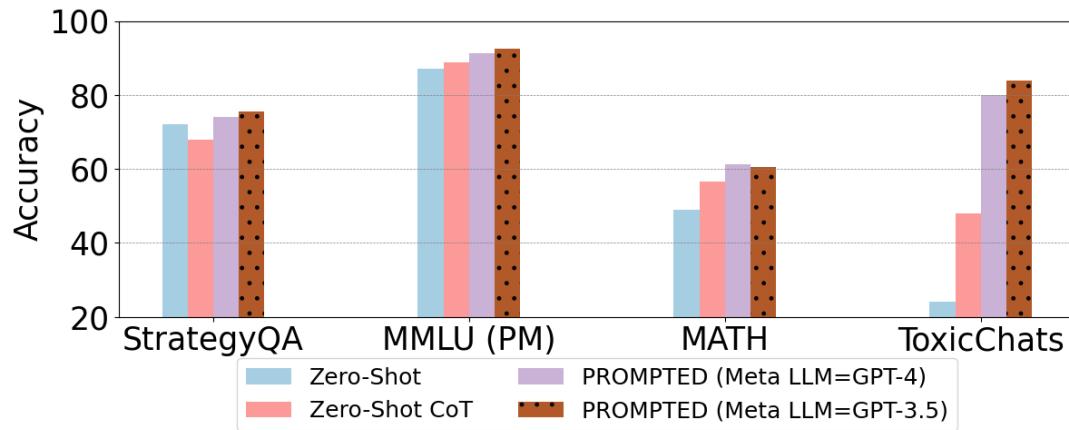
- OpenAI's recent initiative of "Superalignment"



Can a *weaker* meta LLM
(e.g., GPT-3.5)
oversee
a *stronger* task LLM (e.g.,
GPT-4)?

Can a Weaker Meta LLM Oversee a Stronger Task LLM?

- OpenAI's recent initiative of “Superalignment”



Key: “task LLM in the loop”

“evaluation” is easier than
“generation” (Leike 2022)

Takeaway: a weaker LLM can oversee and rewrite prompts for a stronger LLM

Future Work

- A lot of potential...
 - Rewriting prompts for AI safety
 - Rewriting prompts for domain-specific applications
 - “User-in-the-loop” prompt optimization: allowing users to clarify their intent



Learning to Simulate Natural Language Feedback for Interactive Semantic Parsing

Addressing “data bottleneck” & Promise for “benchmark” and “personalization”



Hao Yan



Saurabh
Srivastava



Yintao Tai



Sida I. Wang



Scott Yih

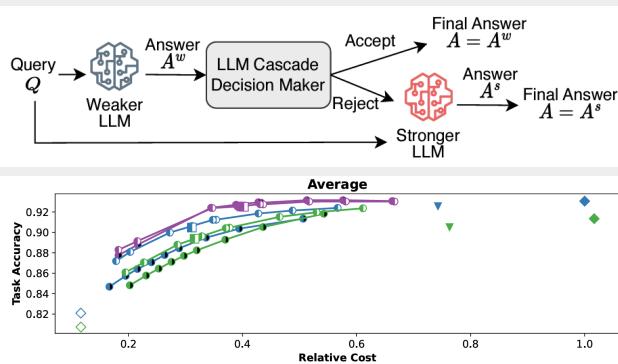


Ziyu Yao

ACL 2023

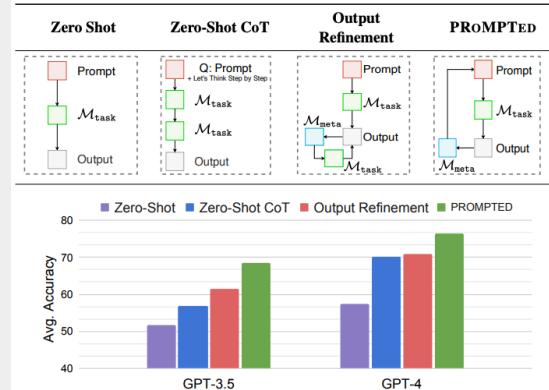
Summary: Towards Enhancing the Utilization of LLMs for Humans

Topic 1: LLM Cascade with Mixture-of-Thought for Cost-Efficient Reasoning



Summary: Mixing diverse thought representations for uncertainty measurement and cost saving

Topic 2: Instance-level Prompt Optimization with LLMs in the Loop



Summary: Employing a meta LLM to oversee and rewrite prompts for the task LLM yields better zero-shot performance

LLM Agents for Education

- LLM agents simulating students in collaborative mathematics problem solving
 - A platform for students to practice their math modeling skills
 - Helping students with limited educational resources



Now let's start a group discussion!

Alice: " So, we need to figure out the right amount of each soup flavor and type of bread roll Martha should buy based on the survey results to avoid leftovers and maximize profit."

Bob: " Right, so we need to work out the best quantities of soup and bread rolls to purchase. This means calculating the preferred soup flavors and bread types from the survey, determining the servings and bottles needed for each soup flavor, the packs of bread rolls required, and then figuring out the costs, revenue, and profit."

Charlie: " I can take on calculating the quantities and costs for the soups based on the survey results. Does anyone want to handle the bread rolls, or should we split that task too?"

Alice: " I can handle the calculations for the bread rolls, and we can review each other's findings to ensure accuracy. Does that sound good to everyone?"

Bob: " Let's start the problem solving by calculating how popular each soup flavor is according to the survey responses, which will help us determine how much of each flavor to prepare."

Alice: "Based on the survey, 15% prefer carrot and coriander, 40% like tomato, 25% choose leek and potato, and 20% go for chicken and vegetable, so for 500 customers, we'll need 75 servings of carrot and coriander, 200 of tomato, 125 of leek and potato, and 100 of chicken and vegetable, which means 8 bottles of carrot and coriander, 20 of tomato, 12.5 of leek and potato, and 10 of chicken and vegetable."

Bob: "Alice, we actually need to round up to 13 bottles for the leek and potato soup because we can't buy half a bottle, so we'll have enough for all the servings."

Alice: "Oh, you're right, Bob; we'll need to get 13 bottles for the leek and potato soup, and now I'll calculate how many packs of bread rolls we need based on the survey."

Bob: "Good catch on the soup bottles, Alice; now let's figure out the bread rolls—we'll need 38 packs of white bread and 13 packs of brown bread to match the survey preferences!"

Charlie: "Looks like we've got the right numbers for the bottles and bread packs, so let's add up the costs and see how much money Martha can make!"

Alice: "Okay, so if we add up the costs for the soup and bread rolls, we get a total cost of \$357, and since we're selling 500 mugs with bread for \$1.25 each, our total revenue will be \$625, which means our profit will be \$268!"

Charlie: "Great, the numbers look good, so let's start planning where to buy the soup and bread rolls and how we'll set up the stall!"

Alice: "Yep, the profit of \$268 sounds awesome, so let's figure out where to buy all the soup and bread rolls and get our stall ready for the market!"



Accelerating Foundation Models Research

Engaging the broader community in reimagining computing research



Murong Yue Wijdane Mifdal

In collaboration with researchers at Mathematics Education and HCI



Thank You!

Email: ziyuyao@gmu.edu
Webpage: <https://ziyuyao.org/>



Commonwealth
Cyber Initiative