CS388: Natural Language Processing

Lecture 12: ICL 2:
Text rationales,
Chain-of-thought

Greg Durrett

TEXAS
The University of Texas at Austin

Administrivia

- Project 3 due in two weeks
- FP proposals back early next week

Recap: Zero-shot/Few-shot prompting

- Single unlabeled datapoint ***x***, want to predict label *y*

  x = *The movie's acting could've been better, but the visuals and directing were top-notch.*

- Wrap ***x*** in a template we call a verbalizer ***v***

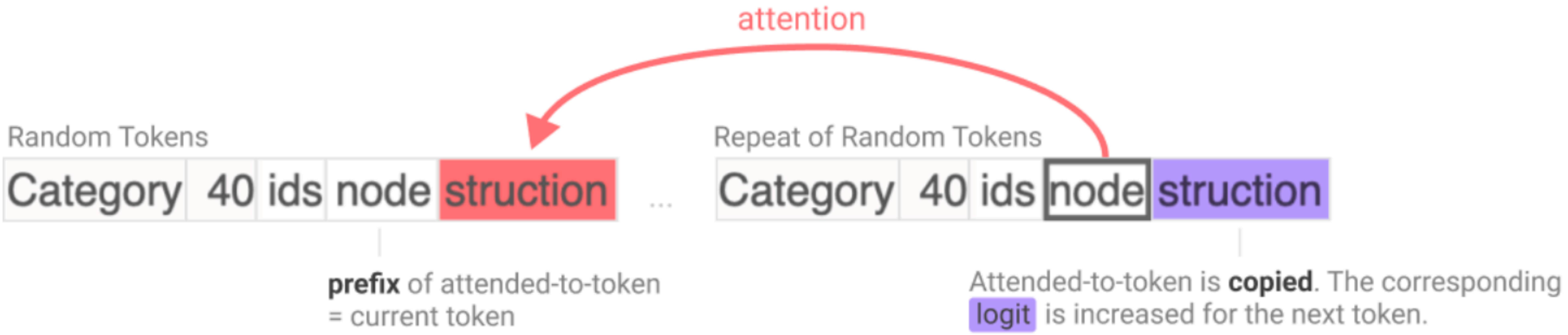
Review: The movie's acting could've been better, but the visuals and directing were top-notch.
Out of positive, negative, or neutral, this review is → GPT-3 → neutral

- Need the right prompt (but there is a "plateau" of prompts that work)

- Few-shot: add one or more examples. Typically works better! Particularly with rich examples like we'll see today

Recap: Understanding ICL

- ICL can learn a strategy like ordinary least-squares (Akyurek et al., 2022)
- We can identify *induction heads* in Transformers; these emerge when ICL performance improves (Olsson et al., 2022)



This Lecture

- Text rationales: text explanations of answers
- Chain-of-thought prompting (zero- and few-shot)
- Extensions
- Analysis of explanations

Text Rationales

Example from Vision

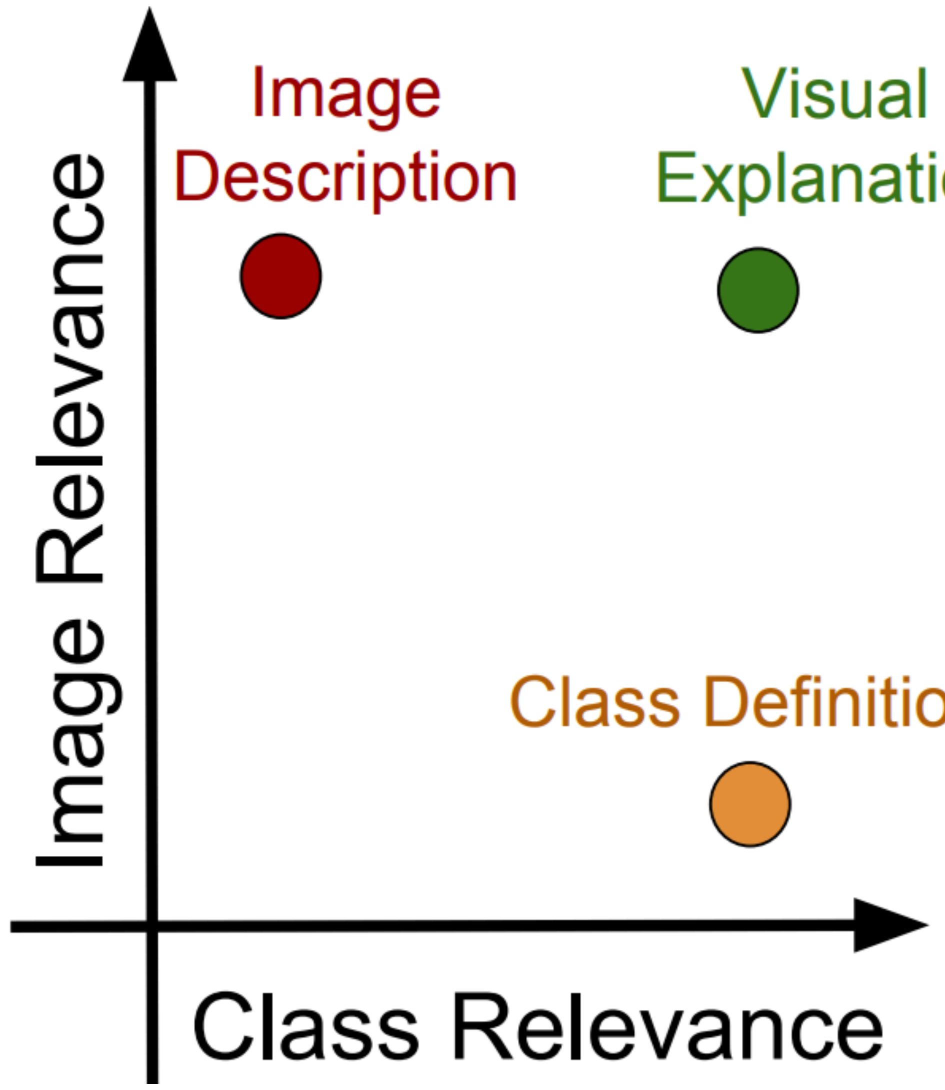
Laysan Albatross

Description: This is a large flying bird with black wings and a white belly.
Class Definition: The *Laysan Albatross* is a large seabird with a hooked yellow beak, black back and white belly.
Visual Explanation: This is a *Laysan Albatross* because this bird has a large wingspan, hooked yellow beak, and white belly.

Laysan Albatross

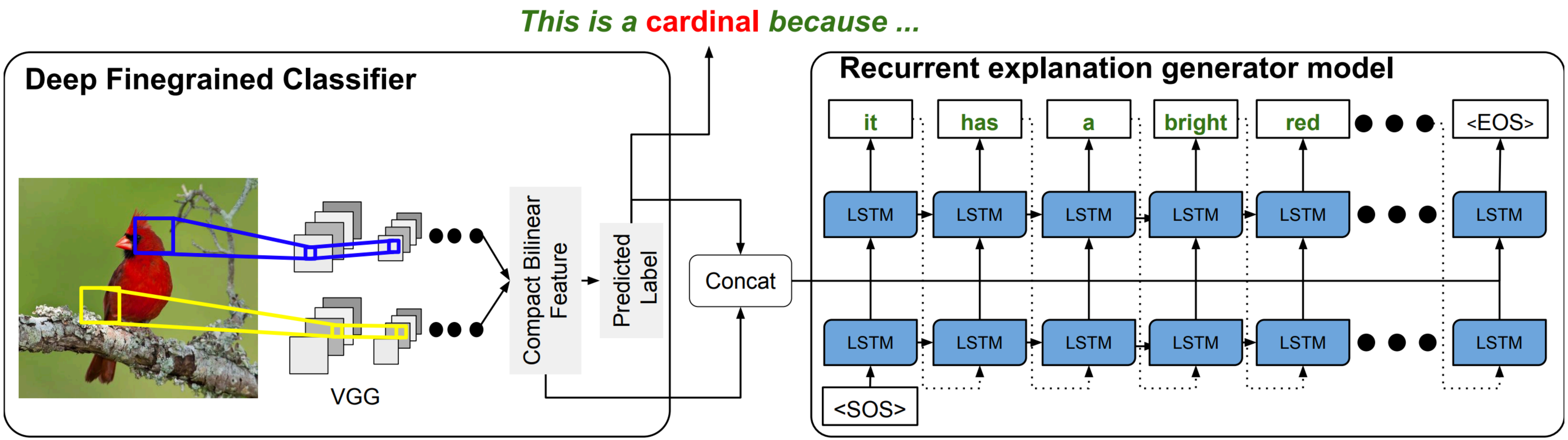
Description: This is a large bird with a white neck and a black back in the water.
Class Definition: The *Laysan Albatross* is a large seabird with a hooked yellow beak, black back and white belly.
Visual Explanation: This is a *Laysan Albatross* because this bird has a hooked yellow beak white neck and black back.

- What makes a visual explanation? Should be relevant to the class and the image
- Are these features *really* what the model used?



Hendricks et al. (2016)

Generating Explanations: Birds



- LSTM decoder looks at a feature vector and predicted label, then generates an explanation from those
- It's trained on human explanations — so it will likely produce explanations that look good (it learns to be a language model)

Hendricks et al. (2016)

E-SNLI

Premise: An adult dressed in black holds a stick.
Hypothesis: An adult is walking away, empty-handed.
Label: contradiction
Explanation: Holds a stick implies using hands so it is not empty-handed.

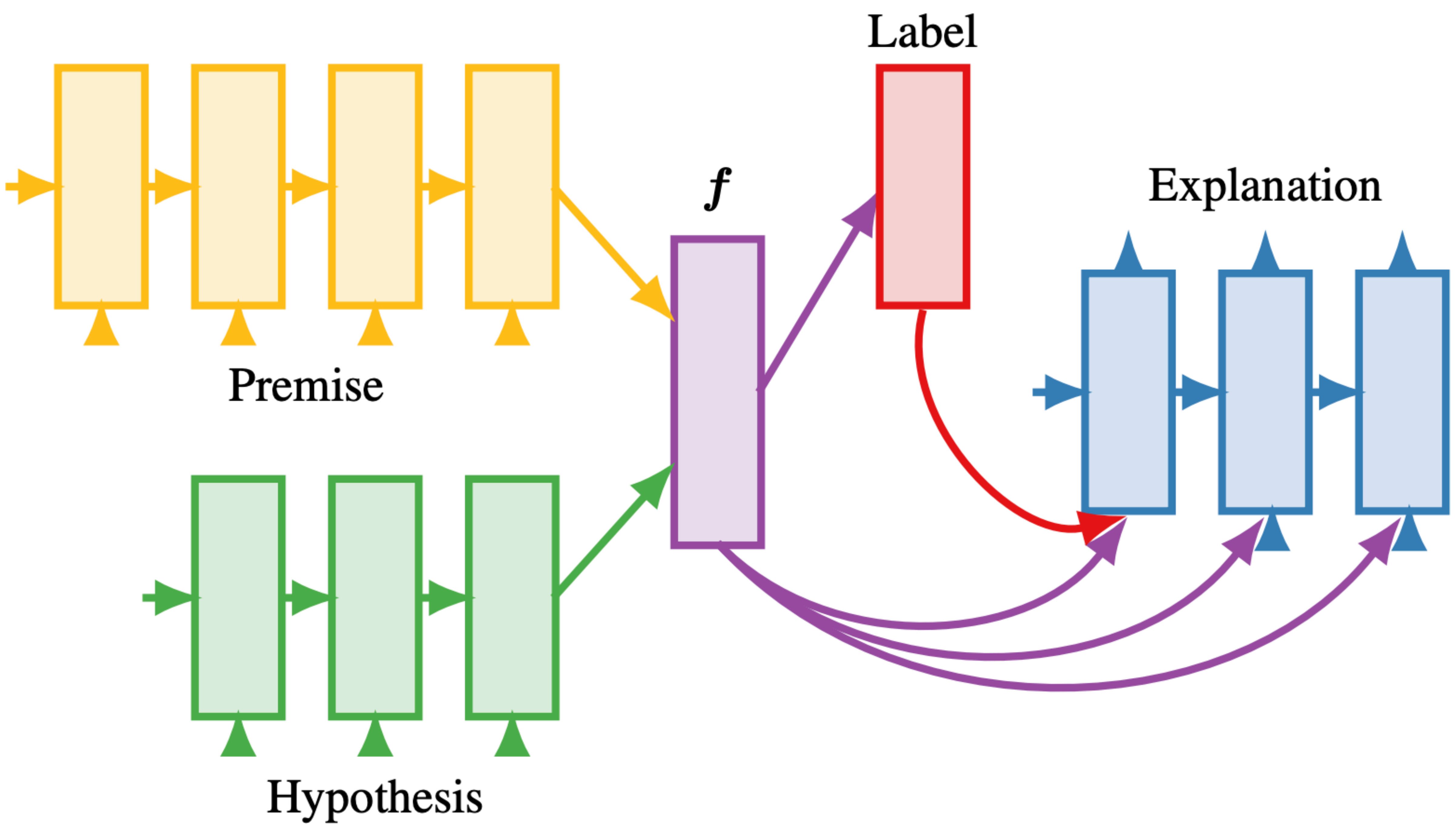
Premise: A child in a yellow plastic safety swing is laughing as a dark-haired woman in pink and coral pants stands behind her.
Hypothesis: A young mother is playing with her daughter in a swing.
Label: neutral
Explanation: Child does not imply daughter and woman does not imply mother.

Premise: A man in an orange vest leans over a pickup truck.
Hypothesis: A man is touching a truck.
Label: entailment
Explanation: Man leans over a pickup truck implies that he is touching it.

- Two formats: highlights and text

Camburu et al. (2019)

Generating Explanations: E-SNLI



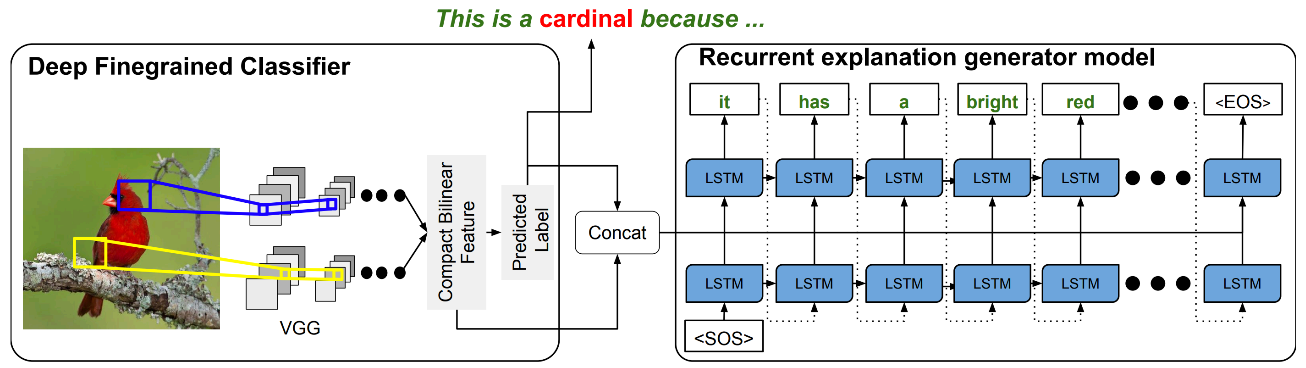
f = function of premise and hypothesis vectors

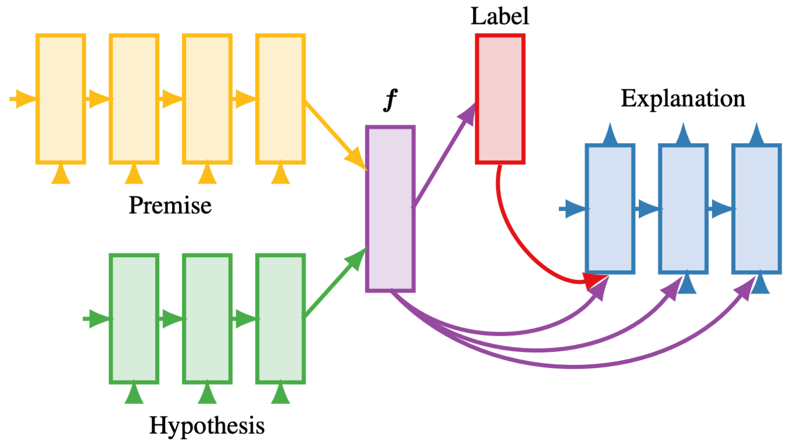
- Similar to birds: explanation is conditioned on the label + network state *f*
- Information from *f* is fed into the explanation LSTM, although we don't know how that information is being used

Camburu et al. (2019)

Text Explanations

- Can we generate a natural language explanation of a model's behavior?
- What are some advantages to this?
  - Easy for untrained users to understand
  - Multitasking to produce human-written explanations may help us learn
- What are some risks/disadvantages?





Text Explanations

- Issues with text explanations:
  - Hard to produce/consume (these models are sort of clunky)
  - Hard to know if they faithfully reflect what a model is doing
  - More broadly, hard to evaluate
- However, writing such explanations comes naturally to us…so that means that they reflect some kind of underlying reasoning process that we're doing?
- Pre-2021: this process would usually be captured structurally in a model. 2022 and beyond: chain of thought

Chain-of-thought

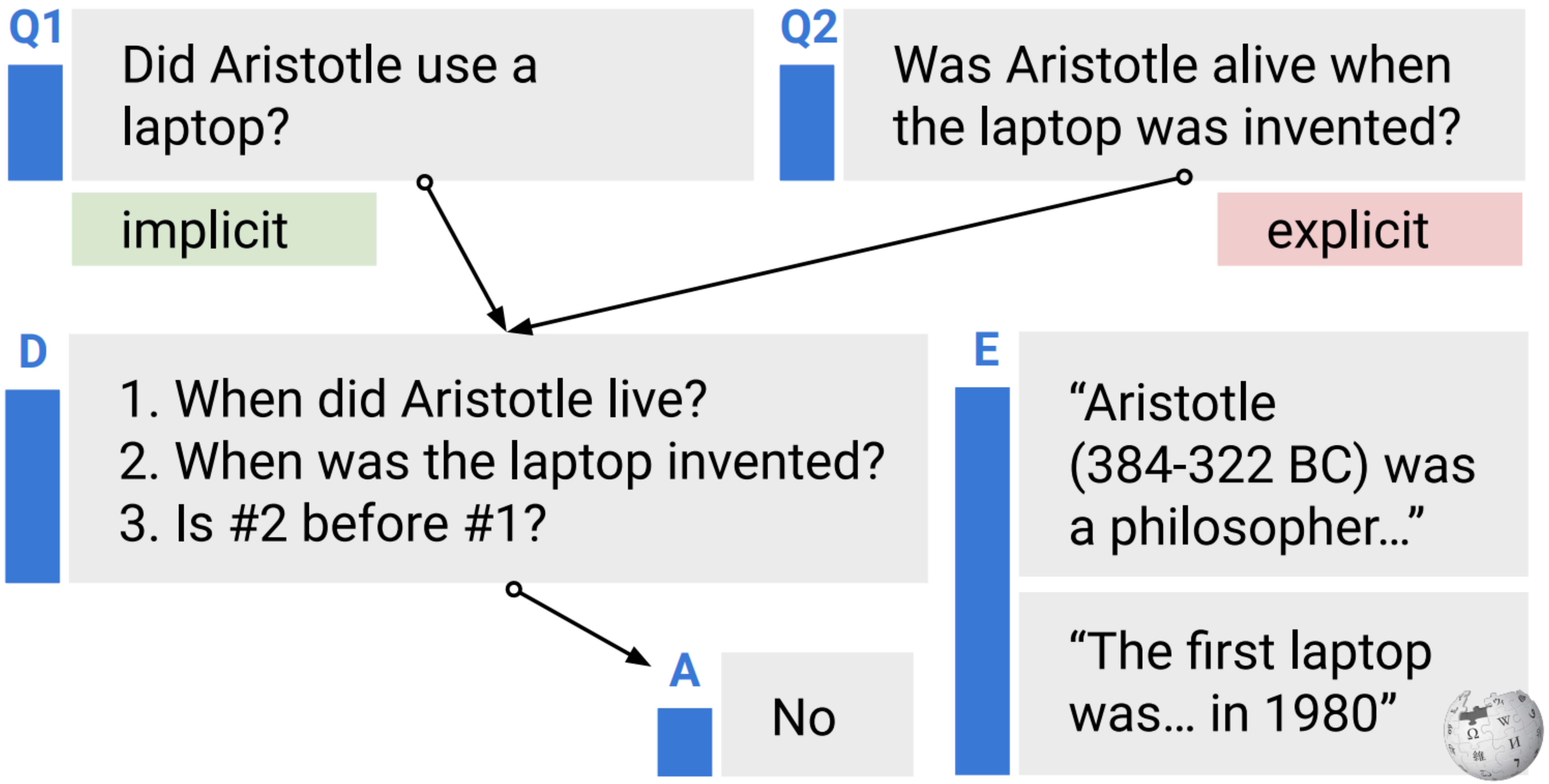
Text rationales vs. programs

Problem 2:
Question: From a pack of 52 cards, two cards are drawn together at random. What is the probability of both the cards being kings?
Options: A) 2/1223 B) 1/122 C) 1/221 D) 3/1253 E) 2/153
Rationale: Let s be the sample space.
Then n(s) = 52C2 = 1326
E = event of getting 2 kings out of 4
n(E) = 4C2 = 6
P(E) = 6/1326 = 1/221
Answer is C
Correct Option: C

- Rationales are most useful for problems where some computation is required. They can articulate the intermediate steps needed to solve it
- Some of the earliest work: math word problems

Ling et al. (2017)

QA



- "StrategyQA": dataset where different reasoning strategies are needed
- Related to multi-hop QA: "*What's the capital of the country where Aristotle lived?*" (but these are easy with current models)

Geva et al. (2021)

Chain-of-thought

- For these kinds of problems, do "computation" entirely in natural language
- Unifies several ideas:
  - For math: relies on the fact that LLMs can at least do single steps of arithmetic okay
  - For QA: many problems involve reasoning decompositions
    E.g., *What's the capital of the country where Aristotle lived?* ->
    ans = "*country where Aristotle lived*"
    return *What's the capital of* [ans]
  - For other tasks: capture the kinds of behavior written in rationales

Wei et al. (2022)

Chain-of-thought

- Typically a few-shot prompting technique where the in-context examples now contain explanations
- Answer is not generated in one go, but comes after an explanation that "talks through" the reasoning

Input:

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

...

Q: John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. How many hours a week does he spend taking care of dogs?
A:

Model output:

John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. So that is 10 x .5 = 5 hours a day. 5 hours a day x 7 days a week = 35 hours a week. The answer is 35 hours a week. ✓

Wei et al. (2022)

Chain-of-thought

From our work: a synthetic test of multi-hop reasoning with extractive explanations:

Context: *Christopher agrees with Kevin. Tiffany agrees with Matthew. Mary hangs out with Danielle. James hangs out with Thomas. Kevin is a student. Matthew is a plumber. Danielle is a student. Thomas is a plumber.*

Q: *Who hangs out with a student?*

A: *Mary.*

- What kind of explanation would you write here?

Explanation: *because Mary hangs out with Danielle and Danielle is a student.*

Ye and Durrett (NeurIPS 2022)

Chain-of-thought

Context: *Christopher agrees with Kevin.* [...] **Q**: *Who hangs out with a student?*

Mary

Standard few-shot learning, no explanation

Context: *Christopher agrees with Kevin.* [...] **Q**: *Who hangs out with a student?*

Mary, ***because Mary hangs out with Danielle and Danielle is a student.***

Predict-explain: answer **is not** conditioned on output explanation (original E-SNLI LSTM)

Context: *Christopher agrees with Kevin.* [...] **Q**: *Who hangs out with a student?*

Because Mary hangs out with Danielle and Danielle is a student, *the answer is* ***Mary.***

Explain-predict: answer is conditioned on output explanation (Chain of Thought)

Chain-of-thought

Prompt

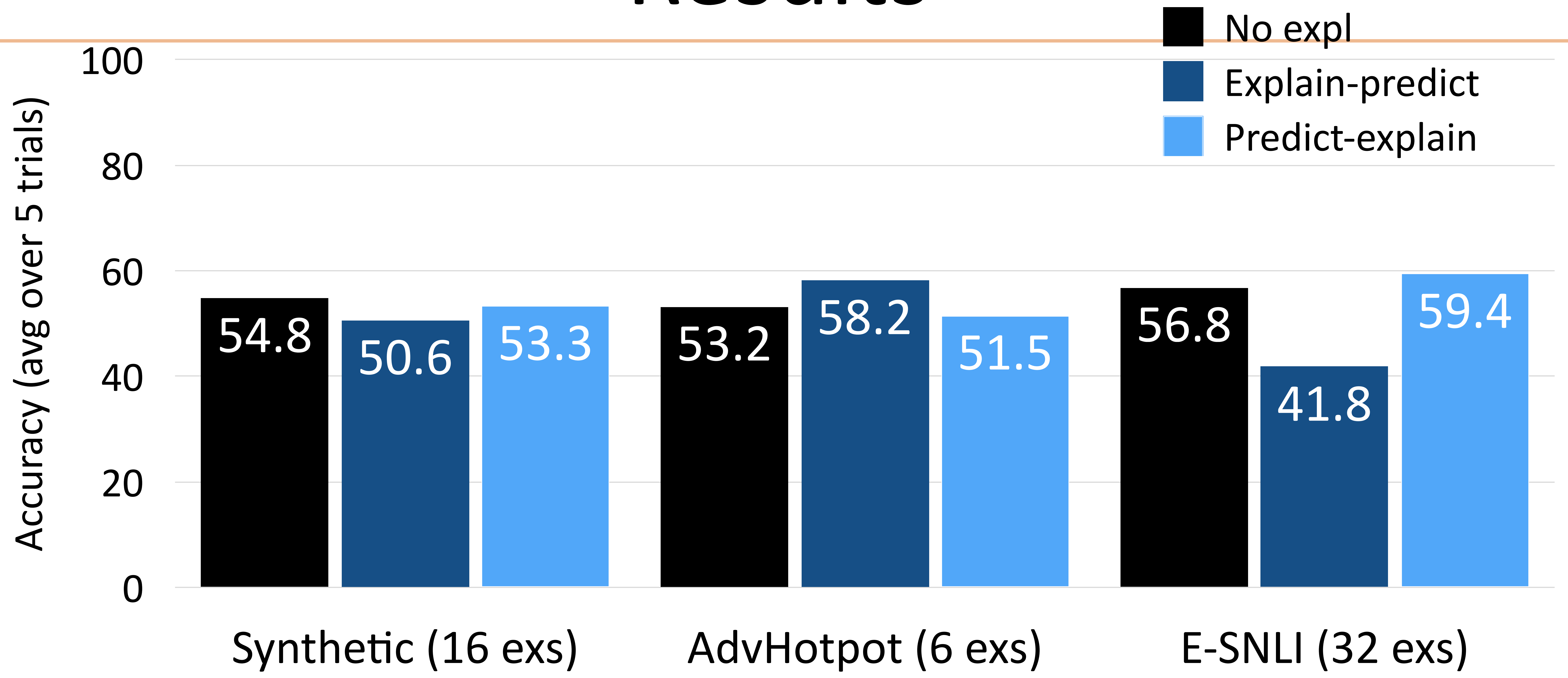
Input	**Context**: *Christopher agrees with Kevin.* [...] **Q**: *Who hangs out with a student?*
Label+ Explanation	***Mary**, **because Mary hangs out with Danielle and Danielle is a student.***
Train Ex	
Train Ex	
Test Input	**Context**: *Adam plays with Ellen.* [...] **Q**: *Who plays with a doctor?*

GPT-3

Output	*Adam*, ***because Adam plays with Ellen and Ellen is a doctor.***

greedy decoding from GPT-3

Results



Does GPT-3 (text-davinci-001) work well without explanations?

- **Not well.** On Synthetic, surface heuristics give 50%.

Q1: Do these explanations help?

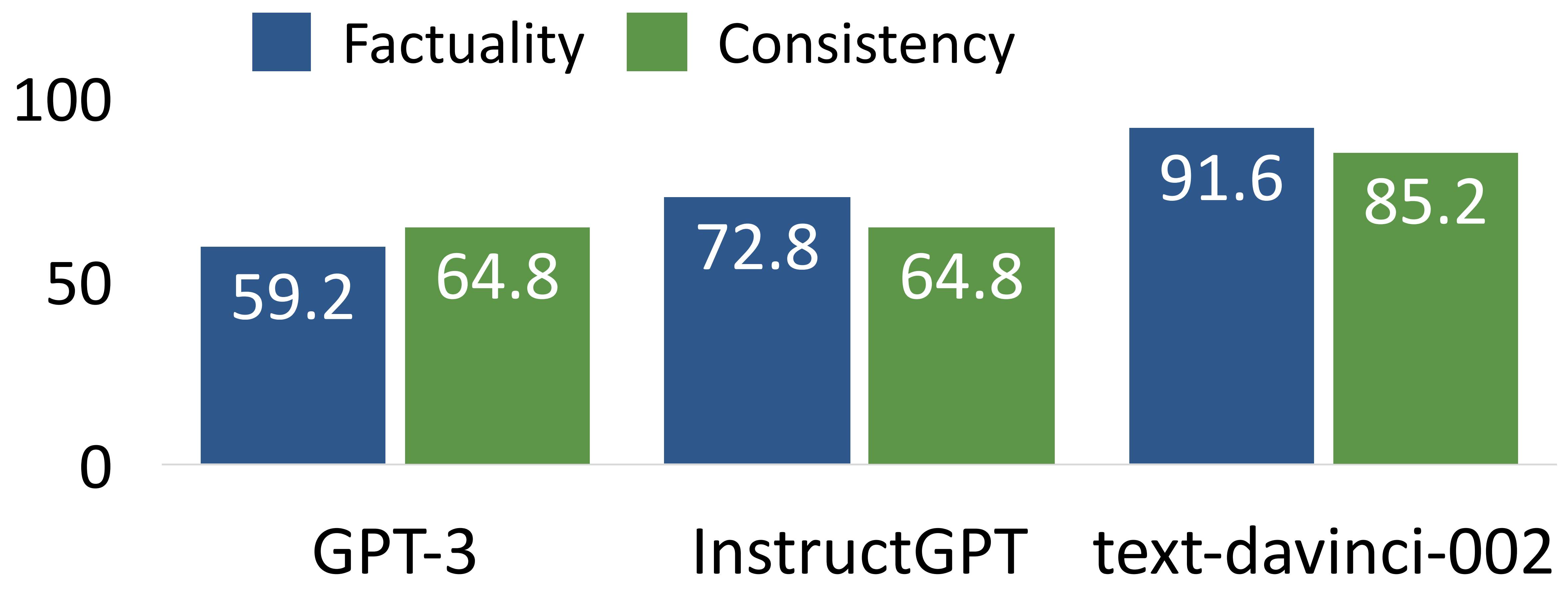
- **Not really.** Small gains on AdvHotpot and E-SNLI. No one technique dominates

Ye and Durrett (NeurIPS 2022)

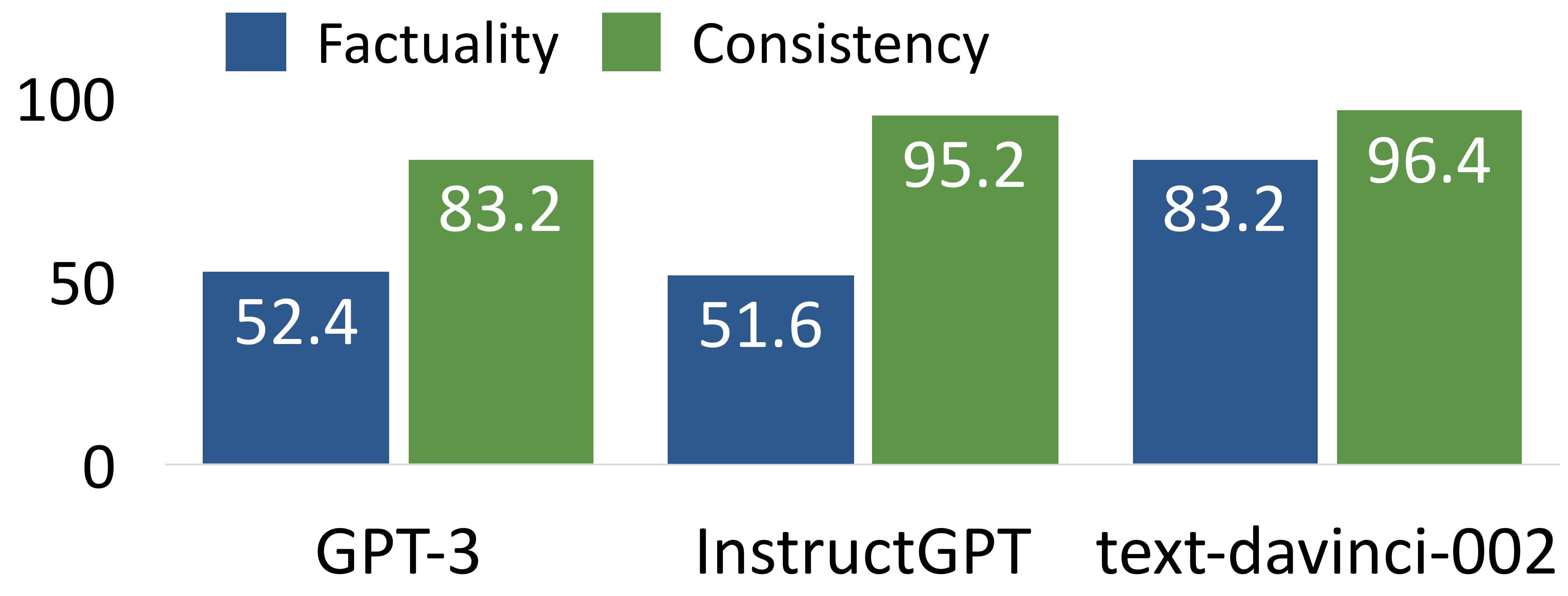
Results

- Can language models generate reliable explanations?
  - **Factuality:** whether an explanation is factually grounded in the input context
  - **Consistency:** whether an explanation entails the answer
- Model-generated explanations are not always **reliable**

Explain-Predict on SYNTH

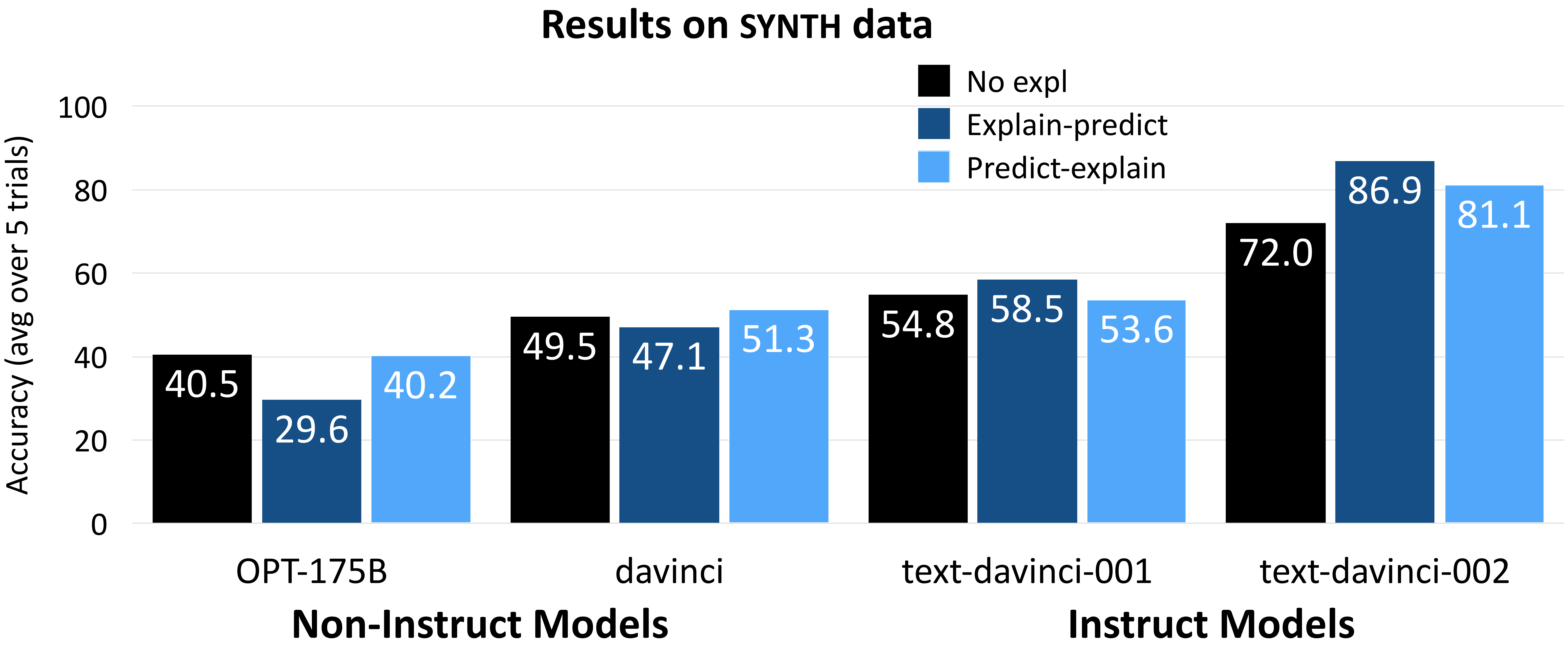


Predict-Explain on SYNTH



Ye and Durrett (NeurIPS 2022)

Results



- Instruct tuning helps but it seems to be not quite sufficient
- **Bigger models are better, and modern models are very good**

Ye and Durrett (NeurIPS 2022)

What Makes Explanations Effective?

- Do LMs “follow” explanations?
- Probing LLMs with perturbed explanations
  - Perturbing **Computation Trace**
  - Perturbing **Natural Language**

Question

Take the last letters of the words in "Bill Gates” and concatenate them.

Gold Explanation

Trace **NL**

The last letter of "Bill" is letter"l". The last of "Gates" is "s". Concatenating “l" and "s" is “ls". So the answer is ls.

Perturbing Trace

The last letter of "Bill" is letter " ". The last of "Gates" is " ". Concatenating “l" and "s" is “ls". So the answer is ls.

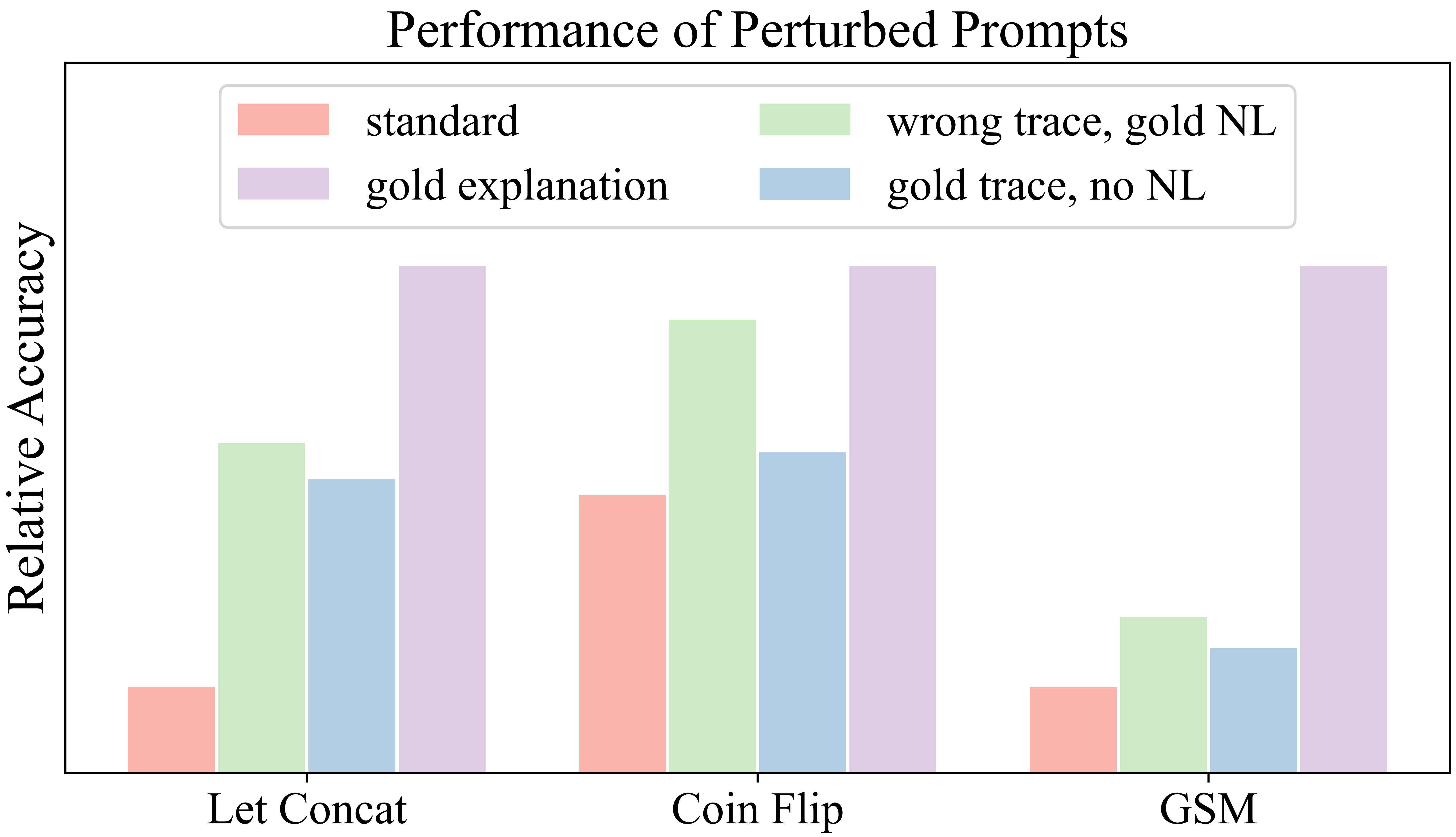
Perturbing NL

"Bill","l","Gates","s","l","s","ls". So the answer is ls.

Ye et al. (2022)

What Makes Explanations Effective?

- Do LMs “follow” explanations? How do explanations work for in-context-learning?
  - YES. Perturbing either trace or NL leads to performance degradation.
  - But perturbed explanations are still beneficial compared to not using explanations at all



Ye et al. (2022)

Chain-of-thought extensions

Step-by-Step

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls.* ✓

- Prompt for step-by-step reasoning: produces chains of thought without including demonstrations
- Separate prompt to extract the answer ("Therefore, the answer is ___")

Kojima et al. (2022)

Step-by-Step

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

- text-davinci-002 (fine-tuned model)

Kojima et al. (2022)

Step-by-Step

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

Kojima et al. (2022)

Chain-of-thought

You

From a pack of 52 cards, two cards are drawn together at random. What is the probability of both cards being kings?

ChatGPT

To calculate the probability of both cards drawn from a pack of 52 cards being kings, we can use the formula for the probability of A and B happening, which is $P(A \text{ and } B) = \frac{\text{Number of ways A and B can happen}}{\text{Total number of possible outcomes}}$.

There are 4 kings in a deck of 52 cards. So, the number of ways to draw the first king is 4 (since there are 4 kings available out of 52 cards). After drawing the first king, there are 3 kings left out of the remaining 51 cards.

Thus, the probability of drawing one king and then another king is calculated as follows:

$$P(\text{first card is a king and second card is a king}) = \frac{4}{52} \times \frac{3}{51}$$

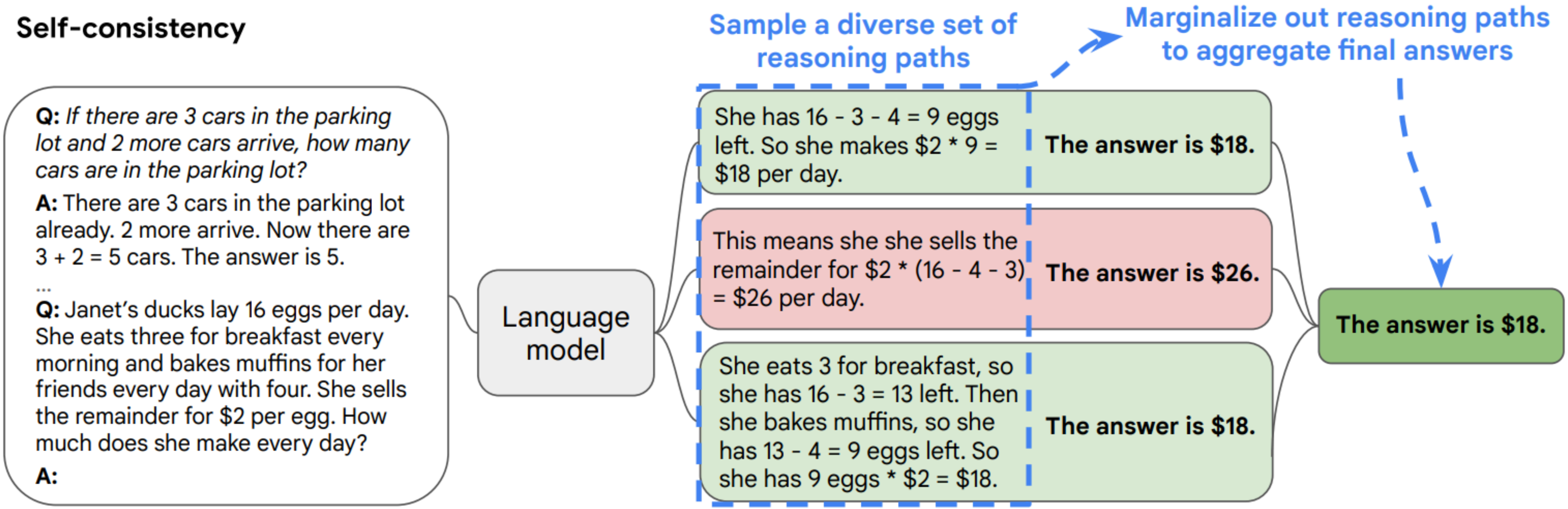
Let's calculate this probability.

The probability of both cards drawn from a pack of 52 cards being kings is approximately 0.0045, or about 0.45%.

- **Models have been RLHFed to do this without prompting**

Demo: Step-by-Step
(Math QA, StrategyQA)

Self-Consistency



- Ensembling across multiple outputs (either zero-shot or few-shot)
- GSM8k: 56.5 -> 74.4, 5% gains on several other math datasets, lower gains on text tasks

Wang et al. (2022)

Program-aided Language Models

- For math: why are we doing the arithmetic in the LLM itself?

- Instead: generate code fragments and actually execute them to get an answer (how most earlier math word problem systems worked)

- Many flavors of this: “Faithful Chain-of-thought”, “Program-of-thought”, Toolformer, etc.

Model Output

A: The bakers started with 200 loaves
```
loaves_baked = 200
```
They sold 93 in the morning and 39 in the afternoon
```
loaves_sold_morning = 93
loaves_sold_afternoon = 39
```
The grocery store returned 6 loaves.
```
loaves_returned = 6
```
The answer is
```
answer = loaves_baked - loaves_sold_morning
  - loaves_sold_afternoon + loaves_returned
```

```
>>> print(answer)
74
```

Gao et al. (2022)

Self-ask

- Similar idea but with QA/a search engine in the loop
- Demonstration shows sub-questions and sub-answers, can potentially do search at these intermediate points
- Bing's "Sydney" agent has some capabilities around this

GPT-3

Question: Who lived longer, Theodor Haecker or Harry Vaughan Watkins?
Are follow up questions needed here: Yes.
Follow up: How old was Theodor Haecker when he died?
Intermediate answer: Theodor Haecker was 65 years old when he died.
Follow up: How old was Harry Vaughan Watkins when he died?
Intermediate answer: Harry Vaughan Watkins was 69 years old when he died.
So the final answer is: Harry Vaughan Watkins

Question: Who was president of the U.S. when superconductivity was discovered?
Are follow up questions needed here: Yes.
Follow up: When was superconductivity discovered?
Intermediate answer: Superconductivity was discovered in 1911.
Follow up: Who was president of the U.S. in 1911?
Intermediate answer: William Howard Taft.
So the final answer is: William Howard Taft.

Press et al. (2022)

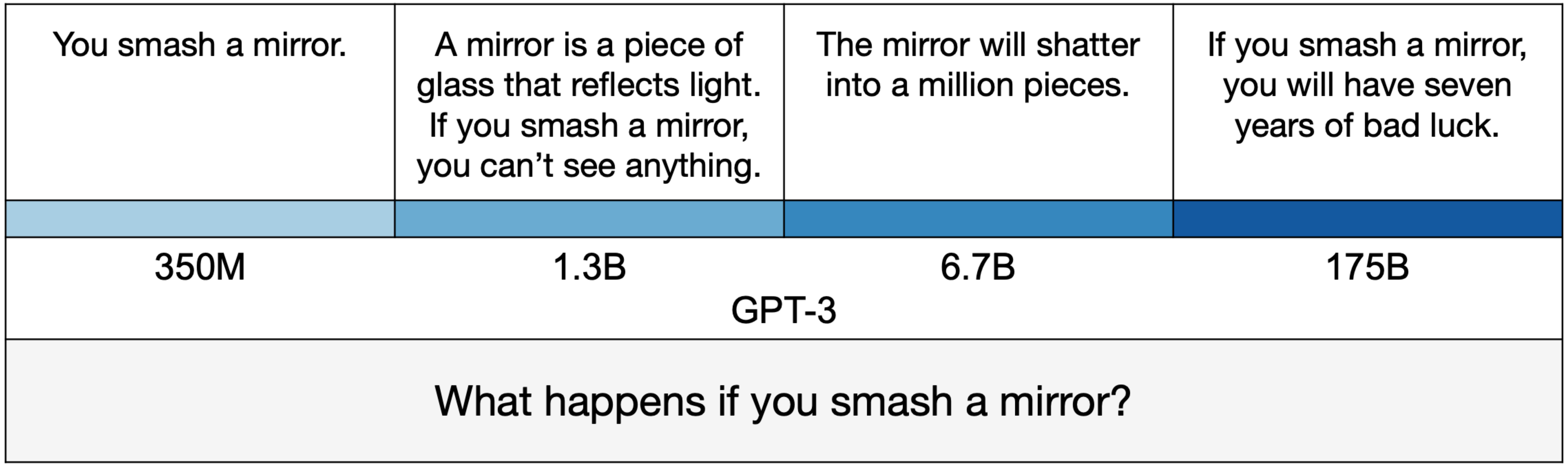
Other ideas

- For math: can having various other ways of doing programmatic verification
- For natural language reasoning: missing component of search and planning, discussed in “Language Model Cascades”
- For problems like fact-checking or QA involving complex reasoning, its difficult to verify all of the individual steps…so if CoT goes wrong, it may even be hard for a human to spot

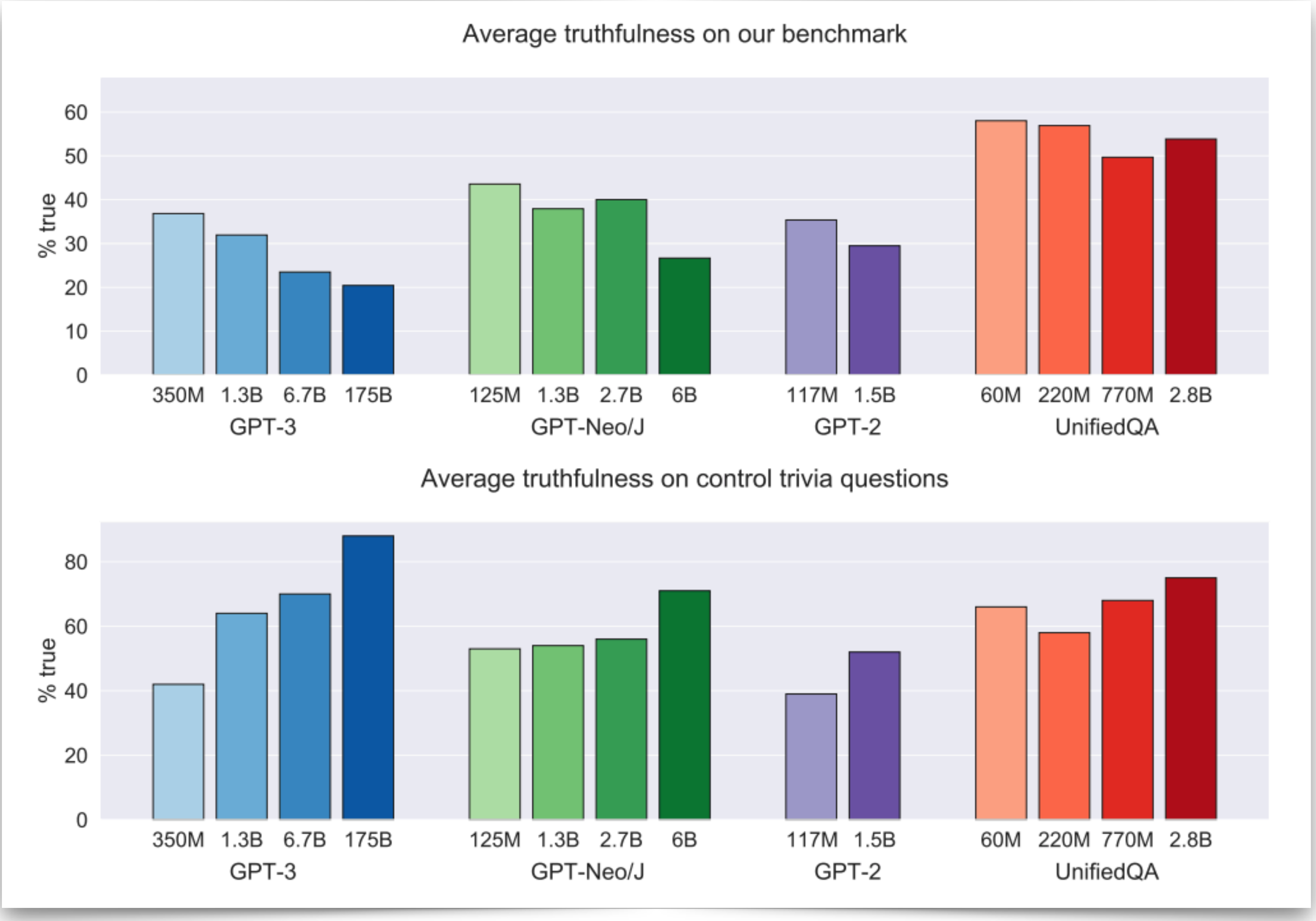
Factuality and Hallucination

Factuality

- Language models model distributions over text, not facts. There's no guarantee that what they generate is factual:
  - Language models are trained on the web. Widely-popularized falsehoods may be reproduced in language models
  - A language model may not be able to store all rare facts, and as a result moderate probability is assigned to several options



TruthfulQA



Average truthfulness on our benchmark
% true
0
10
20
30
40
50
60
350M 1.3B 6.7B 175B
GPT-3
125M 1.3B 2.7B 6B
GPT-Neo/J
117M 1.5B
GPT-2
60M 220M 770M 2.8B
UnifiedQA
Average truthfulness on control trivia questions
% true
0
20
40
60
80
350M 1.3B 6.7B 175B
GPT-3
125M 1.3B 2.7B 6B
GPT-Neo/J
117M 1.5B
GPT-2
60M 220M 770M 2.8B
UnifiedQA

Factuality

- Language models model distributions over text, not facts. There's no guarantee that what they generate is factual:
  - Language models are trained on the web. Widely-popularized falsehoods may be reproduced in language models
- A language model may not be able to store all rare facts, and as a result moderate probability is assigned to several options
- There are many proposed solutions to factuality. How do we evaluate them? How do we check facts "explicitly"?

Grounding LM Generations

- Suppose we have text generated from an LM. We want to check it against a source document. What techniques have we seen so far that can do this?

- What steps are involved?

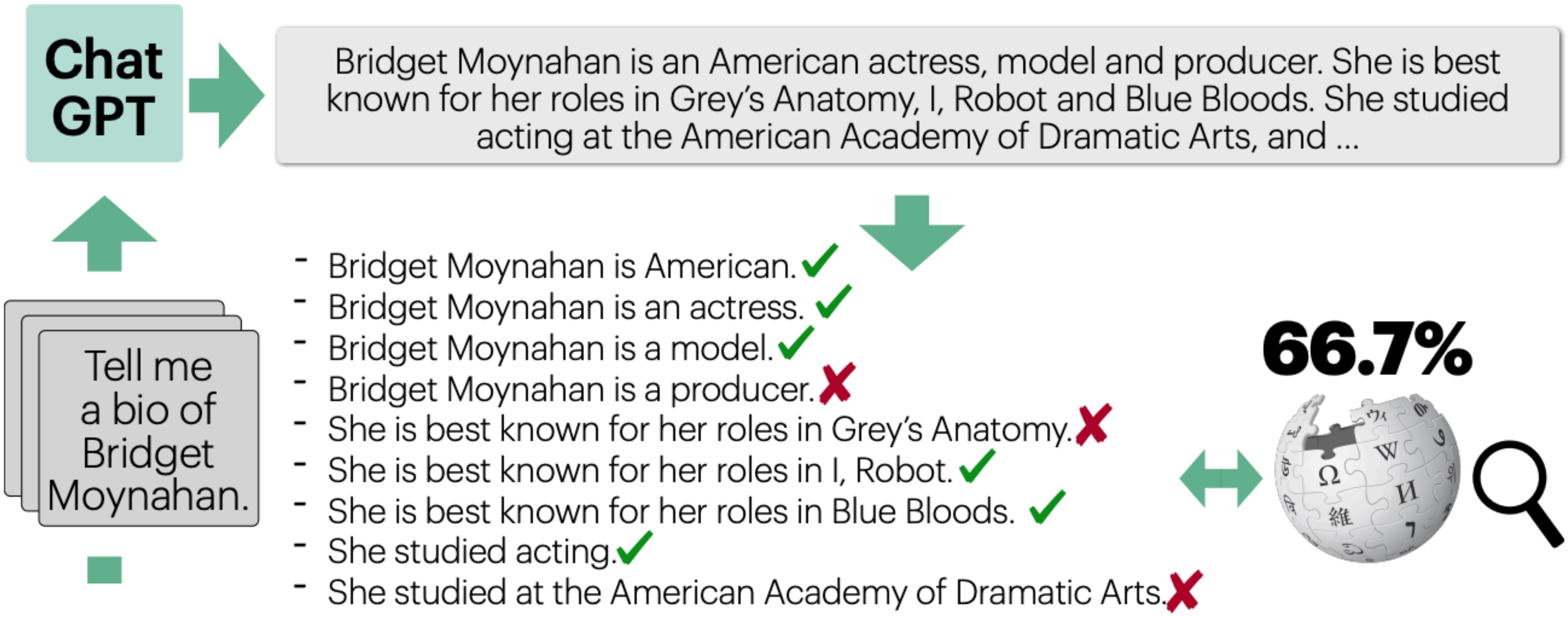
  1. Decide what text you are grounding in (may involve retrieval)

  2. Decompose your text into pieces of meaning to ground

  3. Check each piece

- For now, we'll assume the reference text/documents are given to us and not focus on step 1

Concrete Setting

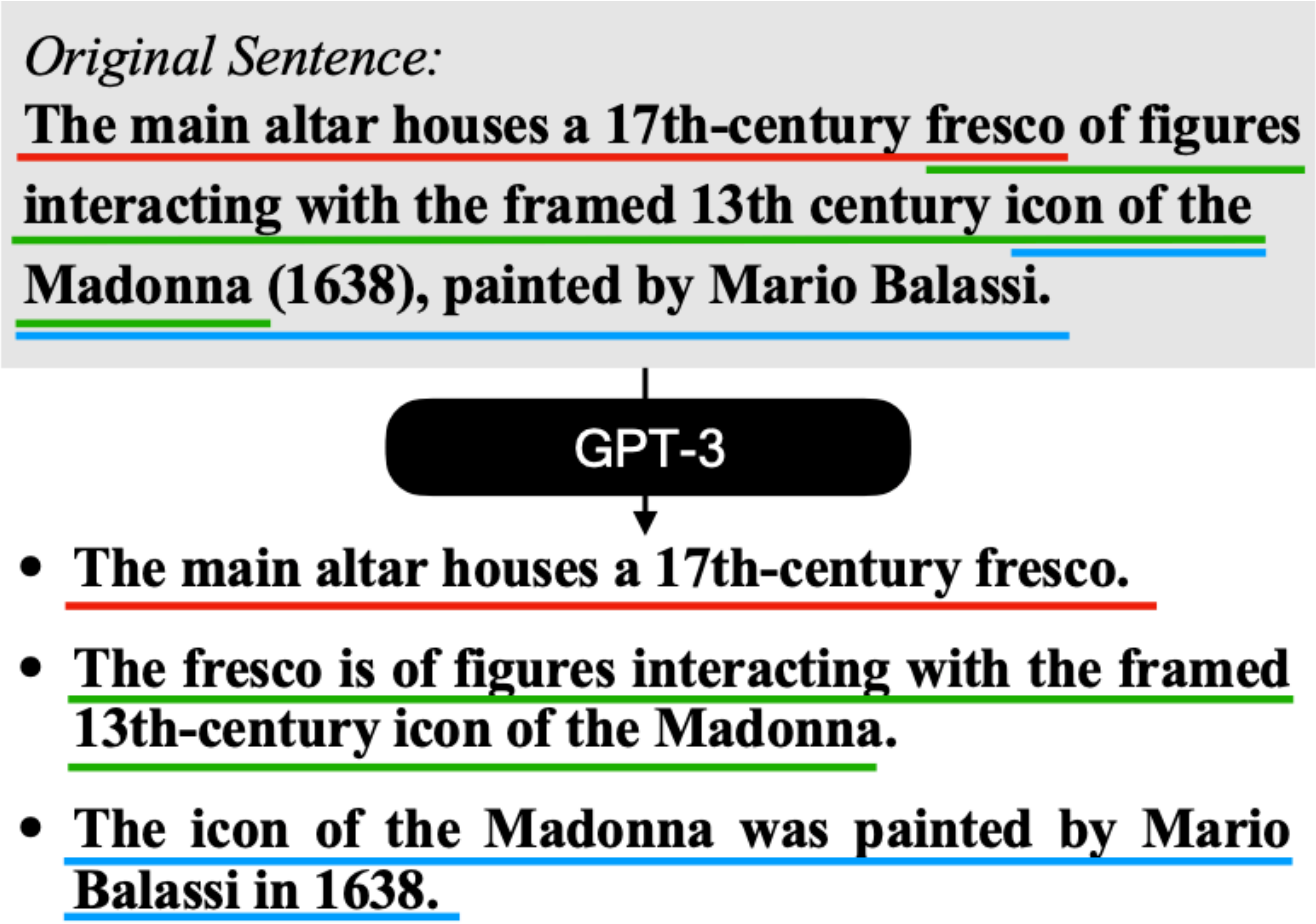


- Dataset: ChatGPT-generated biographies of people. May contain errors, particularly when dealing with obscure people!

Sewon Min and Kalpesh Krishna et al. (2023)

Step 2: Decomposition

- Simplest approach: each sentence needs to be grounded
- Can go deeper: think of sentences as expressing a collection of propositions
- Long history in frame semantics of defining these propositions. Many propositions anchor to verbs

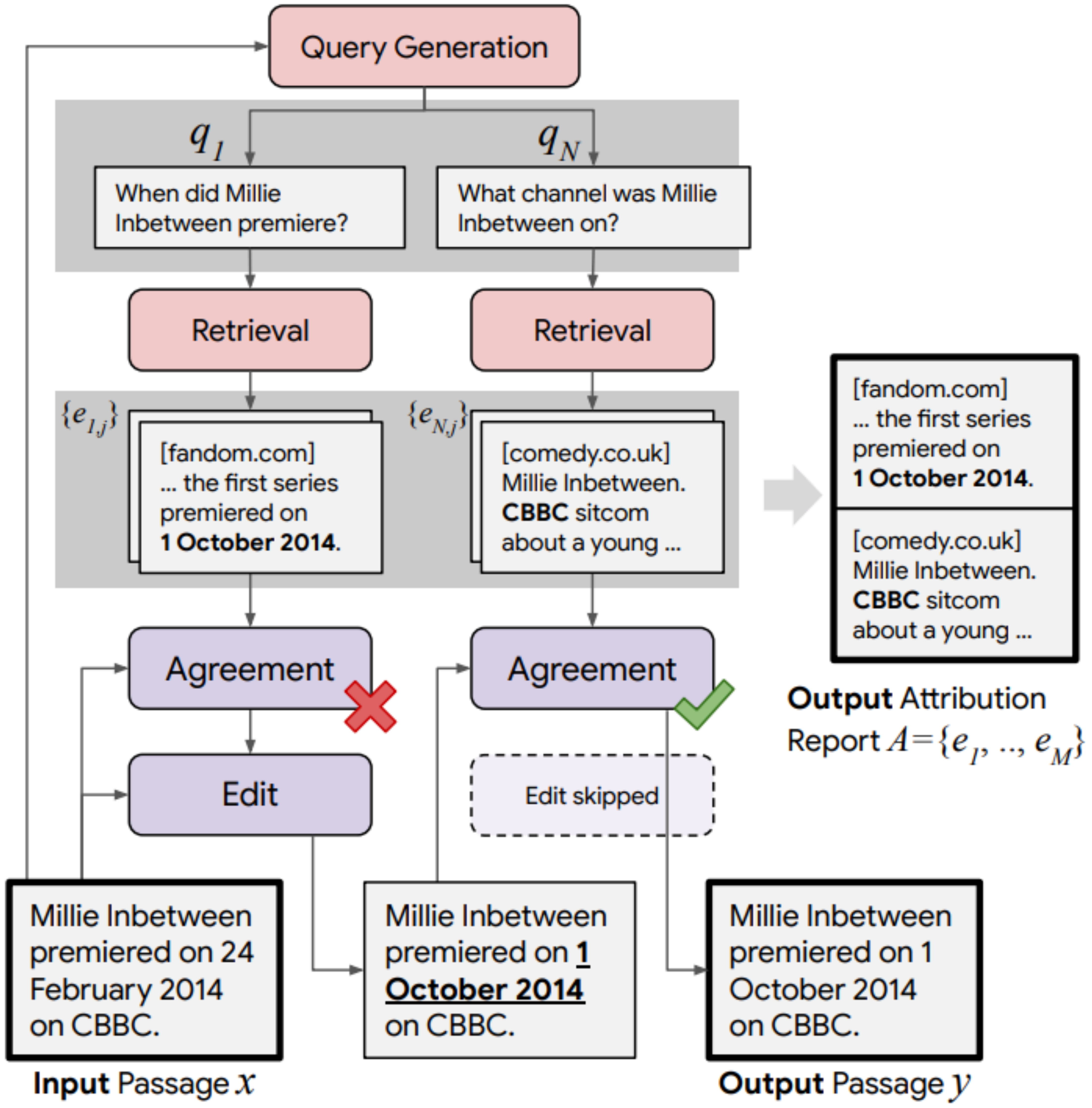


- Recent work: extract propositions with LLMs

Yixin Liu et al. (2023)

Ryo Kamoi et al. (2023)

Pipeline: RARR



- Full pipeline including retrieval
- Decomposition is framed as question generation
- The "checking" stage is also implemented with LLMs here
- Final stage: try to revise the output

Luyu Gao et al. (2022)

Takeaways

- Chain-of-thought prompting (zero- and few-shot) can work well for tasks involving reasoning, especially mathematical reasoning and textual question answering with multiple steps

- Several things needed to improve them, such as self-consistency and the ability to use other resources like code execution or APIs

- Next time: RLHF, makes models better at zero-shot prompting and producing well-structured chain-of-thought responses