



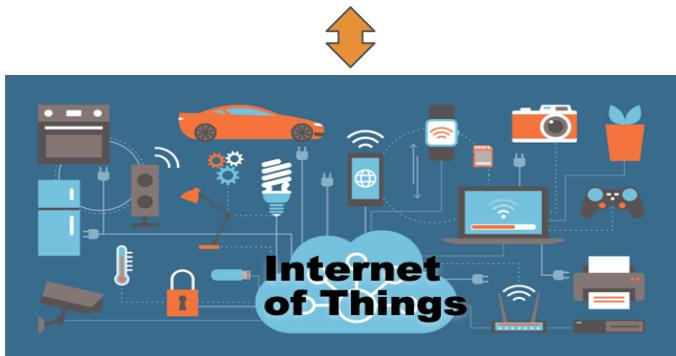
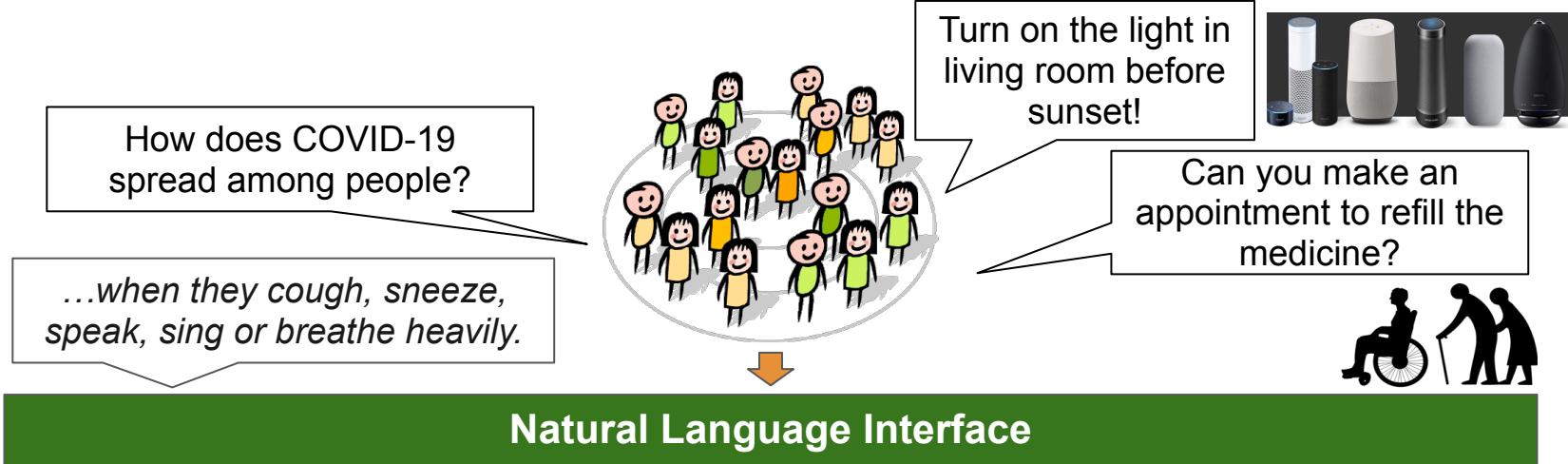
# Building Natural Language Interfaces in the Age of LLMs

Ziyu Yao

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

Department of Human-Centered Computing  
Luddy School of Informatics, Computing, and Engineering  
Indiana University, Indianapolis  
April 12, 2024

# Natural Language Interfaces (NLIs)



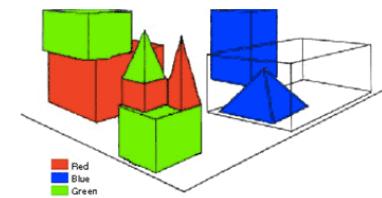
# Natural Language Interfaces (NLIs) in History

```
Welcome to
      EEEEEE LL      IIII  ZZZZZZ  AAAA
      EE  LL      II      ZZ  AA  AA
      EEEE  LL      II      ZZZ  AAAAAAA
      EE  LL      II      ZZ  AA  AA
      EEEE  LLLLLL IIII  ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU: Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU: They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU: Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU: He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU: It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU: ■
```

ELIZA (1966)



Person: Pick up a big red block.

Computer: OK.

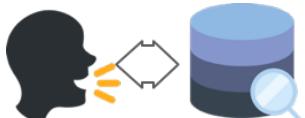
Person: Grasp the pyramid.

Computer: I don't understand which pyramid you mean.

SHRDLU (1971)



Ask Jeeves (1997)



NLI to Database (NLIDB)

LUNAR  
(1972)

CHAT-80  
(1982)

CHILL  
(1996)

Seq2Seq w  
Attention (2016)

SQLova  
(2019)

PLANE  
(1978)

ASK  
(1983)

PCCG  
(2005) DCS  
(2011)

WikiSQL Spider  
(2017) (2018)

RatSQL  
(2019)

Learning  
started

Neural net  
started

Contextual embedding  
(e.g., BERT) started

# Large Language Models (LLMs)

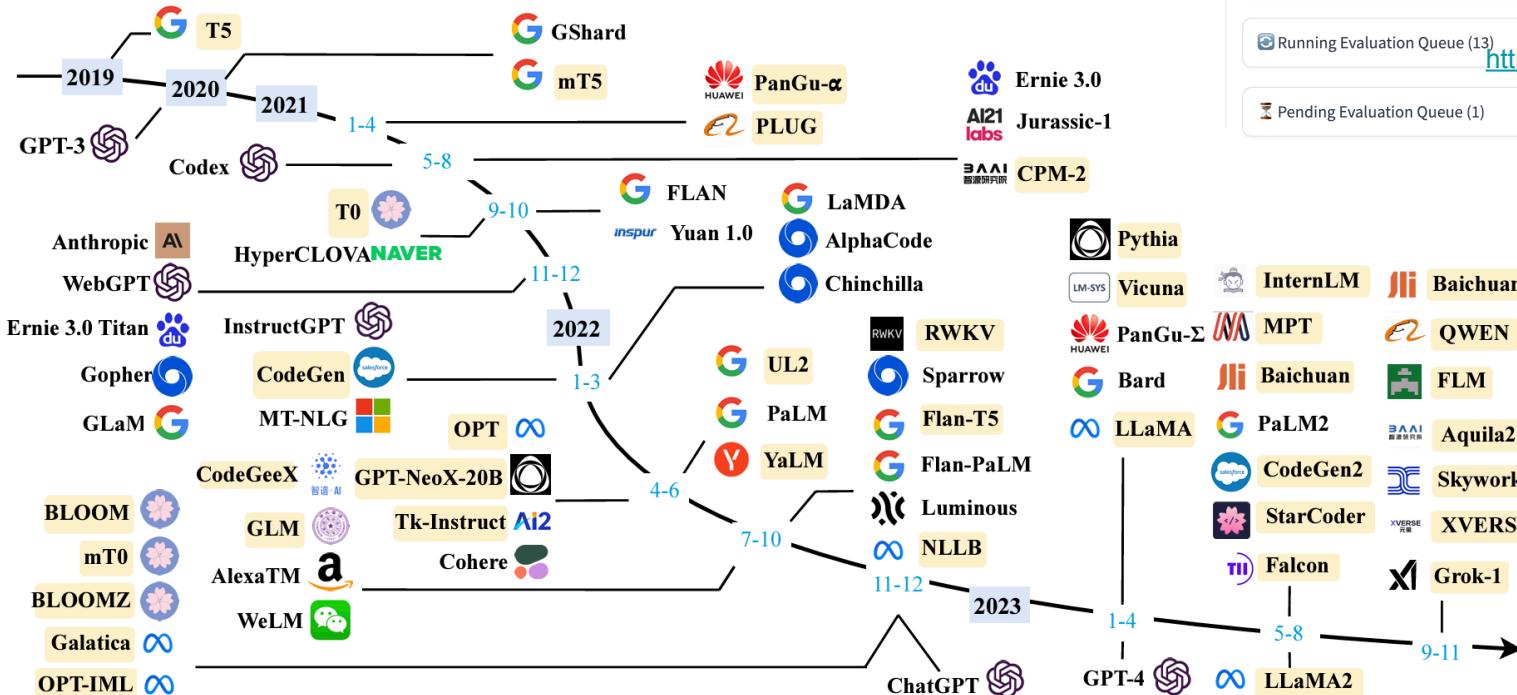


Image source: Zhao et al. "A survey of large language models." *arXiv preprint arXiv:2303.18223*.

LLM Benchmark   About   Submit here!

## Evaluation Queue for the 😊 Open LLM Leaderboard

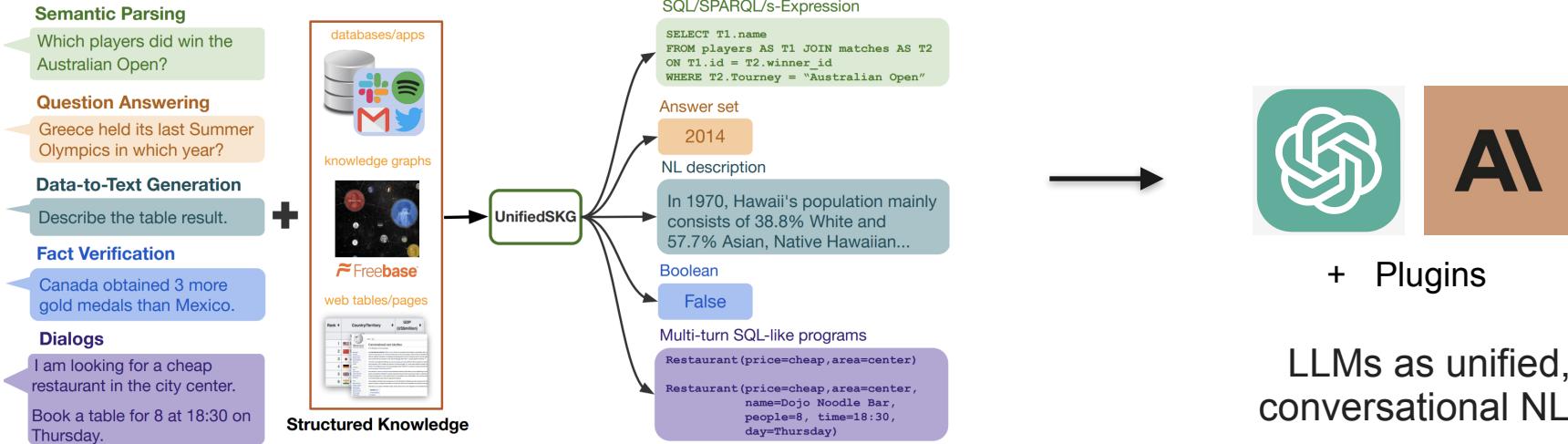
These models will be automatically evaluated on the 😊 cluster.

- Finished Evaluations (5402)
- Running Evaluation Queue (13) [https://huggingface.co/spaces/HuggingFaceH4/open\\_llm\\_leaderboard](https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard)
- Pending Evaluation Queue (1)

# 5000+ LLMs!

# NLIs in the Age of Large Language Models

Paradigm shift: unified architecture, task generalization, instruction following

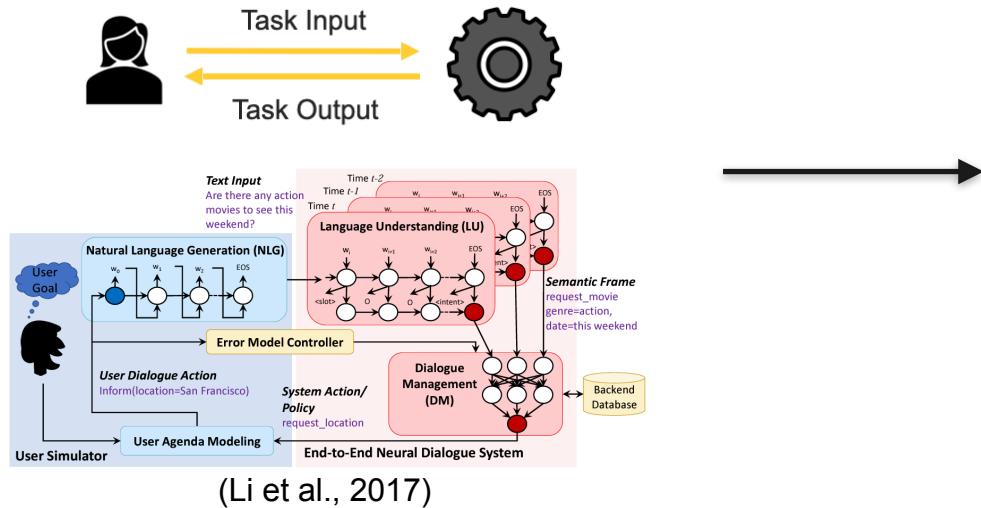


UnifiedSKG (Xie...Yao et al., 2022)

# NLIs in the Age of Large Language Models

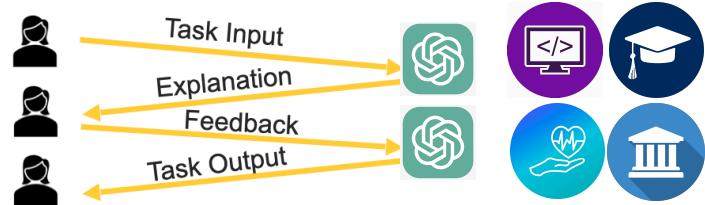
Paradigm shift: unified architecture, task generalization, instruction following

↪ Paradigm shift in how humans interact with NLIs



(Li et al., 2017)

No interaction, or task-specific interaction

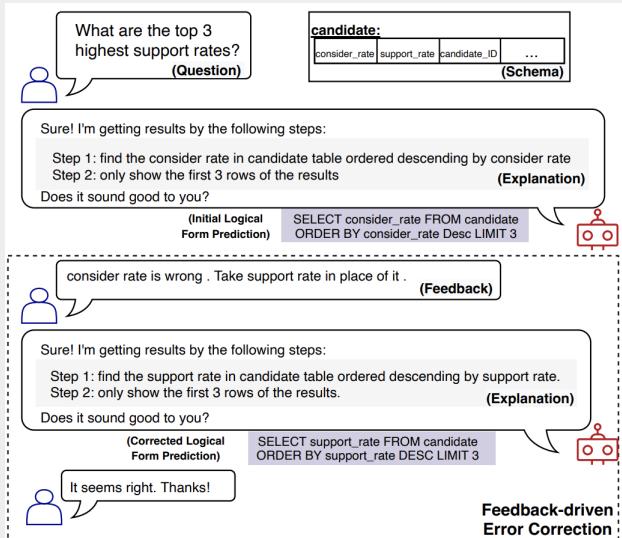


Task-agnostic, multi-turn interactions  
& Broader application areas

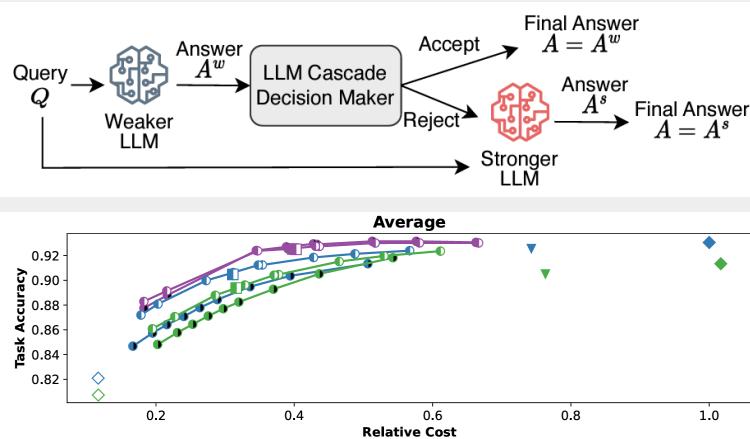
*Do LLMs interact well with humans?  
How to deal with the \$ cost of  
frequent queries to LLMs?*

# This Talk: Building NLIs in the Age of LLMs

## Topic 1: Modeling Language Feedback in Human-NLI Interaction (Task: Text-to-Code Generation)

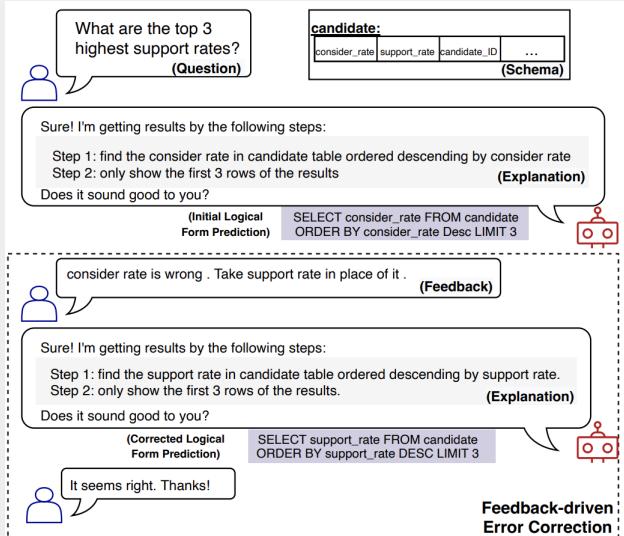


## Topic 2: Saving the Monetary Cost of LLM API Usage (Task: Arithmetic/Symbolic/etc. Reasoning)

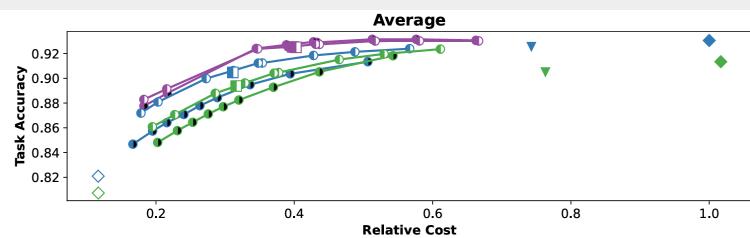
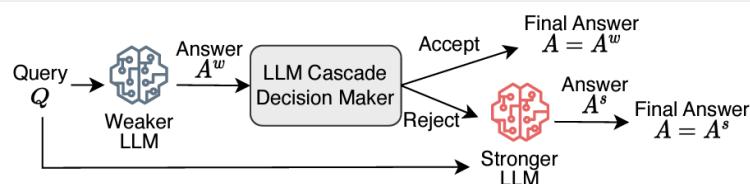


# This Talk: Building NLIs in the Age of LLMs

## Topic 1: Modeling Language Feedback in Human-NLI Interaction (Task: Text-to-Code Generation)

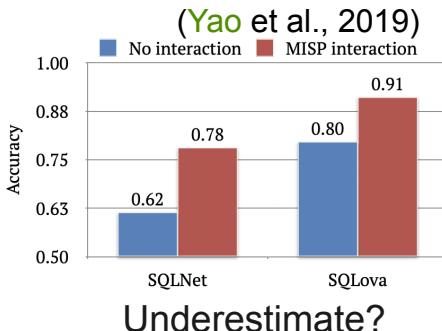


## Topic 2: Saving the Monetary Cost of LLM API Usage (Task: Arithmetic/Symbolic/etc. Reasoning)



# Feedback-driven Human-NLI Interaction

- Humans naturally provide feedback while interacting with NLIs
  - e.g., “You should not do this; the result is not what I asked for!”
- Gap: existing NLIs are rarely evaluated with human interaction
  - Need more practical assessments, i.e., *when NLIs can interact with humans*
  - Feedback understanding and incorporation: not an easy task for LLMs!

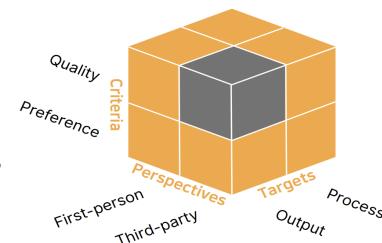


## Evaluating Human-Language Model Interaction

Mina Lee   Megha Srivastava   Amelia Hardy   John Thickstun   Esin Durmus  
Ashwin Paranjape   Ines Gerard-Ursin<sup>§</sup>   Xiang Lisa Li   Faisal Ladak  
Frieda Rong   Rose E. Wang   Minae Kwon   Joon Sung Park   Hancheng Cao  
Tony Lee   Rishi Bommasani   Michael Bernstein   Percy Liang

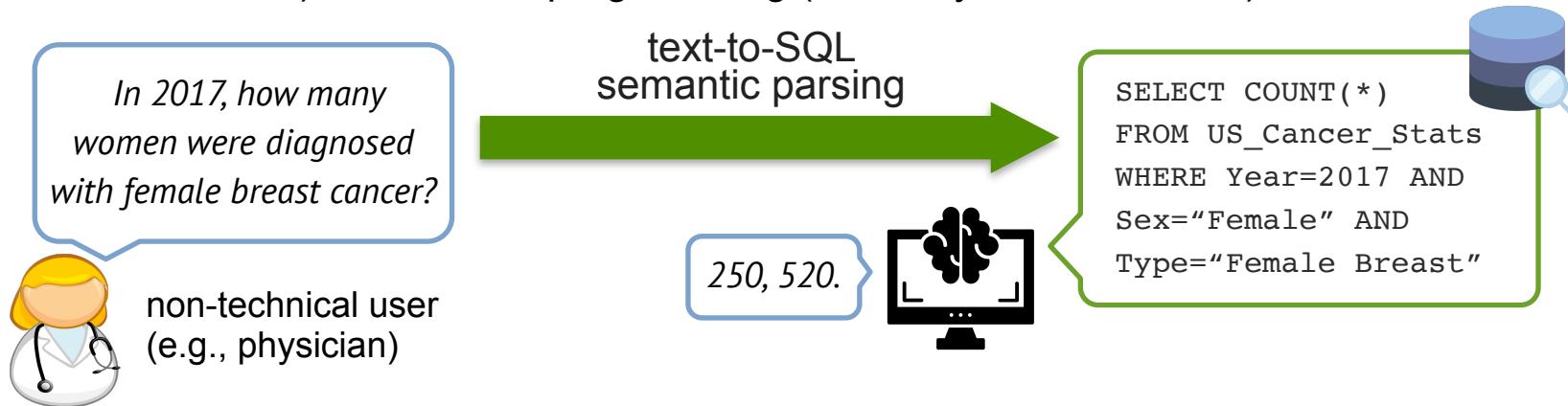
Stanford University   <sup>§</sup>Imperial College London

... or Overestimate?



# Semantic Parsing

- Translating a natural language (NL) question/command to its logical meaning representation
  - e.g., NL-to-SQL parsing for database querying
  - Other applications: robotics (NL-to-LTL), knowledge base query (NL-to-Lambda Calculus), AI-assisted programming (NL-to-Python/Java/C/...)

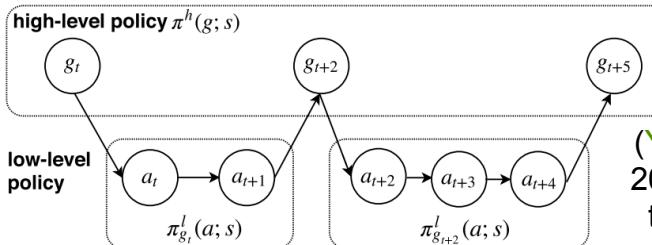
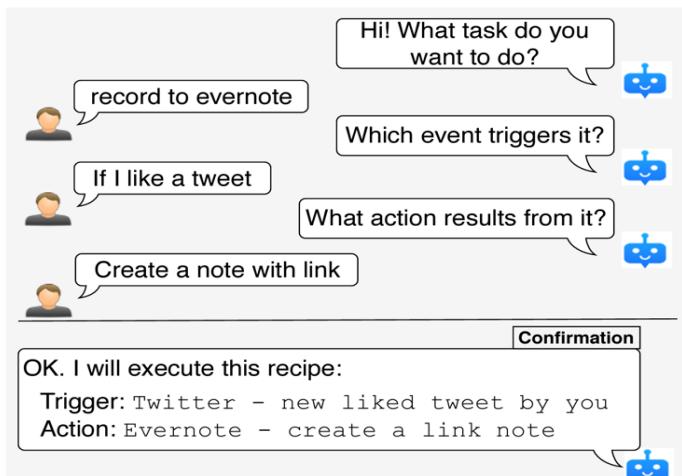


(Zhong et al., 2017; Yu et al., 2018)

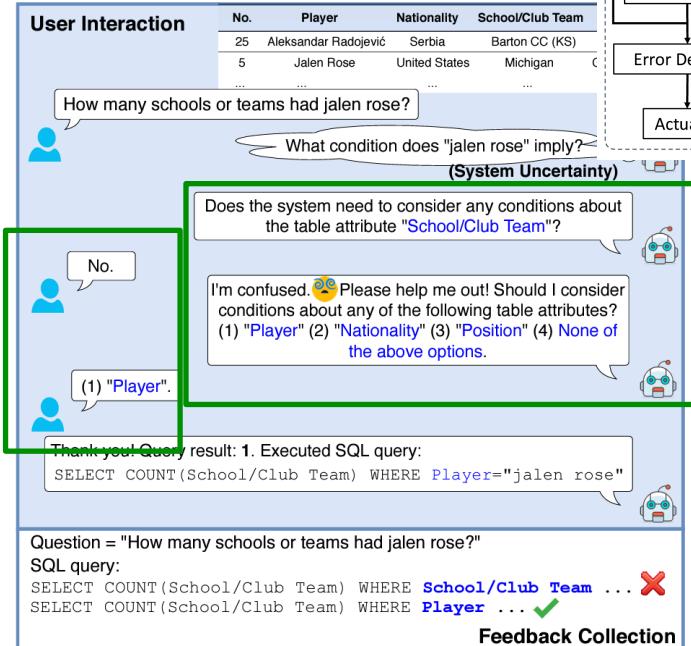
<https://gis.cdc.gov/Cancer/USCS/DataViz.html> 10

# Interactive Semantic Parsing/Code Generation

- Semantic parsing with humans proving clarification and corrective feedback



User Feedback



System Explanation

(Yao et al., 2019&20)  
A general framework, showcased in Text-to-SQL

# Interactive Semantic Parsing/Code Generation

- SPLASH dataset by Microsoft Research: text-to-SQL with natural language (NL) feedback

The data bottleneck:  
Costly and *model-dependent* feedback  
annotation



Find all the locations whose names contain the word "film"

finding the Address of Locations table for which Location\_Name contains "film"

Address

770 Edd Lane Apt. 098

14034 Kohler Drive



Address is wrong. I want the name of the locations

finding the Location\_Name of Locations table for which Location\_Name contains "film"

Location\_Name

Film Festival

Film Castle

...

System  
Explanation

User's Corrective  
Feedback in NL

(Elgohary et al., 2020)

# Learning to Simulate Natural Language Feedback for Interactive Semantic Parsing



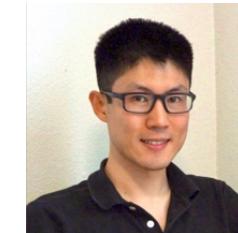
Hao Yan  




Saurabh  
Srivastava  




Yintao Tai  

Sida I. Wang  




Scott Yih  

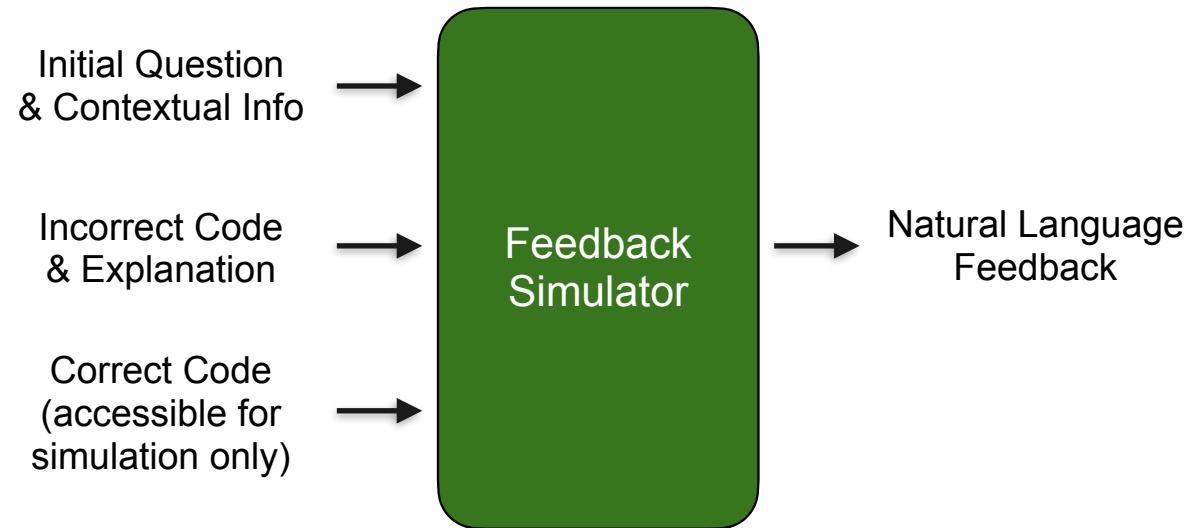



Ziyu Yao  


ACL 2023

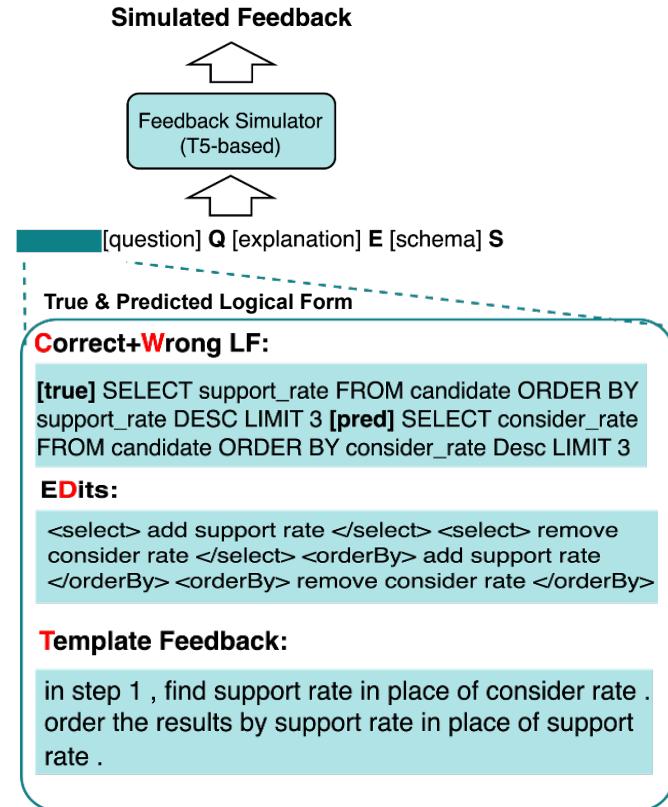
# Learning to Simulate Natural Language Feedback

- Idea:
  - *Build* a simulator with *small-scale* feedback annotations
  - *Apply* the simulator to generate *large-scale* synthetic feedback for model training



# Learning to Simulate Natural Language Feedback

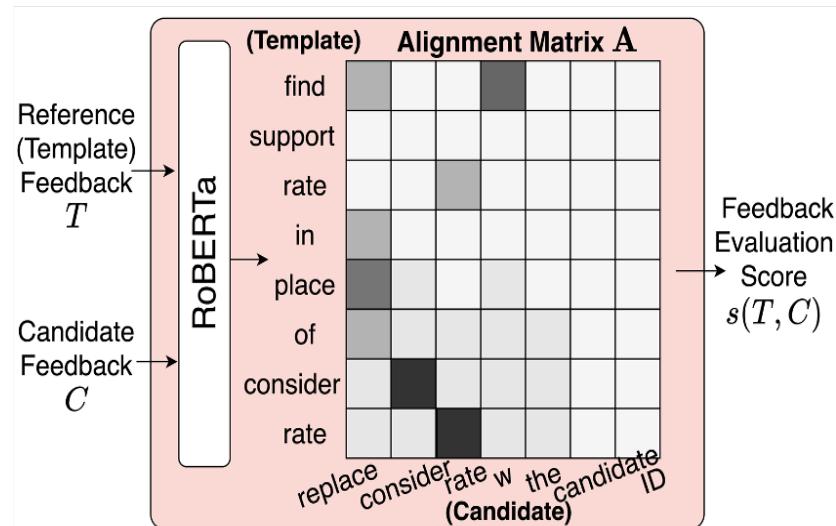
- The importance of **task representations** (“prompt engineering”):
  - **CWQES**: Simply include the **Correct** and **Wrong** code snippets as input.
  - **DQES**: Inspired by NL-Edit (Elgohary et al. 2021), feed the **EDits** of revising the incorrect code snippet into the correct one.
  - **TQES**: Verbalize the edits using **Templates**.



# Evaluating the Faithfulness of the Simulated Feedback

- **Faithfulness:** Does the simulated feedback precisely reflect the user intent of error correction?
  - Traditional metrics such as BLEU (Papineni et al., 2002) cannot measure it
  - More recent metrics such as BERTScore (Zhang et al., 2019) are too generic
- Our approach: fine-tuning BERTScore with contrastive examples
  - **Template feedback** as reference

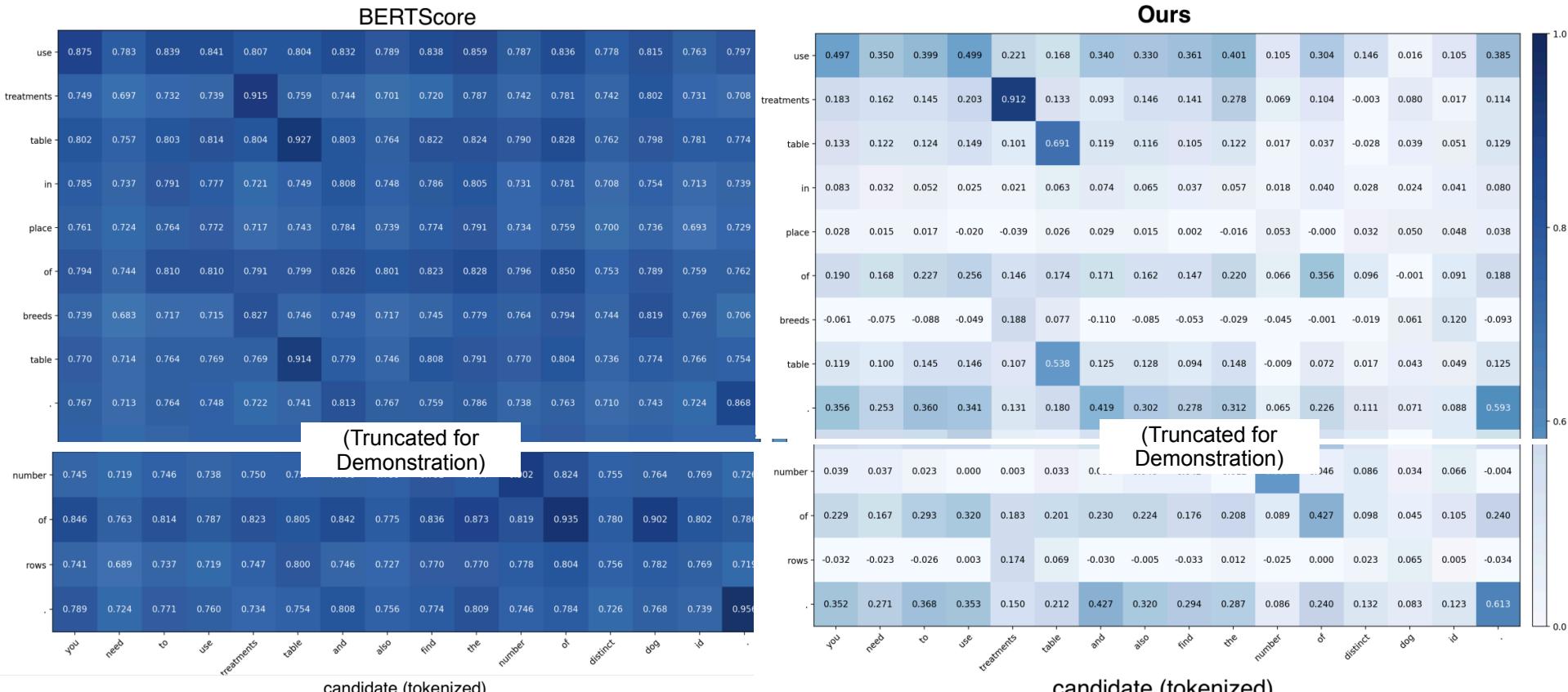
Metrics	MRR (dev)	Human
BLEU	0.57	0.03
BERTScore	0.55	0.08
Our Evaluator	<b>0.88</b>	<b>0.19</b>



$$s(T, C) = \frac{1}{2} \left( \frac{1}{M} \sum_{m=1}^M \max_n \mathbf{A}_{nm} + \frac{1}{N} \sum_{n=1}^N \max_m \mathbf{A}_{nm} \right)$$

(Please refer to details in our paper)

# Evaluating the Faithfulness of the Simulated Feedback



# Example

Model	BLEU	BERTScore	Our Evaluator
CWQES	0.132	0.881	0.491
DQES	<b>0.134</b>	0.882	0.518
TQES	0.125	<b>0.884</b>	<b>0.535</b>

## Easy Example from SPLASH-dev

- Question: How many dogs went through any treatments?
- Correct Parse: SELECT count(DISTINCT dog\_id) FROM treatments
- Wrong Parse: SELECT count ( \* ) FROM breeds
- Explanation: find the number of rows in breeds table
- Template Feedback: use treatments table in place of breeds table . find number of different dog id in place of number of rows .
- Human Feedback: Change breeds table with treatments table .

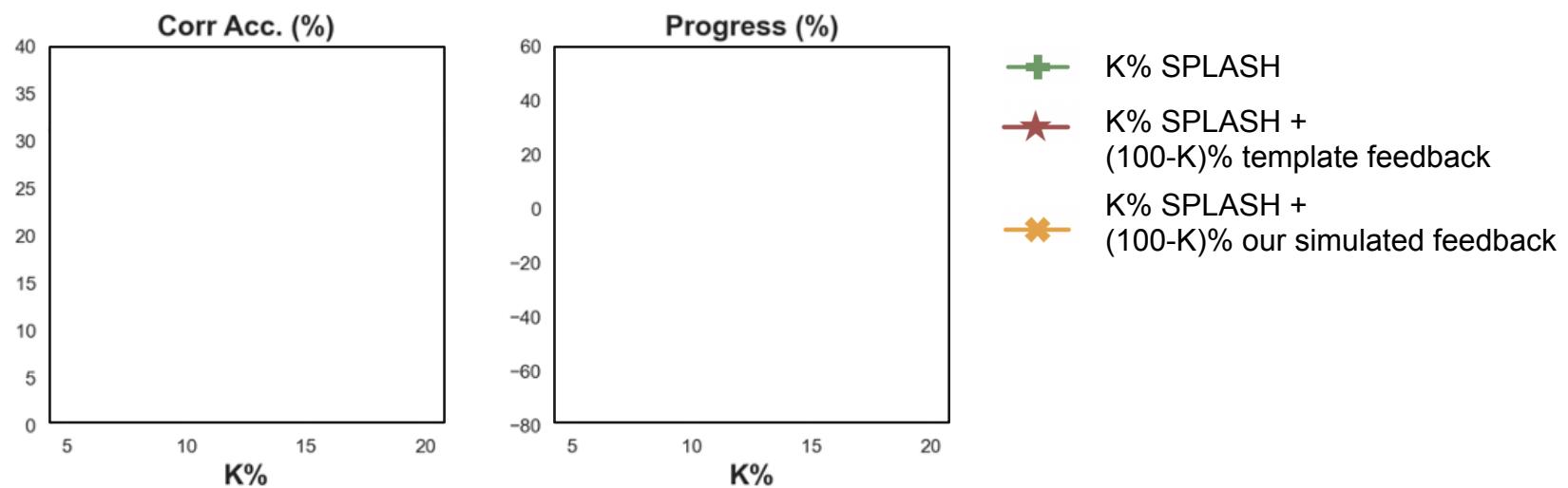
## Simulated Feedback & Evaluation Results

- CWQES you need to use treatments table in place of breeds table .  
BLEU: 0.308, BERTScore: 0.876, Ours: 0.468
- DQES you need to use treatments table and search for the number of distinct dog id .  
BLEU: 0.063, BERTScore: 0.879, Ours: 0.528
- TQES you need to use treatments table and also find the number of distinct dog id .  
BLEU: 0.065, BERTScore: 0.889, Ours: 0.529

Our evaluator is better than  
BERTScore in capturing differences in  
simulated feedback

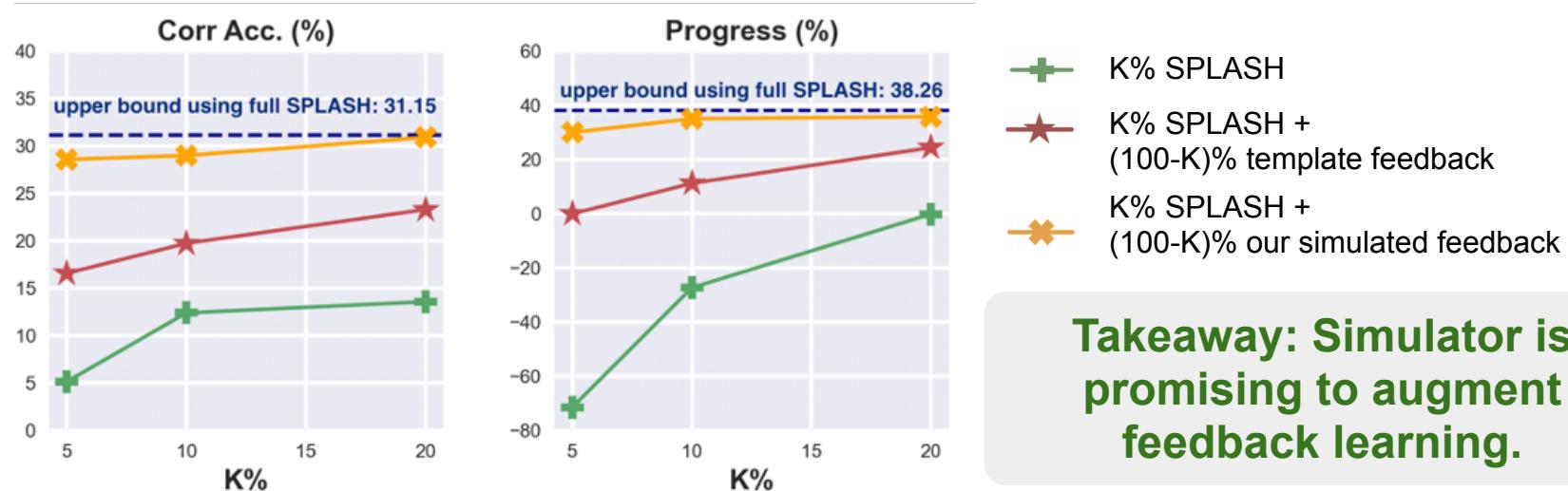
# Experimental Results

- “Low data” experiment: train a simulator with a small amount of feedback annotations, and apply it to synthesize more for model training
  - Text-to-SQL. Performance on error correction based on feedback.



# Experimental Results

- “Low data” experiment: train a simulator with a small amount of feedback annotations, and apply it to synthesize more for model training
  - Text-to-SQL. Performance on error correction based on feedback.

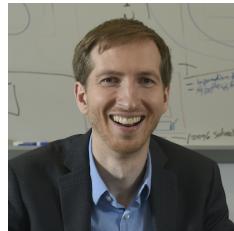


# Discussion

- While we were working on the project (late 2022), ChatGPT came out...
- Are problems solved with ChatGPT?
  - If it does, this feedback simulator is not necessary:(
  - However, NO! Feedback modeling is not trivial even for ChatGPT!
  - Could be even more challenging with real human users, e.g., humans may not fully understand the code explanation



Hao Yan 



Thomas LaToza 



Ziyu Yao 

Work in Progress, 2024

# Interactive Code Generation w/ ChatGPT-3.5

- Focus: *non-professional programmers* who have basic knowledge of computation and mathematics but are not professional in programming
- Very painful for them to interact with vanilla ChatGPT for programming
  - Users cannot understand or verify complicated code (they are not professional!)
  - Unstructured, back-and-forth queries for code explanation lead to frustration
- Experimental tasks: text-to-SQL and Python code generation

# Text-to-Python

## LLM-Generated Code Explanation

## User NL Feedback

Text-to-SQL    Code Generation

A Test Input  
The input of tests

```
move_num('I love you 143you55three3000thousand')
move_num('Avengers124Assemble')
move_num('It's our path to 12path13tol4see15things16do17things')
```

B Expected Output  
Expected outputs

```
'I love you three thousand1143553000'
'AvengersAssemble124'
'It's our path to see things do things11121314151617'
```

B Execution Results  
Actual outputs by running the answer code against tests

```
I love you three thousand1143553000
AvengersAssemble124
It's our path to see things do things11121314151617
```

C Chatbot

Write a function to move all the numbers in it to the given string.

I generated a code that cannot pass all test cases. Can you tell me what is wrong with my code? Here is the description of my code:

This program takes a string as input and separates the numbers and characters from the string. It then sorts the numbers in ascending order and returns a new string with the characters followed by the sorted numbers.

The execution results is shown on the top right.

You do not need to sort the number.

I've finalized my answer baed on your inputs. Here is it:

```
def move_num(string):
    nums = []
    chars = []
    for char in string:
        if char.isdigit():
            nums.append(char)
        else:
            chars.append(char)
```

D

Human-L

Complete

submit

# User Study Results

**Takeaway: still many challenges for LLMs serving as interactive NLIs!**

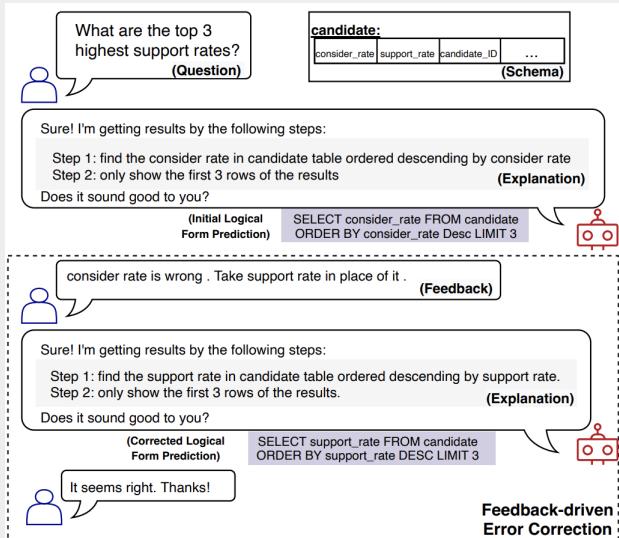
- Overall, how does our system help users in programming?
  - Double the success rate of vanilla ChatGPT-3.5, but still large room for improvement (20% for SQL and 50% for Python)
- Can users identify potential problems from our code explanation?
  - Yes but not always, for ~50% (SQL) and ~80% (Python) of the incorrect generations
- How do users provide NL feedback when they identify problems?
  - Direct instruction for error correction (58% for SQL and 70% for Python), question rephrasing, or step-by-step instructions
- Can the LLM understand the user feedback and successfully incorporate it for error correction?
  - Still very challenging! e.g., 35% (SQL) and 65% (Python) success rates for “direct instruction for error correction” feedback type

# Open Research Problems

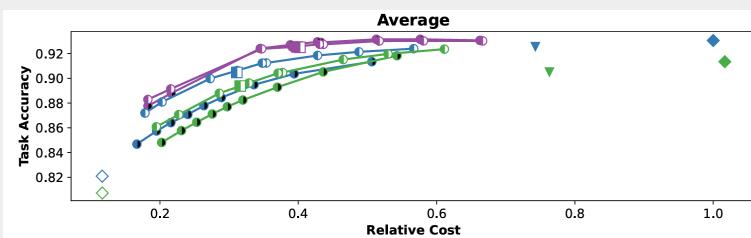
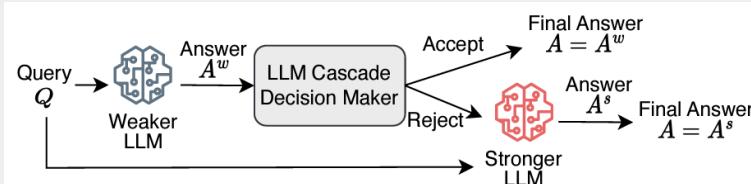
- Future of human-LLM interaction
  - How to prompt LLMs to generate explanations that are *helpful to users*?
  - Psychological problems, e.g., cognitive bias, sycophancy (Wei et al., 2023)
  - Personalization requires modeling users beyond their feedback
- Improve *human feedback following*
  - Many efforts on *instruction following* (Webson and Pavlick 2022; Jang et al., 2022)
  - Being more challenging given the huge language variation of human feedback
- Benchmark for human-LLM interactions
  - Still an understudied field. Recent work: MINT (Wang et al., 2023)
  - Our work characterized how humans express feedback in AI-assisted programming

# This Talk: Building NLIs in the Age of LLMs

## Topic 1: Modeling Language Feedback in Human-NLI Interaction (Task: Text-to-Code Generation)



## Topic 2: Saving the Monetary Cost of LLM API Usage (Task: Arithmetic/Symbolic/etc. Reasoning)



# 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!

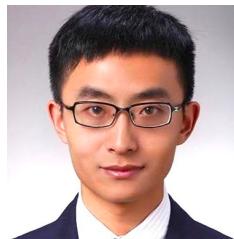


**How can we save \$ without sacrificing task performance?  
(Focus: Reasoning tasks)**

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



Murong Yue 



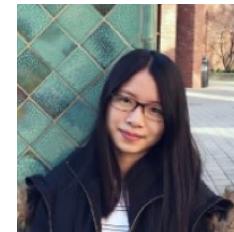
Jie Zhao 



Min Zhang 



Liang Du 



Ziyu Yao 

ICLR 2024

# LLM Cascades for Cost Saving

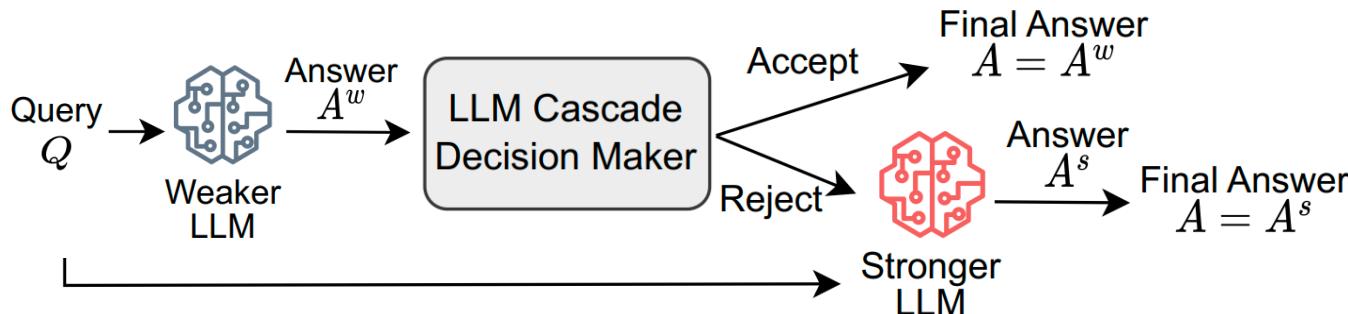
Intuition: easy questions can be handled by relatively weaker (and cheaper) LLMs to save \$.

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*



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

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")

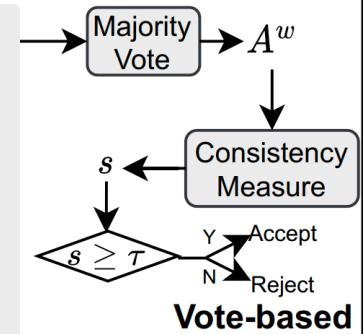
# This Work: Answer Consistency-based Decision Making

- 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 certainty on an answer?
  - See how often it *samples* the same answer to the given question
  - 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?

# Approaches

- Vote-based decision making

Sampled  $K$  Answers



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

# Approaches

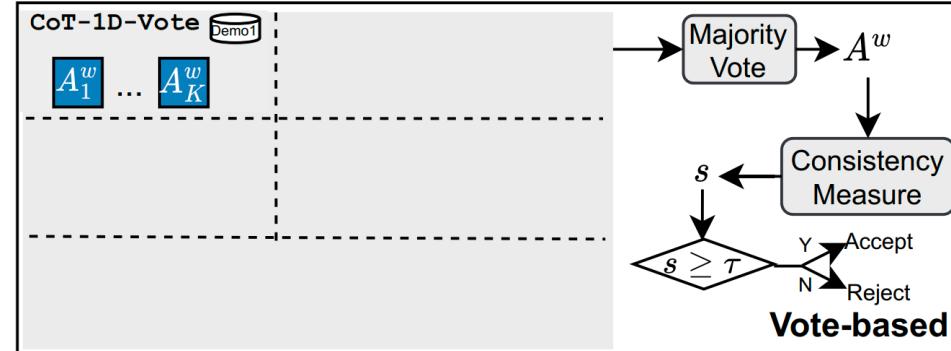
- Vote-based decision making, sampling from
  - a single thought representation
  - a single demonstration set

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

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

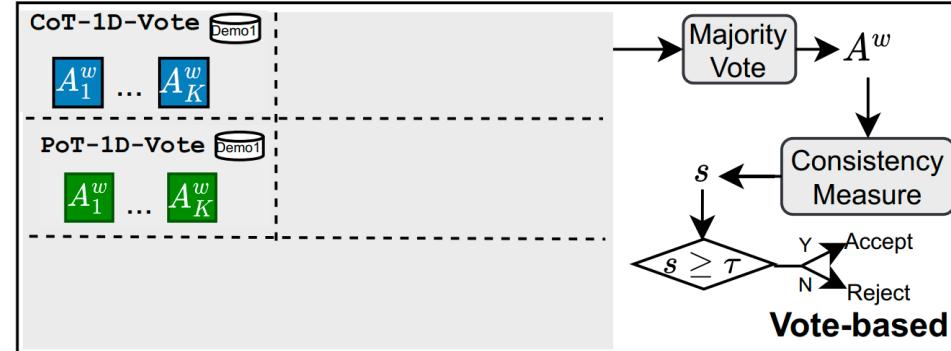
Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?

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

- Vote-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)

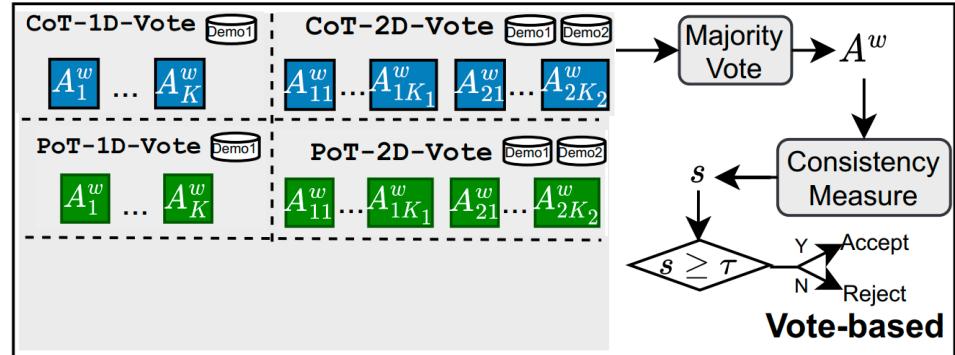
Q: Test question  
 A:



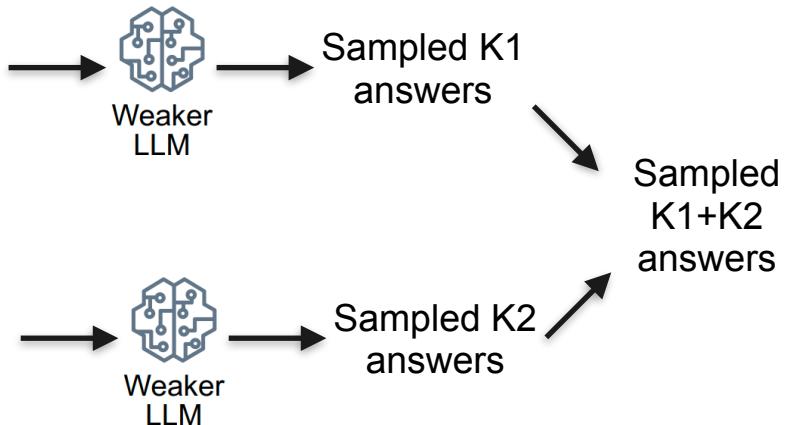
Q: Manny had 3 birthday cookie pies to share with his 24 classmates and his teacher, Mr. Keith. ...  
 A: There is a total of  $3 \times 10 = 30$  cookie slices... ans = 4

... (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

- Vote-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)

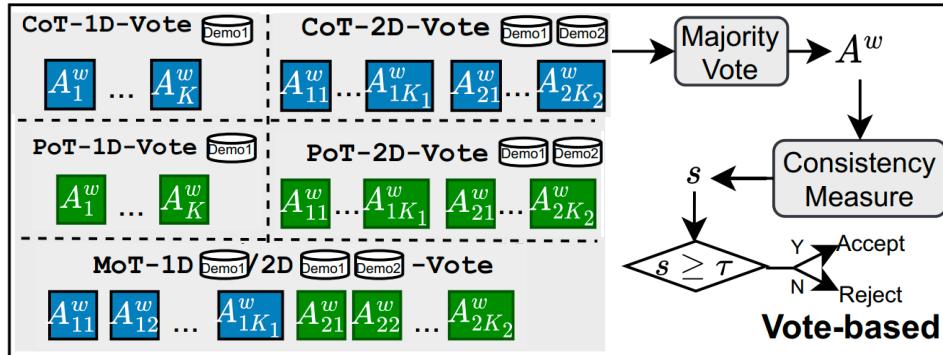
Q: Test question  
 A:



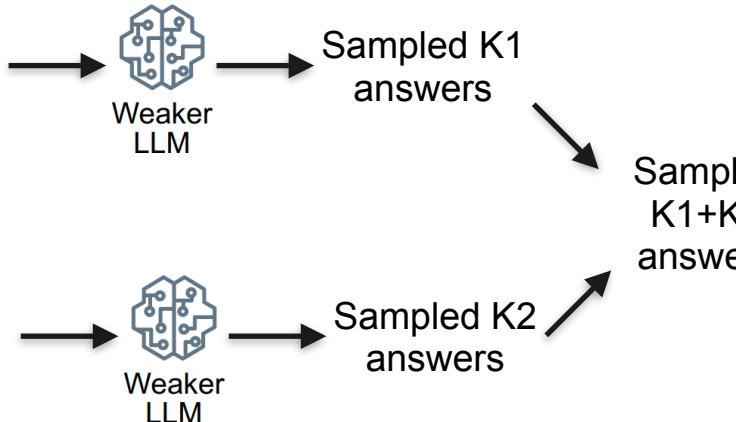
Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?  
 A:  
 # Python code, return ans  
`#ans = bolts_of_blue_fiber + bolts_of_white_fiber`

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

Q: Test question  
 A:



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



# Approaches

- Vote-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)

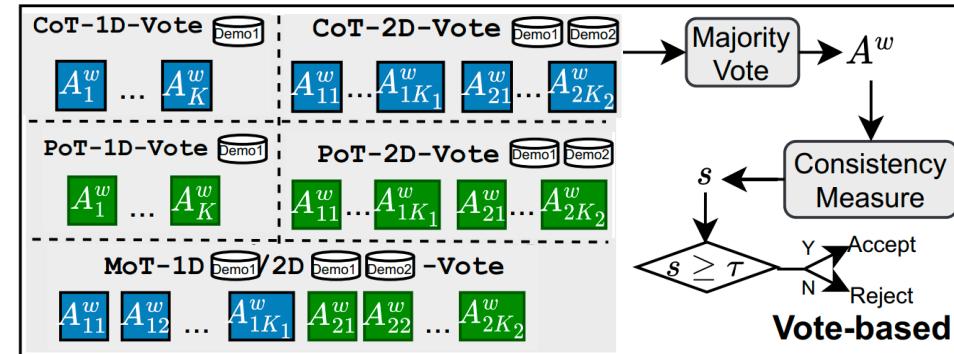
Q: Test question  
 A:



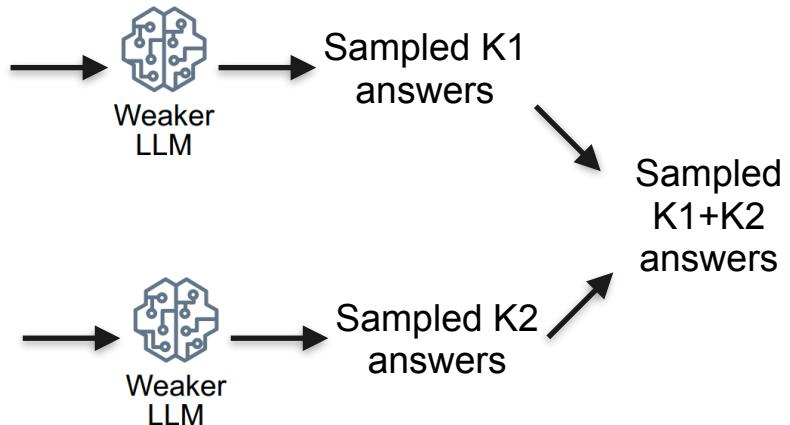
Q: Manny had 3 birthday cookie pies to share with his 24 classmates and his teacher, Mr. Keith. ...  
 A:  
 # Python code, return ans  
 ...ans = total\_cookie\_pies - total\_person\_count

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

Q: Test question  
 A:



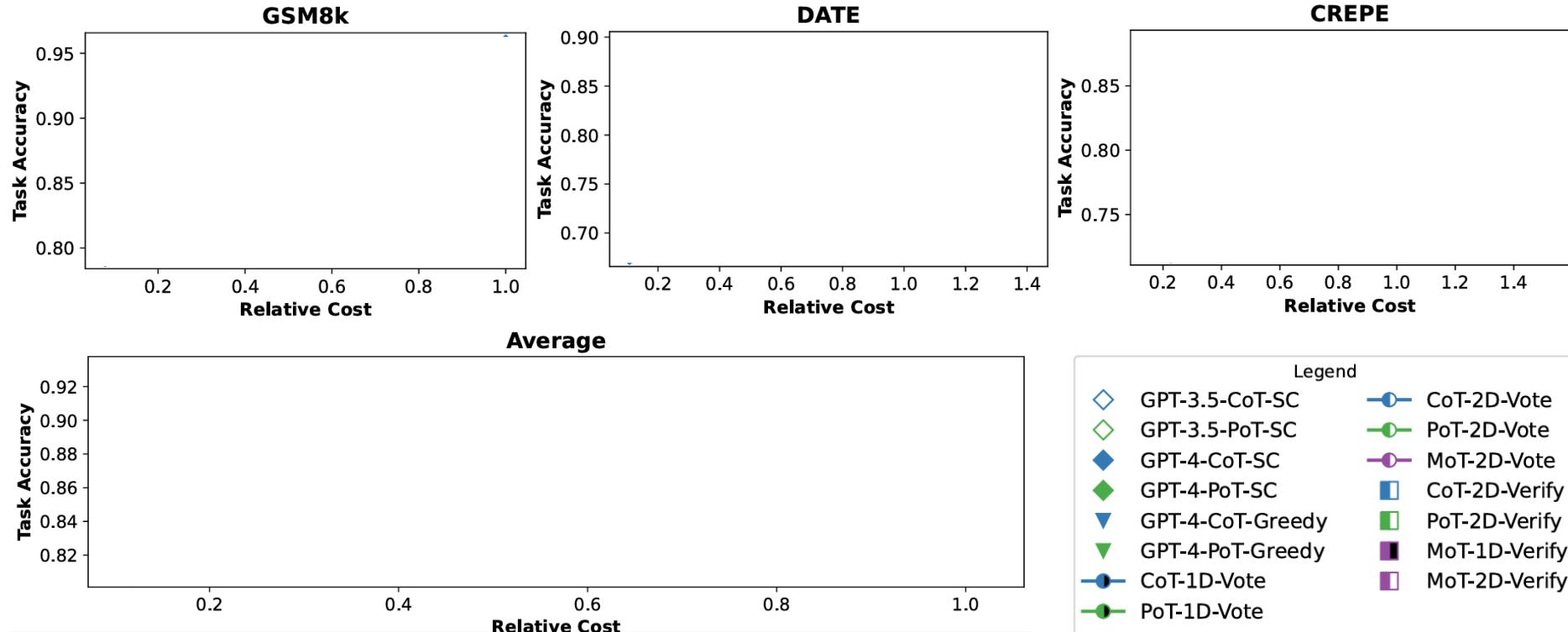
$$s = \frac{\sum_{i=1}^{K_1} \mathbb{1}_{A_{1i}^w = A^w} + \sum_{i=1}^{K_2} \mathbb{1}_{A_{2i}^w = A^w}}{K_1 + K_2}$$



Method: MoT-2D-Vote

# Experimental Results

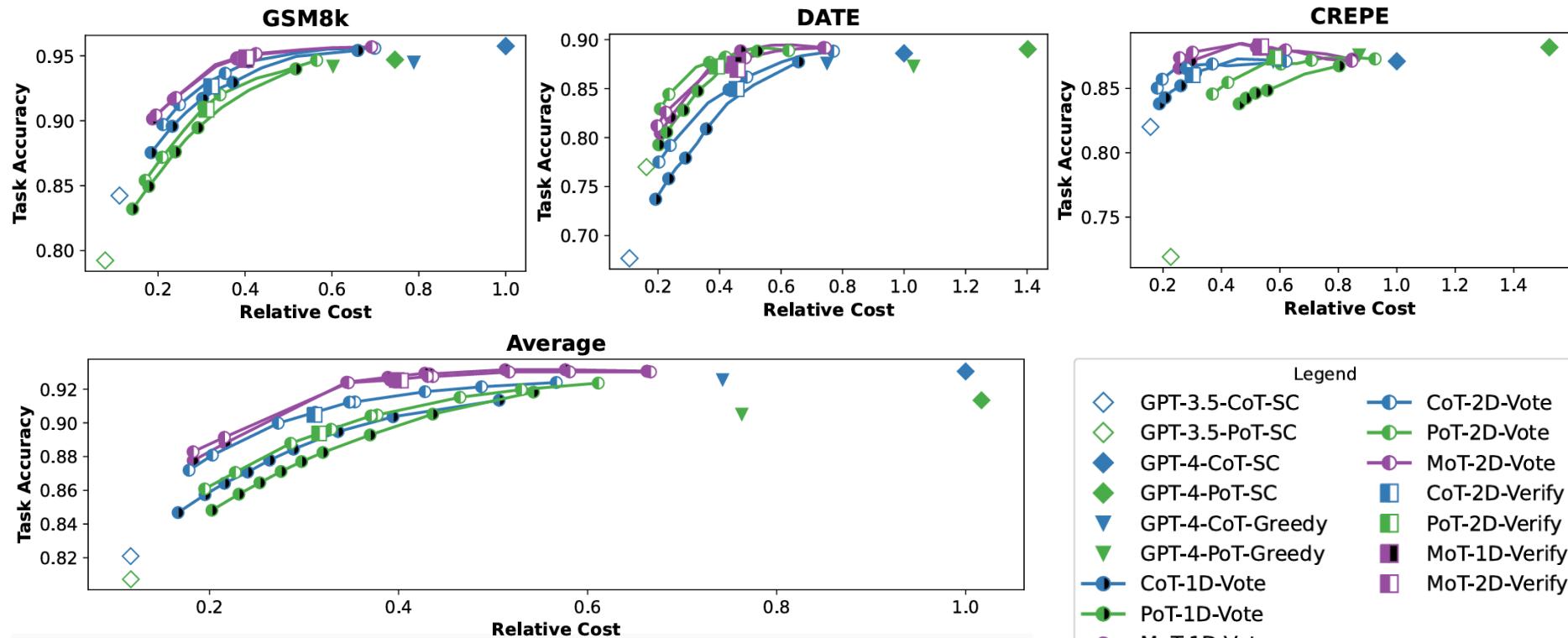
Weaker LLM: GPT-3.5-turbo  
Stronger LLM: GPT-4



(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) introduces diverse “opinions”, similar to *model ensemble*, which helps uncertainty measurement

**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

# Diversify Thought Representations for Uncertainty Measurement

- Mixture of Thought (MoT) introduces diverse “opinions”, similar to *model ensemble*, which helps uncertainty measurement
  - Applies to factual reasoning tasks as well

<p>Question</p> <p>Is a curling iron necessary in curling?</p>	<p>Golden</p> <p>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.</p>
<p>CoT</p> <p>Yes, a curling iron is necessary for curling as it is the tool used to create curls in the hair.</p>	<p>PoT</p> <pre>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)</pre>

Dataset: StrategyQA  
(Geva et al., 2021)

# Other Findings

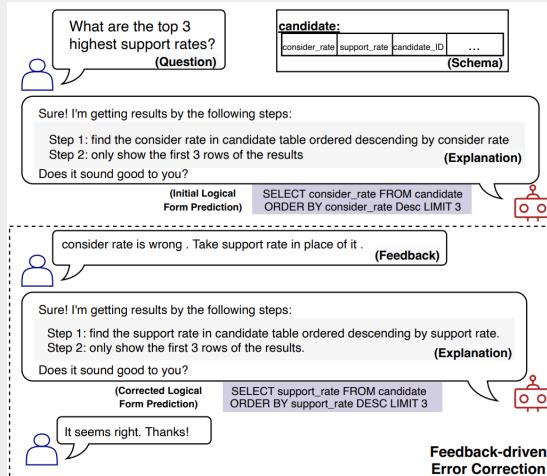
- Decision-making based on textual hints (e.g., FrugalGPT)?
  - Takeaway: it is very challenging to distinguish between easy and hard questions solely based on textual hints
- How weak can the weaker LLM be?
  - Experiments using LLAMA2 13B
  - Takeaway: if an LLM is too weak, it won't contribute to the cost saving, i.e., 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

# 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
  - Similar synergy between CoT and PoT: e.g., model selection (Zhao et a., 2023), fine-tuning (Yue et al., 2023)
  - Generally speaking, tasking a cohort of LLMs, e.g., weaker vs. stronger, in-house vs. closed API, domain-specific vs. domain-general, etc.

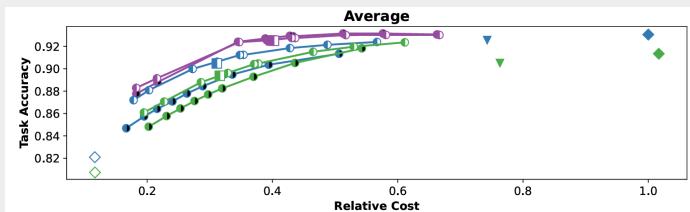
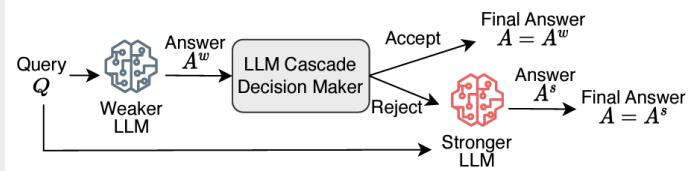
# This Talk: Building NLIs in the Age of LLMs

## Topic 1: Modeling Language Feedback in Human-NLI Interaction (Task: Text-to-Code Generation)



Building simulators for feedback modeling; still challenges for LLMs as interactive NLIs

## Topic 2: Saving the Monetary Cost of LLM API Usage (Task: Arithmetic/Symbolic/etc. Reasoning)



LLM cascades with Mixture-of-Thought decision-making helps uncertainty measurement and enables cost efficiency

# New Preprint: LLM Agents for Education

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

## Problem Description

Martha hopes to sell 500 mugs of soup, each with a white or brown bread roll. She will sell a mug of soup with a bread roll for \$1.25. She can buy the soup in 2.5 liter. Each bottle of soup costs \$5 and provides ten servings. Bread rolls are sold in packs of 10. Each pack costs \$2. To better meet the requirement, she made a preference survey. The response cards are shown in the left Data Panel. What exactly should Martha buy so that she can make the most profit?

## Data Panel

Martha made a preference survey and received the following 40 responses:



Alice Hi guys, we should think about which flavors are popular to avoid wasting any soup.

Bob I agree! Let's say she decides on those flavors! Now how do we decide which flavor to buy?

Charlie That's exactly what the survey responses told us! I counted it a bit. Among 40 people, 15 pick tomato and  $15/40=0.375$ . So I think it says 0.375% people like tomato!

Human Student Wait, did you count it right? I found 16 tomato actually... and the percentage should be multiplied by 100!



Send



Skip



## Accelerating Foundation Models Research

Engaging the broader community in reimagining computing research



Murong Yue  
(AI/LLM)      Wijdane Mifdal  
(AI/LLM)



Janice Zhang  
(HCI)



Ziyu Yao  
(AI/LLM)

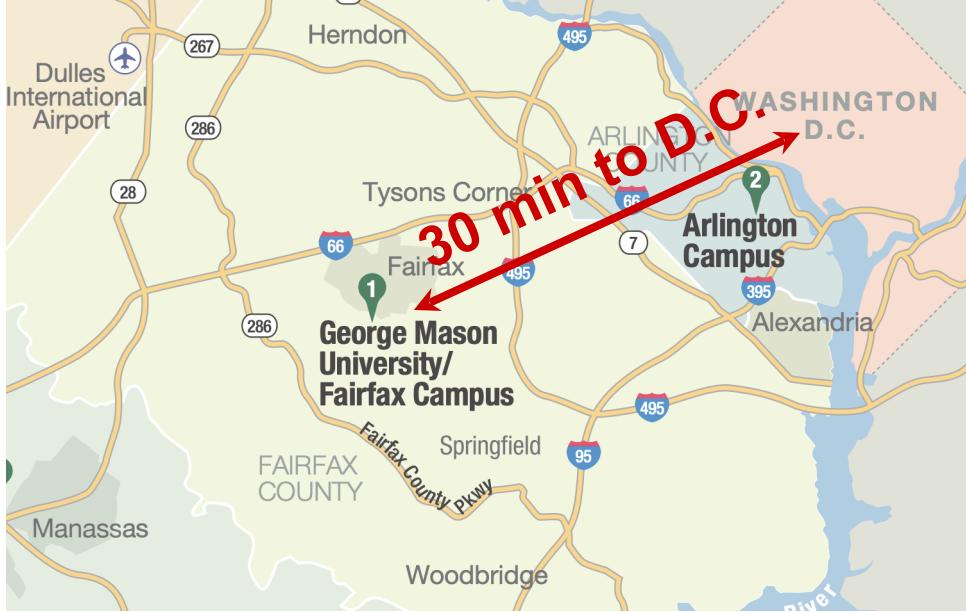


Jenn Suh  
(MathEdu)



# Thank You!

Email: [ziyuyao@gmu.edu](mailto:ziyuyao@gmu.edu)  
Webpage: <https://ziyuyao.org/>



APPLIED RESEARCH LABORATORY FOR  
INTELLIGENCE AND SECURITY



Commonwealth  
Cyber Initiative

