Ask Me Anything

A Simple Strategy for Prompting Language Models

Arora, Narayan, Chen, Orr, Guha, Bhatia, Chami, Sala and Ré,
International Conference for Learning Representations
May 2023

Center for Research on Foundation Models

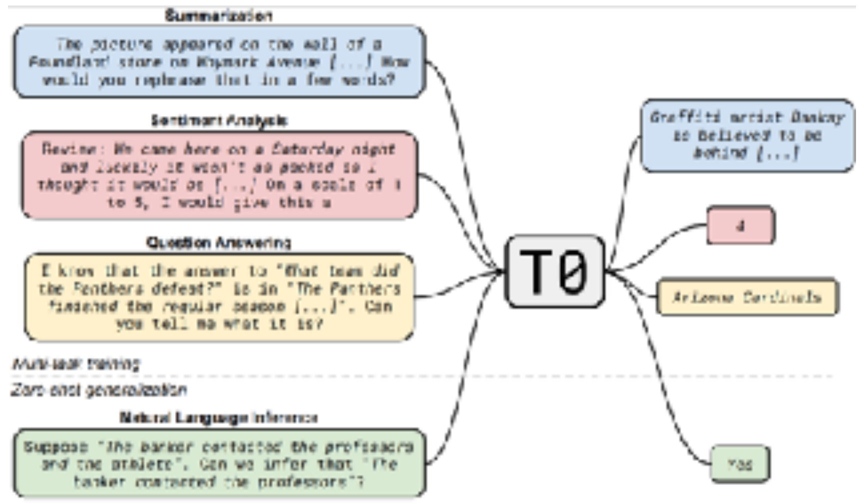
Ask Me Anything (AMA) enables:

1. An open-source 6B model to outperform OpenAI's 175B parameter GPT-3 model on **15 tasks** used in the original GPT-3 paper!

2. A **10.2 ± 6.1% absolute (21.4 ± 11.2% relative)** performance improvement over the few-shot baseline in evaluations on **14 unique language models** spanning **5 orders of magnitude in model size** (125M - 176B) and **four families** of models:

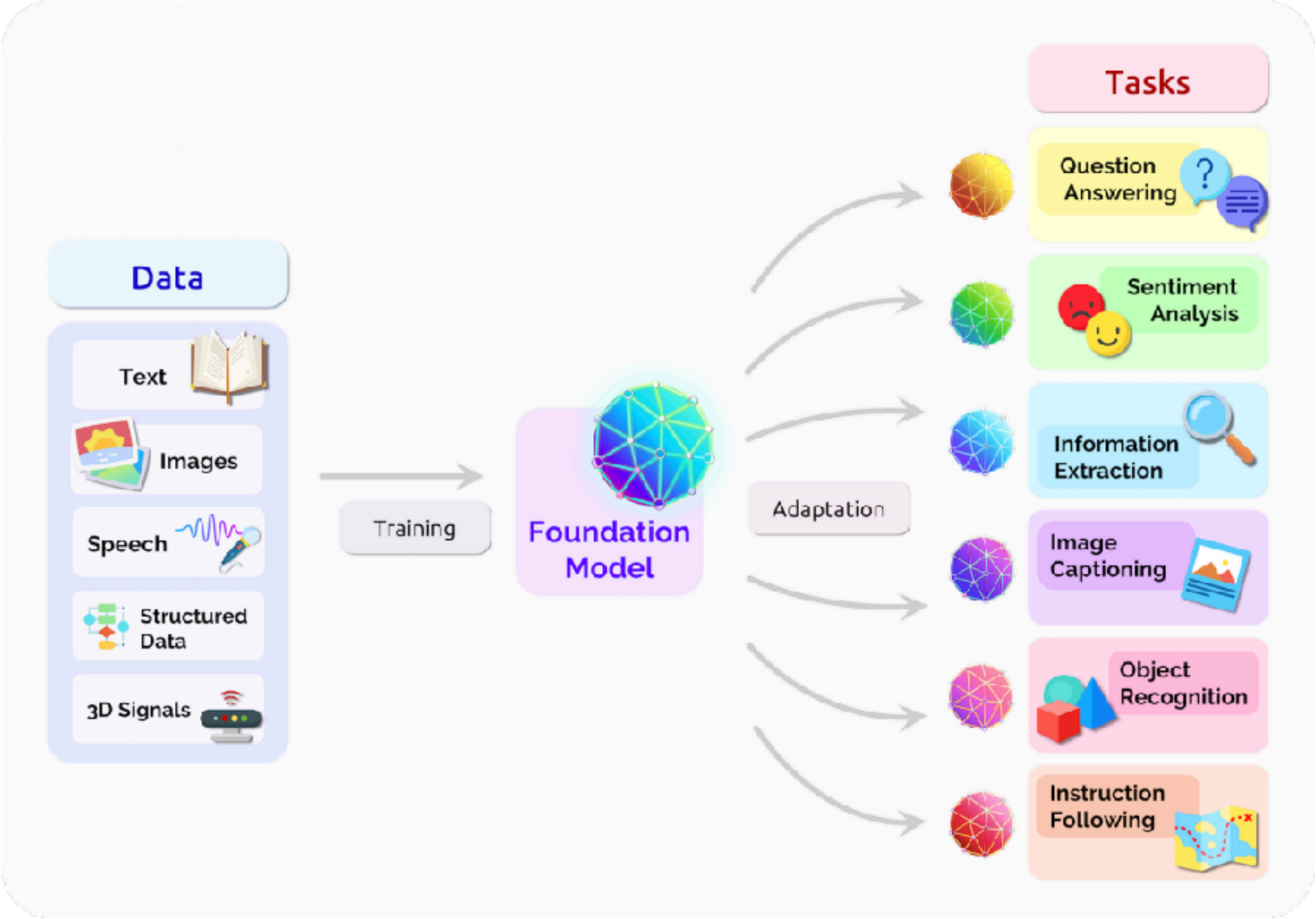
a BigScience initiative

176B params · 59 languages · Open-access



Emergent properties of recent language models

Language models are models trained on **broad data** (generally using self-supervision at scale) that can be adapted to a **wide range of downstream tasks**. [1]

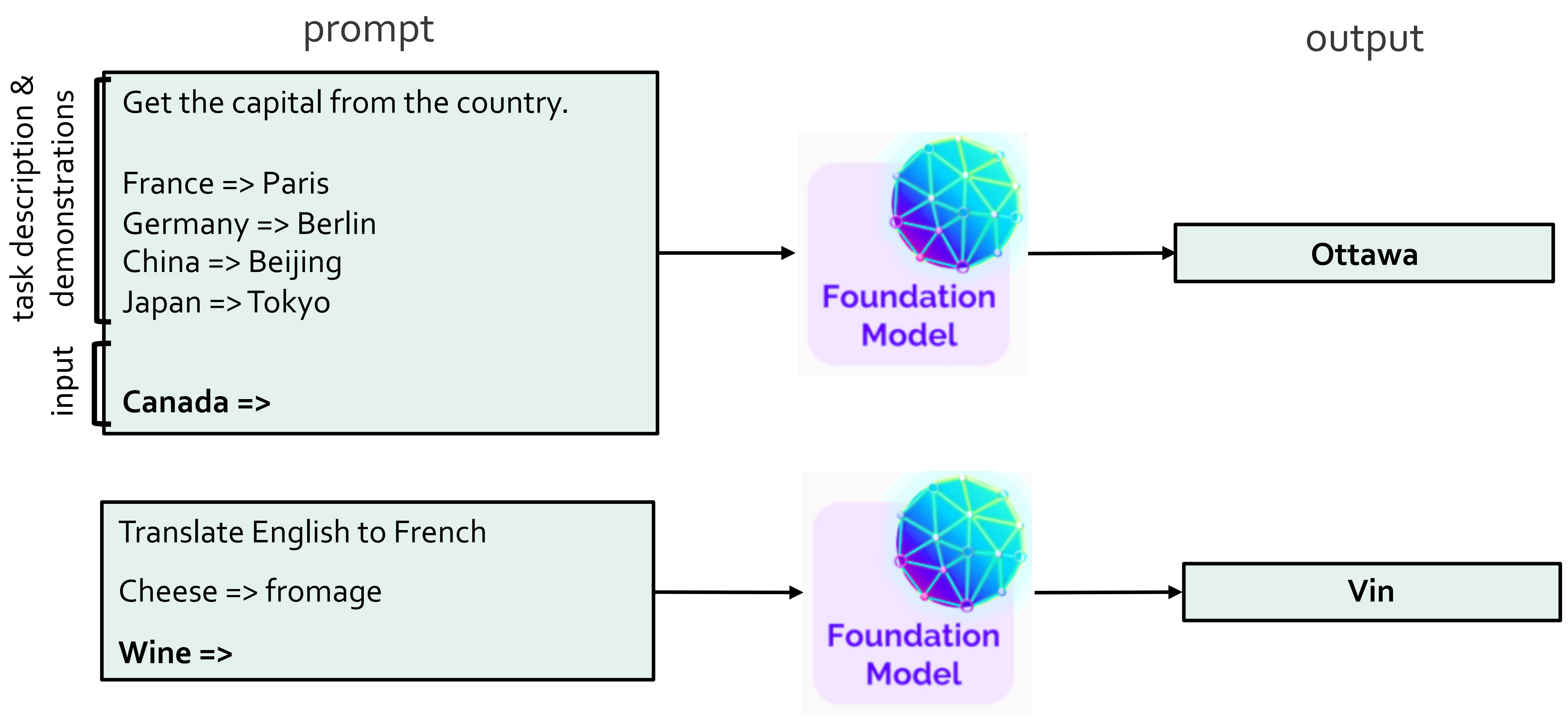


[1] Bommasani, Hudson, Altman, **Arora**, von Arx, Bernstein, Bohg, Bosselut, Brunskill et al., On the Opportunities and Risks of Foundation Models. 2021.

How can we use recent language models?

Prior: full-model and parameter-efficient fine-tuning, with one model per task

Recent models display *in-context learning* abilities: they can be controlled by prompts, to support many task types and languages with *no* additional training

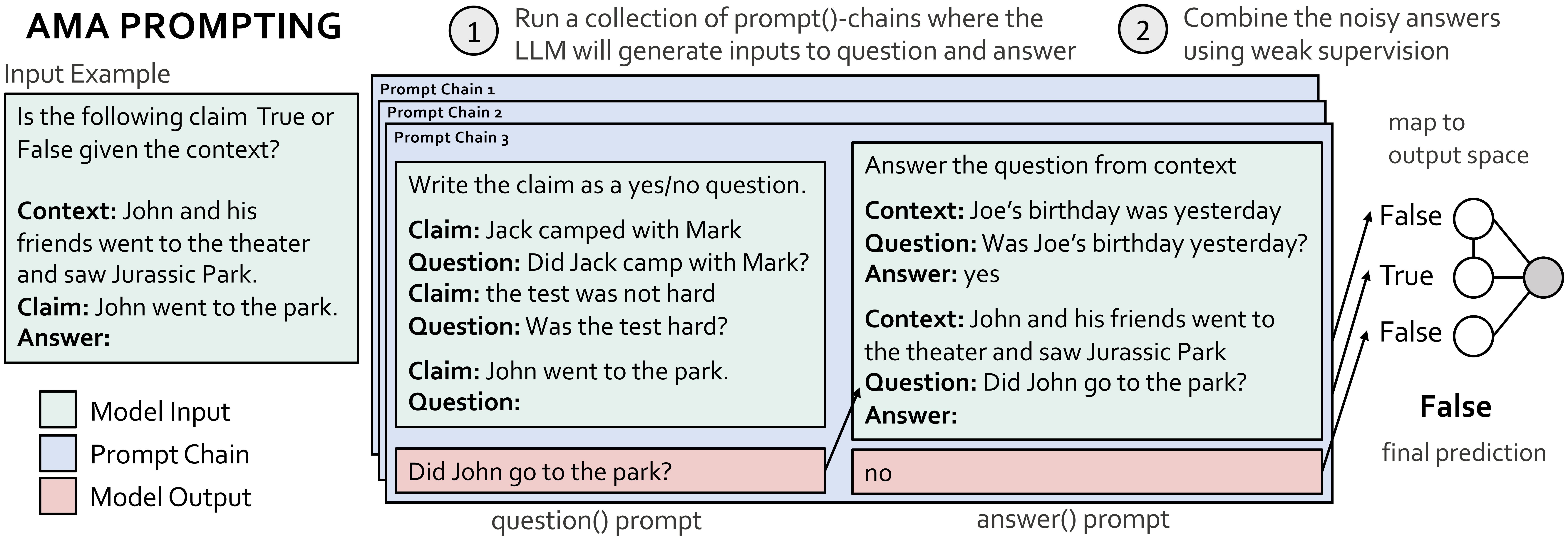


In-context Learning is Amazing!

Photo Credit Dalle-2. “An Astronaut Riding a Horse in a Photo-Realistic Style”

- We (ML and non-ML experts) can express our goals to models in **natural language!**
- We can build apps in **hours** that would have taken **years**!
- Learned representations **reduce the manual engineering effort** to capture *many variations* in machine learning pipelines.

Ask Me Anything (AMA)



AMA aggregates multiple decent, yet ultimately noisy prompts using weak-supervision to surpass OpenAI's few-shot 175B parameter GPT-3 on 15 popular benchmark tasks with an open-source 6B parameter model!

Ask Me Anything (AMA)

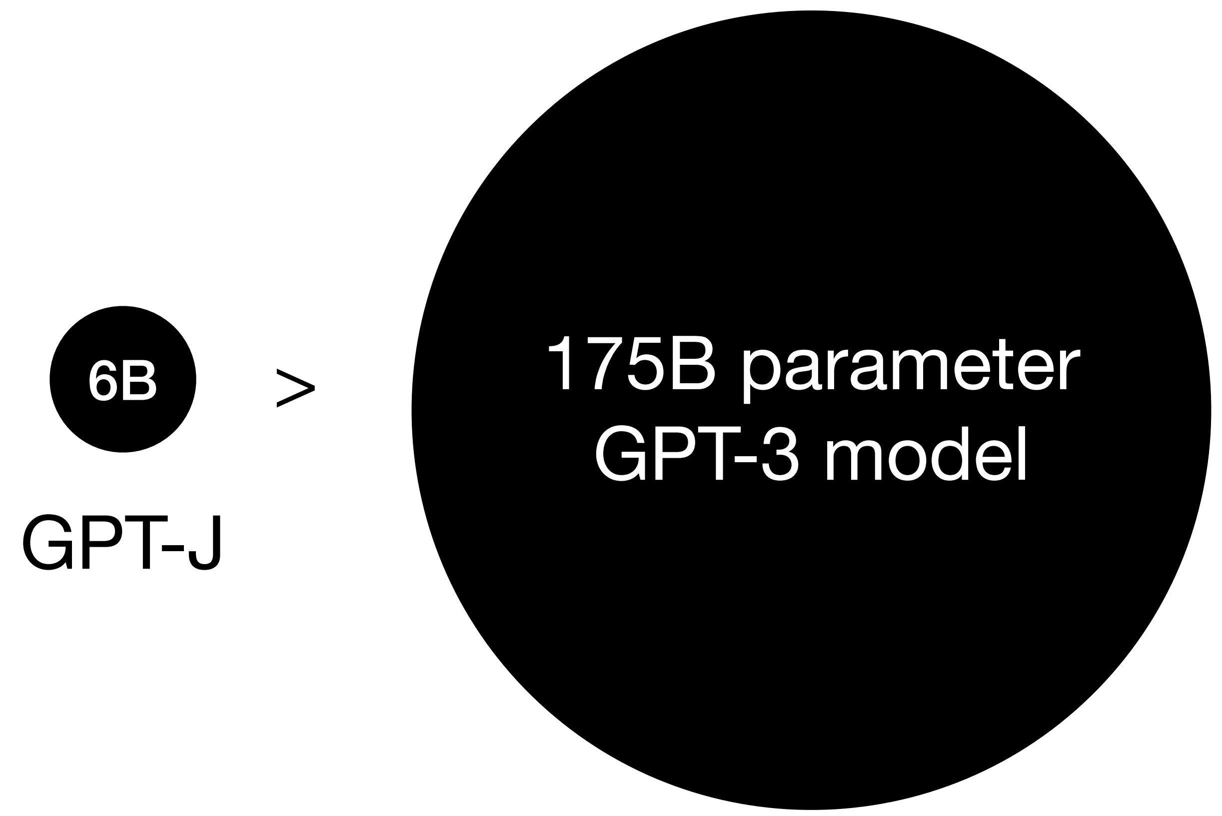
Three key questions. Across tasks and language models:

1. How do we get prompts that are of decent-quality? We need to **understand** properties of *effective* prompts.
2. How do we **generate** those effective prompts *efficiently* at scale?
3. How do we **aggregate** the predictions generated by the different prompts *reliably*?

We Beat GPT-3 on their Benchmarks!

Model	Neo Few-Shot	Neo (QA)	Neo (QA + WS)	GPT-3 Few-Shot
# Params	6B	6B	6B	175B
Natural Language Understanding				
BoolQ	$66.5_{(3)}$	64.9	$67.2_{\pm 0.0}$	$\mathbf{77.5}_{(\mathbf{32})}$
CB	$25.0_{(3)}$	83.3	$\mathbf{83.9}_{\pm 0.0}$	$82.1_{(32)}$
COPA	$77.0_{(3)}$	58.2	$84.0_{\pm 0.0}$	$\mathbf{92.0}_{(\mathbf{32})}$
MultiRC	$60.8_{(3)}$	58.8	$63.8_{\pm 0.0}$	$\mathbf{74.8}_{(\mathbf{32})}$
ReCoRD	$75.6_{(3)}$	74.5	$74.4_{\pm 0.0}$	$\mathbf{89.0}_{(\mathbf{32})}$
RTE	$58.8_{(3)}$	61.7	$\mathbf{75.1}_{\pm 0.0}$	$72.9_{(32)}$
WSC	$36.5_{(3)}$	74.7	$\mathbf{77.9}_{\pm 0.0}$	$75.0_{(32)}$
WiC	$53.3_{(3)}$	59.0	$\mathbf{61.0}_{\pm 0.2}$	$55.3_{(32)}$
Natural Language Inference				
ANLI R1	$32.3_{(3)}$	34.6	$\mathbf{37.8}_{\pm 0.2}$	$36.8_{(50)}$
ANLI R2	$33.5_{(3)}$	35.4	$\mathbf{37.9}_{\pm 0.2}$	$34.0_{(50)}$
ANLI R3	$33.8_{(3)}$	37.0	$\mathbf{40.9}_{\pm 0.5}$	$40.2_{(50)}$
StoryCloze	$51.0_{(3)}$	76.3	$\mathbf{87.8}_{\pm 0.0}$	$87.7_{(70)}$
Classification				
AGNews	$74.5_{(3)}$	83.7	$\mathbf{86.4}_{\pm 0.0}$	$79.1_{(8)}$
Amazon	$62.5_{(3)}$	66.8	$\mathbf{68.2}_{\pm 0.0}$	$41.9_{(8)}$
DBPedia	$50.7_{(3)}$	81.4	$\mathbf{83.9}_{\pm 0.0}$	$83.2_{(8)}$
SST	$64.9_{(3)}$	94.5	$\mathbf{95.7}_{\pm 0.0}$	$95.6_{(8)}$
Question-Answering				
DROP	$32.3_{(3)}$	51.0	$\mathbf{51.6}_{\pm 0.0}$	$36.5_{(20)}$
NQ	$13.7_{(3)}$	19.7	$19.6_{\pm 0.0}$	$\mathbf{29.9}_{(64)}$
RealTimeQA	$34.7_{(3)}$	34.7	$\mathbf{36.0}_{\pm 0.0}$	$35.4_{(1)}$
WebQs	$29.1_{(3)}$	44.2	$\mathbf{44.1}_{\pm 0.0}$	$41.5_{(64)}$

With an open-source model that's 1/30th the size!

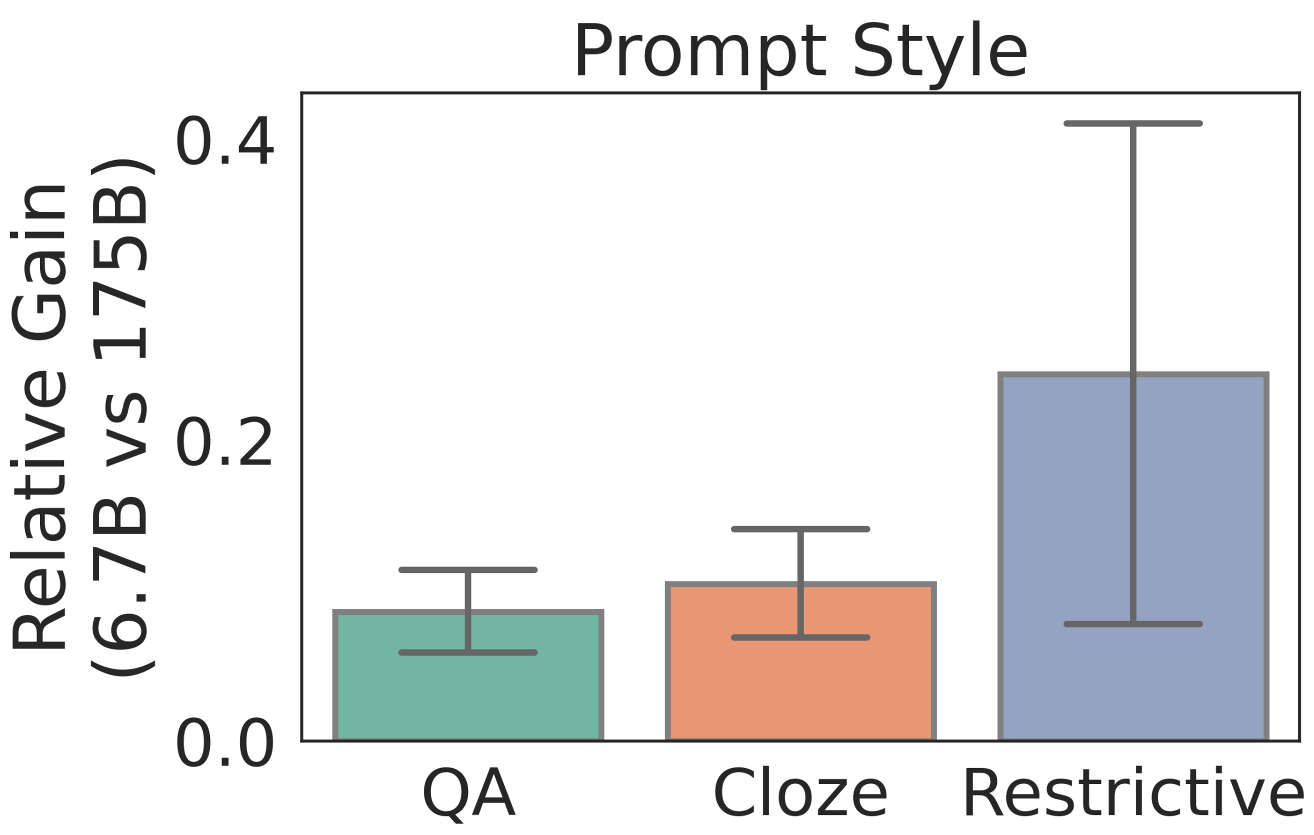


What makes for an effective prompt?

Open-ended questions make for an effective prompt format.

We group the results in the GPT-3 paper by the prompt-format used for the task.

Scaling from GPT3-6.7B to 175B, the relative gain is far lower on open-ended formats vs. restricted formats:



Cloze	Yes/No QA	*Wh-* QA
Context: John and his friends went to the theater and saw Jurassic Park. **Claim**: John went to the ___ **Answer**:	**Context**: John and his friends went to the theater and saw Jurassic Park. **Question**: Did John go to the park? **Answer**:	**Context**: John and his friends went to the theater and saw Jurassic Park. **Question**: Where did John go? **Answer**:

Open-ended question formats

Context: John and his friends went to the theater and saw Jurassic Park.
Claim: John went to the park. True or False?
Answer:

Restrictive

Measuring the effect of prompt-reformatting

Model	Neo Few-Shot	Neo (QA)	Neo (QA + WS)	GPT-3 Few-Shot
# Params	6B	6B	6B	175B
Natural Language Understanding				
BoolQ	$66.5_{(3)}$	64.9	$67.2_{\pm 0.0}$	$\mathbf{77.5}_{(32)}$
CB	$25.0_{(3)}$	83.3	$\mathbf{83.9}_{\pm 0.0}$	$82.1_{(32)}$
COPA	$77.0_{(3)}$	58.2	$84.0_{\pm 0.0}$	$\mathbf{92.0}_{(32)}$
MultiRC	$60.8_{(3)}$	58.8	$63.8_{\pm 0.0}$	$\mathbf{74.8}_{(32)}$
ReCoRD	$75.6_{(3)}$	74.5	$74.4_{\pm 0.0}$	$\mathbf{89.0}_{(32)}$
RTE	$58.8_{(3)}$	61.7	$\mathbf{75.1}_{\pm 0.0}$	$72.9_{(32)}$
WSC	$36.5_{(3)}$	74.7	$\mathbf{77.9}_{\pm 0.0}$	$75.0_{(32)}$
WiC	$53.3_{(3)}$	59.0	$\mathbf{61.0}_{\pm 0.2}$	$55.3_{(32)}$
Natural Language Inference				
ANLI R1	$32.3_{(3)}$	34.6	$\mathbf{37.8}_{\pm 0.2}$	$36.8_{(50)}$
ANLI R2	$33.5_{(3)}$	35.4	$\mathbf{37.9}_{\pm 0.2}$	$34.0_{(50)}$
ANLI R3	$33.8_{(3)}$	37.0	$\mathbf{40.9}_{\pm 0.5}$	$40.2_{(50)}$
StoryCloze	$51.0_{(3)}$	76.3	$\mathbf{87.8}_{\pm 0.0}$	$87.7_{(70)}$
Classification				
AGNews	$74.5_{(3)}$	83.7	$\mathbf{86.4}_{\pm 0.0}$	$79.1_{(8)}$
Amazon	$62.5_{(3)}$	66.8	$\mathbf{68.2}_{\pm 0.0}$	$41.9_{(8)}$
DBPedia	$50.7_{(3)}$	81.4	$\mathbf{83.9}_{\pm 0.0}$	$83.2_{(8)}$
SST	$64.9_{(3)}$	94.5	$\mathbf{95.7}_{\pm 0.0}$	$95.6_{(8)}$
Question-Answering				
DROP	$32.3_{(3)}$	51.0	$\mathbf{51.6}_{\pm 0.0}$	$36.5_{(20)}$
NQ	$13.7_{(3)}$	19.7	$19.6_{\pm 0.0}$	$\mathbf{29.9}_{(64)}$
RealTimeQA	$34.7_{(3)}$	34.7	$\mathbf{36.0}_{\pm 0.0}$	$35.4_{(1)}$
WebQs	$29.1_{(3)}$	44.2	$\mathbf{44.1}_{\pm 0.0}$	$41.5_{(64)}$

Across 20 tasks, reformatting to open-ended prompts results in a:

23% performance improvement over the few-shot baseline

Investigating the effectiveness of open-ended questions

1. GPT-J-6B is trained on 300B token Pile corpus [1]. On a 2% random subsample of The Pile:
   - Question-patterns are more frequent
   - There are imbalances in the frequencies of "yes vs. no", "true vs. false"

2. At a class-conditional level, there are larger imbalances in performance using zero-to-few shot prompting vs. AMA. Class-imbalances in the Pile appear to be reflected in performance.

Category	Word Counts
Restrictive Prompt Words	true: 69658 false: 41961 neither: 20891 yes: 12391 no: 452042 maybe: 36569
Yes-No Question Prompt Words	Is: 3580578 Was: 1926273 Did: 200659 Do: 394140 Are: 1441487 Will: 619490
Open-Ended Question Prompt Words	When: 583237 Where: 303074 Why: 97324 Who: 417798 What: 548896 How: 298140

Benchmark	Output Space	F1-Score 0-shot	F1-Score Few-shot *two in-context examples per class*	F1-Score AMA QA *single prompt-chain with no aggregation*
CB	True, False, Neither	True: 36.8 False: 0.0 Neither: 21.7	True: 55.6 False: 0.0 Neither: 12.5	True: 95.7 False: 92.3 Neither: 28.6
RTE	True, False	True: 40.4 False: 58.3	True: 70.6 False: 31.3	True: 58.5 False: 64.9
WSC	Yes, No	Yes: 53.5 No: 0.0	Yes: 53.5 No: 13.7	Yes: 61.3 No: 78.2

Restrictive Prompts

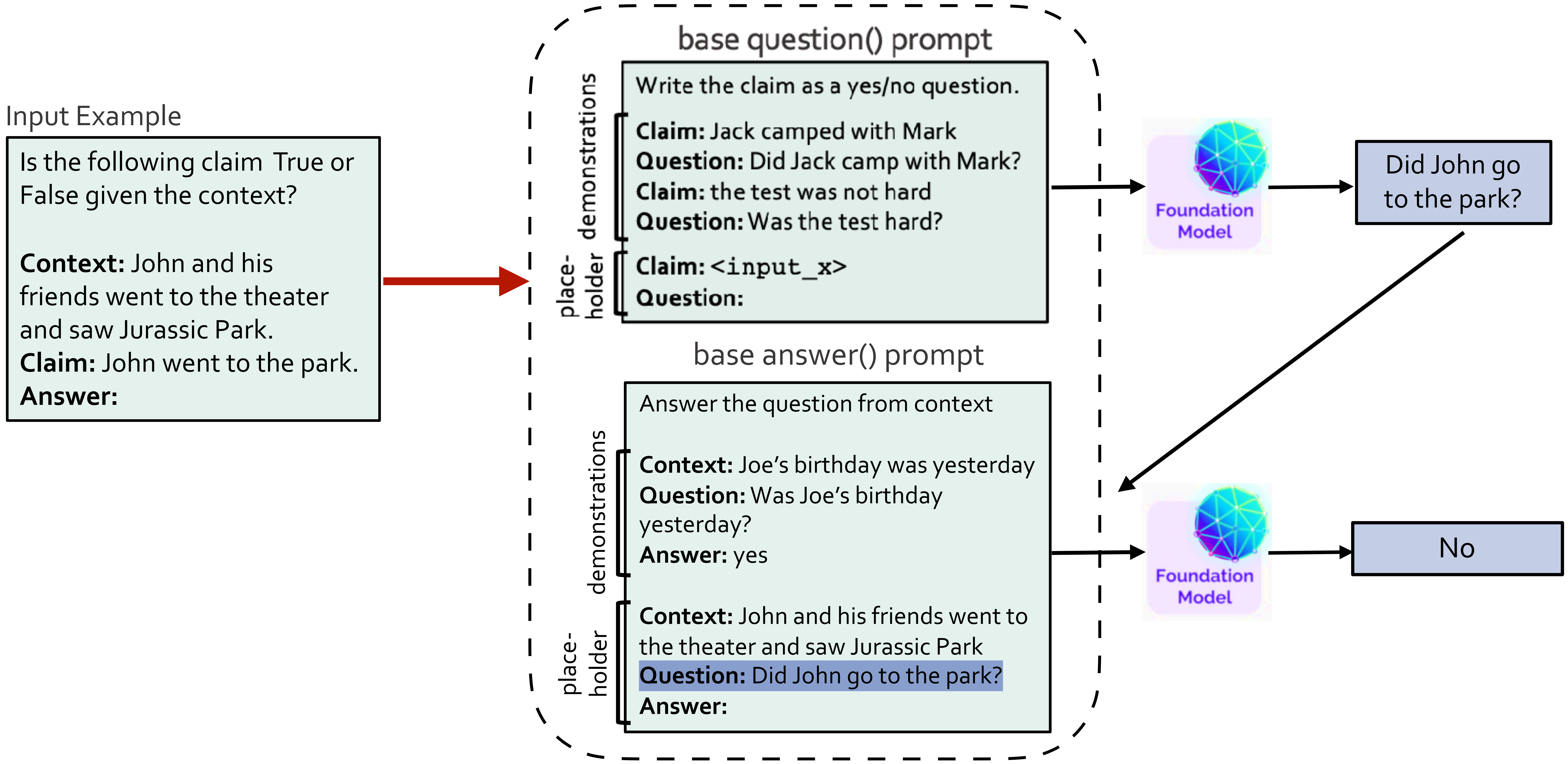
Open-Ended Prompts

[1] Gao, Biderman, Black, Golding, Hope, Foster, Hang, He, That, Nabeshima, Presser, Leahy et al., The Pile: An 800GB Dataset of Diverse Text for Language Modeling, 2020.

How can we reformat inputs to the effective prompt formats at scale?

Prompt-chains

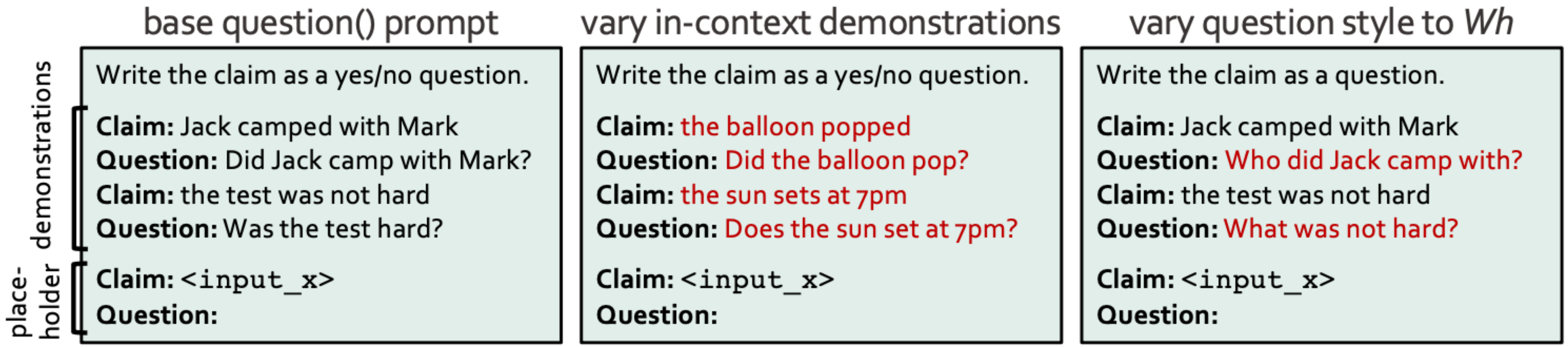
With task-agnostic operations that recursively use the LM.



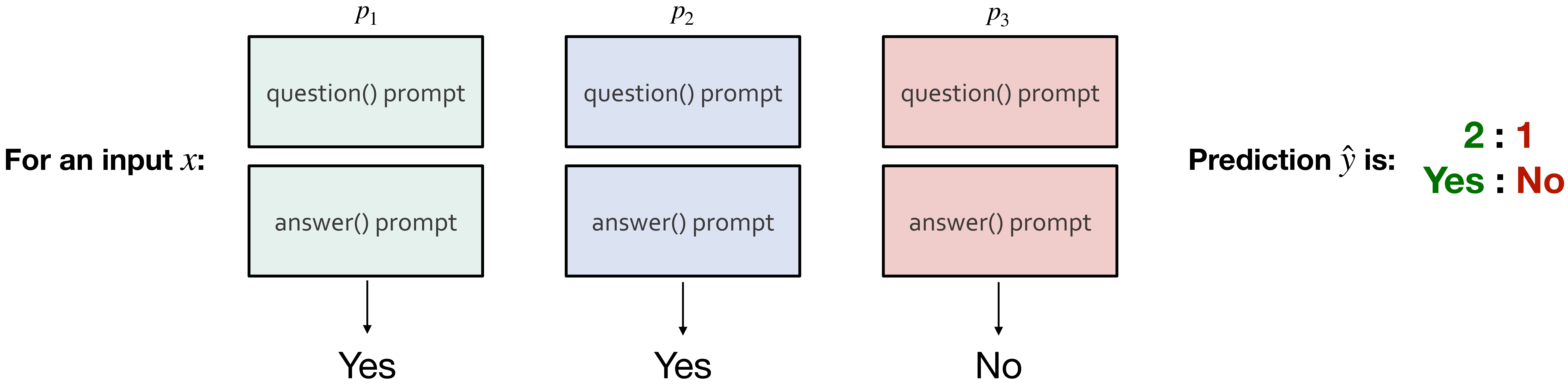
Generating the “perfect prompt” is challenging…

So we produce multiple prompts-chains to obtain multiple predictions per example, then aggregate over the collection!

To obtain varied prompt-chains:



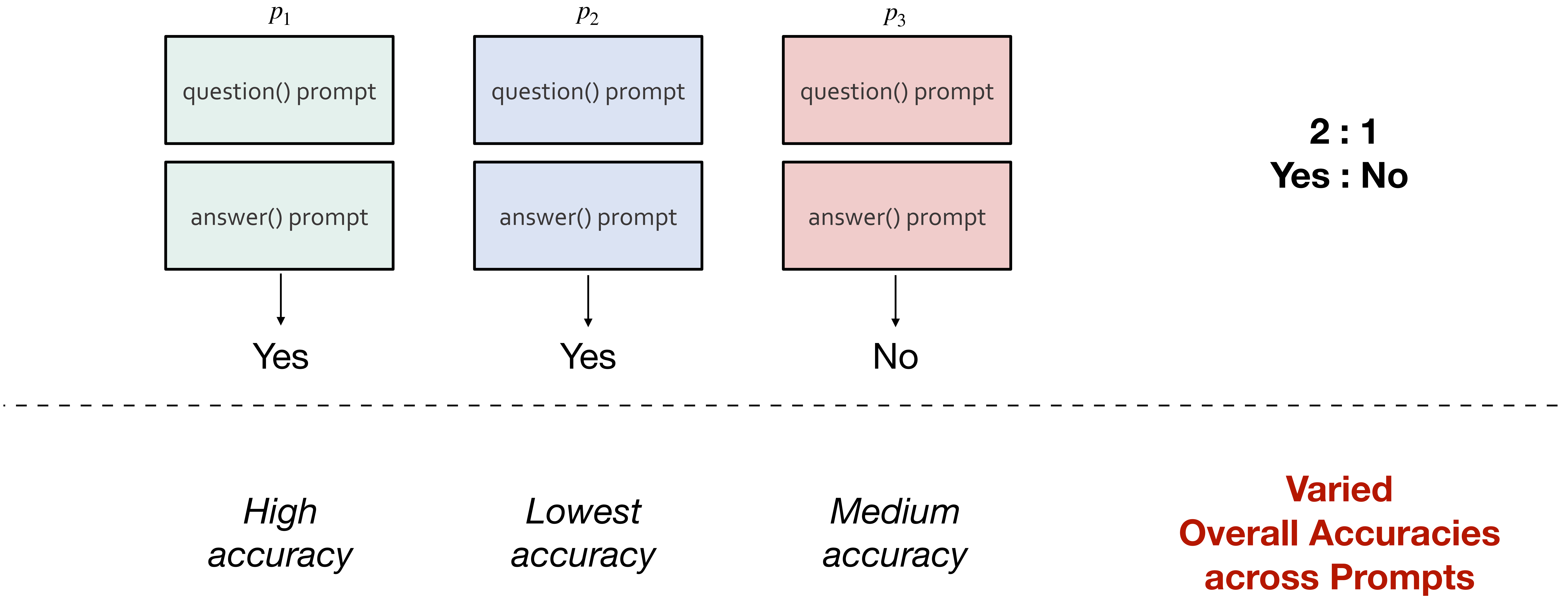
Prior Work Aggregates Using Majority Vote [1, 2]



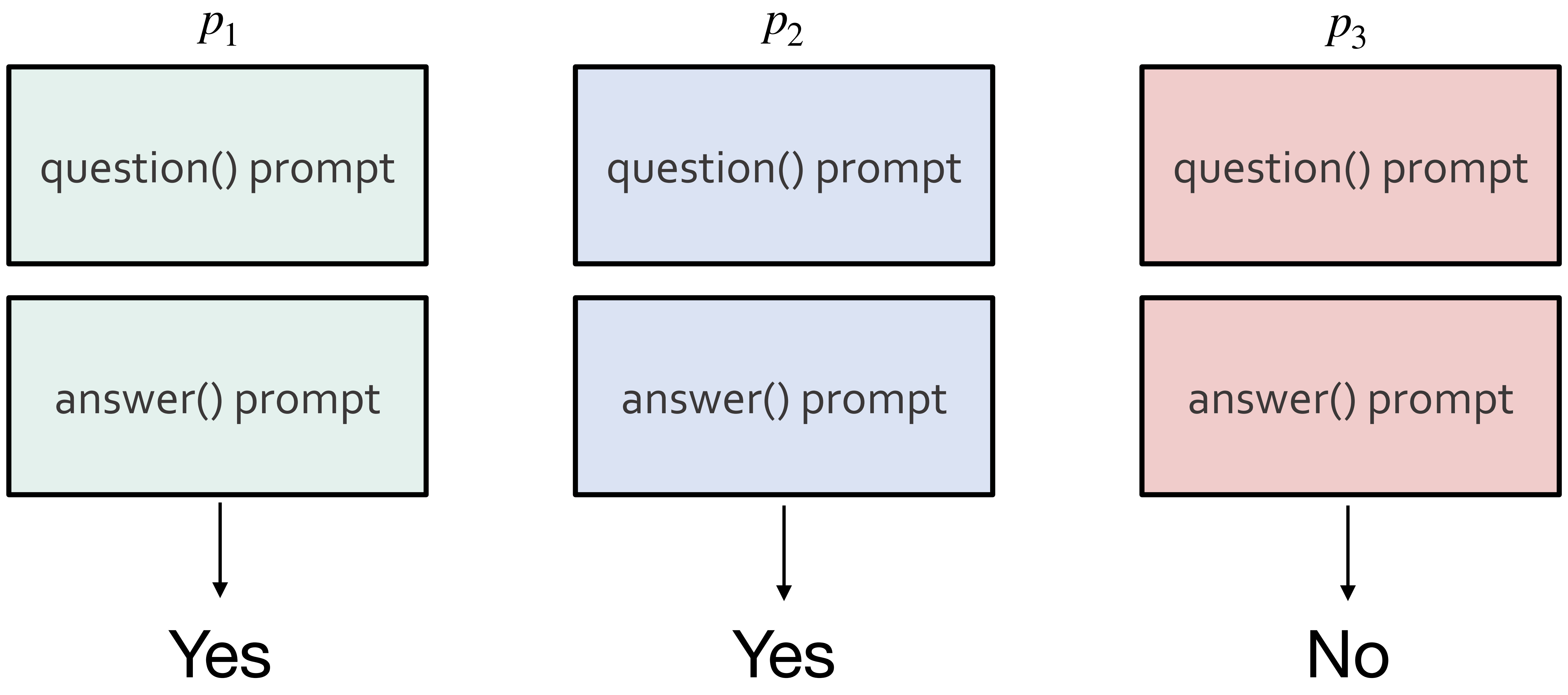
Majority Vote tends to do better than using one prompt, but it weights all prompts equally and treats them independently. In practice, the prompts display properties that make these assumptions suboptimal.

[1] Jiang et al., How can we know what language models know?, *TACL*, 2020.
[2] Schick and Schütze, It's not just size that matters: Small language models are also few-shot learners, 2021.

Majority Vote is not Reliable



Majority Vote is not Reliable



2 : 1
Yes : No

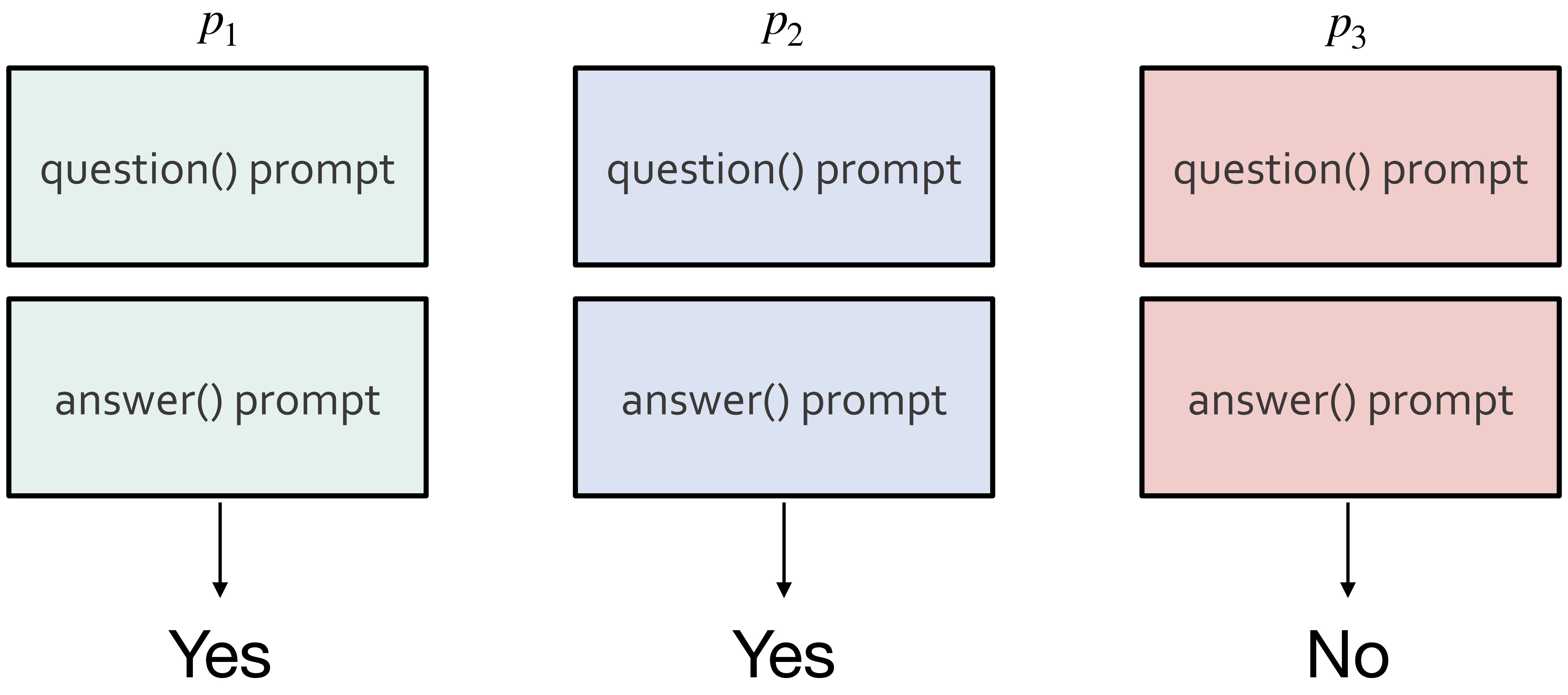
Relatively high quality on "no" class! Poor on "yes".

Relatively high quality on "yes" class! Poor on "no".

Decent quality on "yes" and "no" classes

Varied Class-Conditional Accuracies across Prompts

Majority Vote is not Reliable



2 : 1
Yes : No

Tend to vote together…
Their vote gets "double"-counted

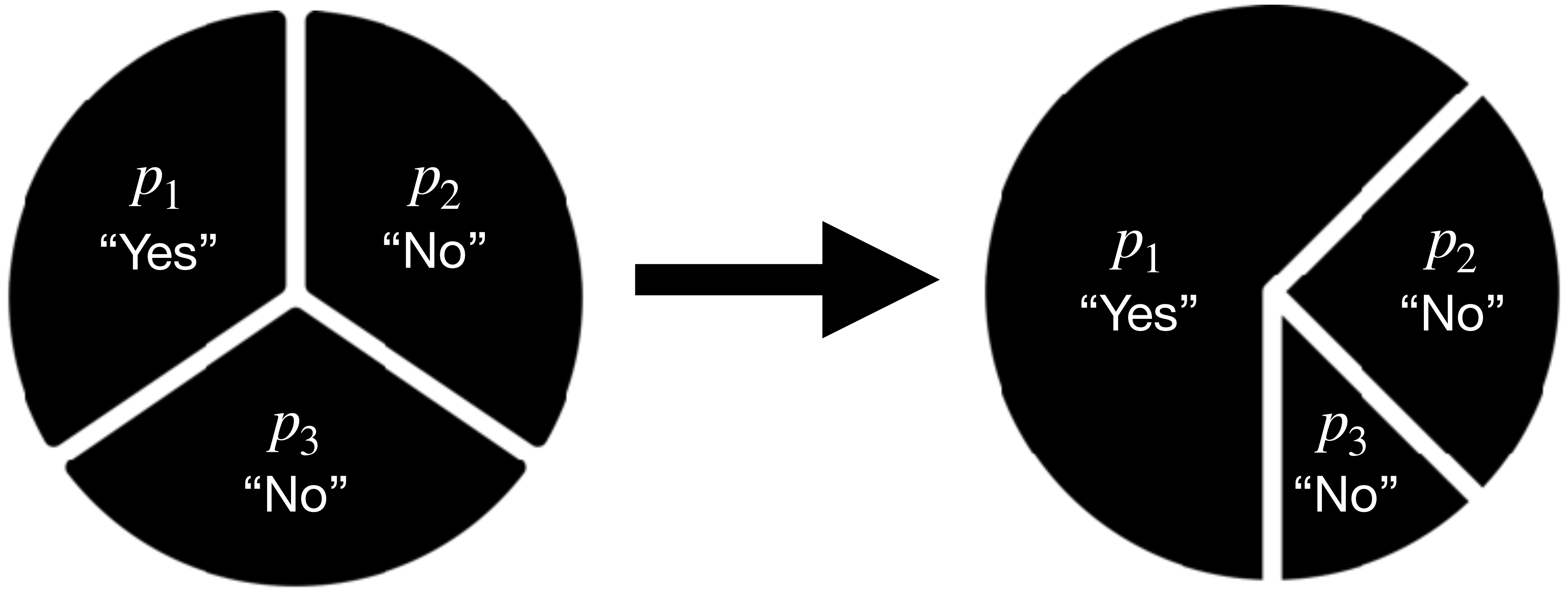
Prompt Predictions have Dependencies (Highly Correlated Outputs)

How can we *reliably* aggregate the predictions?

Suppose the "votes" on an example x are "yes" by p_1, "no" by p_2, and "no" by p_3. And, suppose we have a score of how "good" each prompt is. We want to answer:

What is the ***probability*** that the true label y is "yes"?

Rather than *always* giving each voter equal power, we want to model the relationships between them. Viewing each prompt $p \in \mathbb{P}$ as a random variable, we want to model: $y \mid \mathbb{P}(x)$



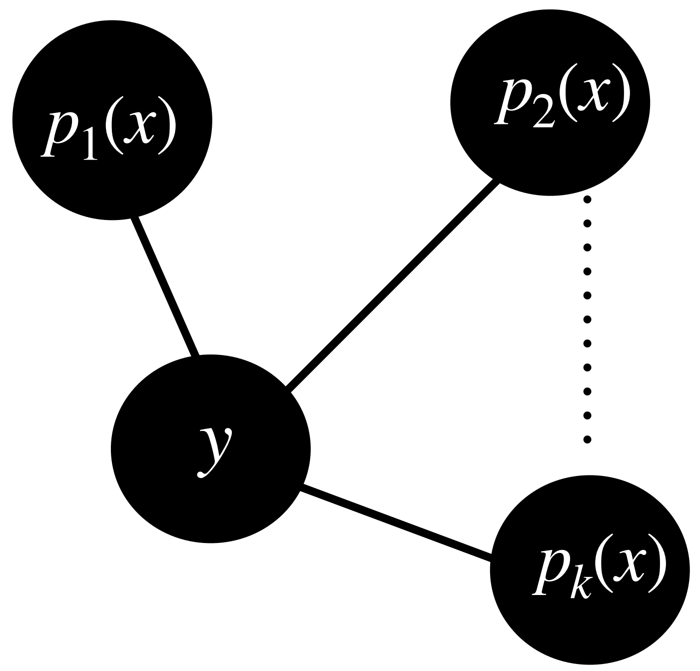
Majority vote: "No"   Weak supervision: "Yes"

[1] Ratner et al., Snorkel: Rapid Training Data Creation with Weak Supervision, 2017.

How can we *reliably* aggregate the predictions?

Formally, our objective is to learn $\phi(\cdot)$, the *aggregator function*, which takes the predictions by $p \in \mathbb{P}$ on input x, expressed as $\mathbb{P}(x)$, and outputs the *final prediction* $\hat{y}$:

$$\phi(x) = \arg\max_{y\in\mathcal{Y}} \Pr_{G,\Theta}\left(y \mid \mathbb{P}(x)\right)$$



An edge exists iff the prompts dependent conditioned on y and the other source labels

$G = (V, E)$ models dependencies

Θ are the accuracies for $p \in \mathbb{P}$

Challenge: We don't have labeled data in our setting, so how can we estimate G, Θ?

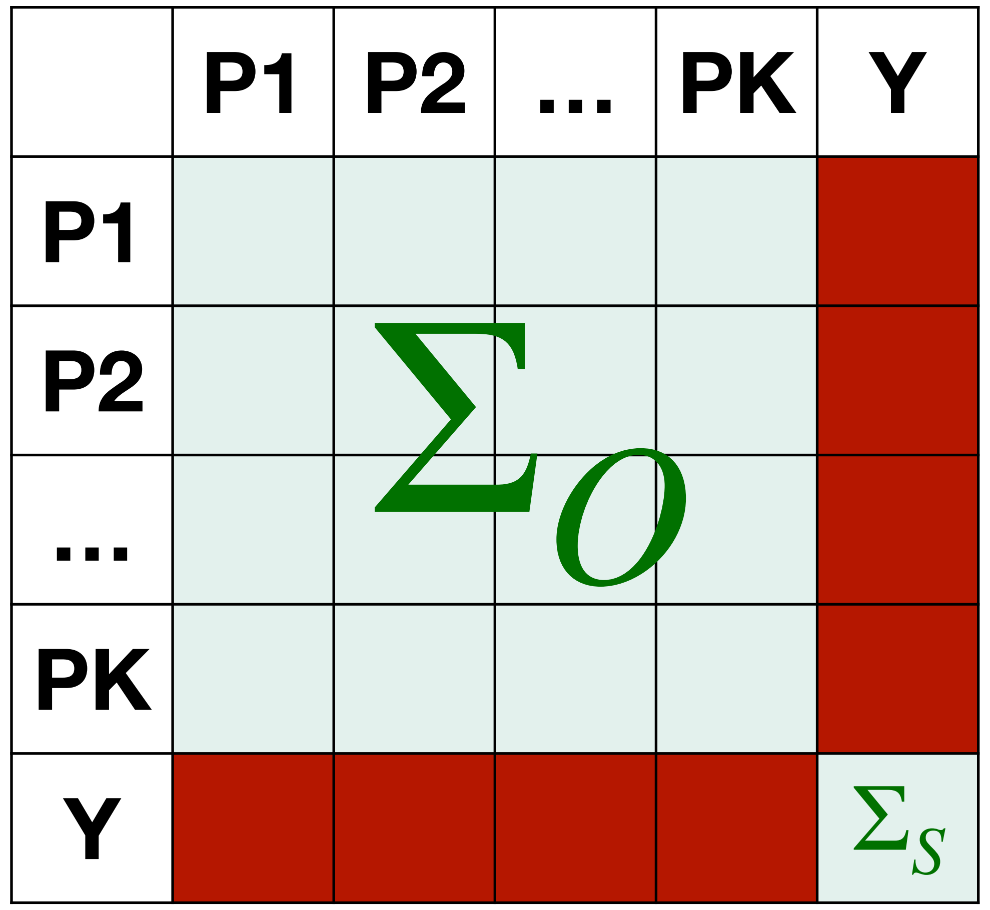
[1] Ratner et al., Snorkel: Rapid Training Data Creation with Weak Supervision, 2017.

Recovering $\hat{G}, \hat{\Theta}$ without labeled data

Key insight: we can use the covariance matrix Σ, i.e. the matrix representing how frequently p_i and p_j predict the same label across inputs our unlabeled dataset $D = \{x_i\}_{i=1}^{n}$! How?

Label y is unobservable. Let's decompose Σ into its observable O and unobservable S terms:

$\Sigma =$



- Σ_O and Σ_S are available
- $\Sigma_{O \cup S}$ is our unknown term and it's a function of $\hat{\Theta}$.
  $E[yp_i]$ is proportional to the accuracy of prompt-chain p_i.
  If we solve for $\Sigma_{O \cup S}$, we can recover $\hat{\Theta}$!

Evaluating AMA's aggregation strategy

We find that AMA can achieve up to 8.7 points of lift over majority vote, improving reliability!

	# Prompts	Avg	MV	WMV	Pick Best	AMA (no dep)	AMA (WS)
No labels:			✓			✓	✓
Natural Language Understanding							
WSC	3	74.7	77.8	77.8	75.0	$77.8_{\pm 0.0}$	**77.8**$_{\pm 0.0}$
WiC	5	59.0	61.3	60.9	60.0	$60.8_{\pm 0.0}$	**61.3**$_{\pm 0.2}$
RTE	5	61.4	66.0	71.4	62.0	$65.1_{\pm 0.5}$	**75.1**$_{\pm 0.0}$
CB	3	83.3	82.1	82.1	83.9	$82.1_{\pm 0.0}$	**83.9**$_{\pm 0.0}$
MultiRC	3	58.8	63.8	63.4	63.4	$63.7_{\pm 0.0}$	**63.8**$_{\pm 0.0}$
BoolQ	5	64.9	65.9	67.2	**68.3**	$65.9_{\pm 0.0}$	$67.2_{\pm 0.0}$
COPA	4	58.3	**85.0**	82.0	82.0	$84.0_{\pm 0.0}$	$84.0_{\pm 0.0}$
Natural Language Inference							
ANLI R1	5	34.6	37.6	36.1	36.8	$37.4_{\pm 1.0}$	**37.8**$_{\pm 0.2}$
ANLI R2	5	35.4	36.3	36.0	36.0	**38.7**$_{\pm 0.4}$	$37.9_{\pm 0.2}$
ANLI R3	5	37.0	39.0	38.4	38.4	$39.6_{\pm 0.9}$	**40.9**$_{\pm 0.5}$
StoryCloze	6	76.3	**87.9**	81.8	81.8	$82.2_{\pm 0.0}$	$87.8_{\pm 0.0}$
Classification							
DBPedia	3	81.4	**84.1**	83.9	82.2	$83.9_{\pm 0.0}$	$83.9_{\pm 0.0}$
SST2	3	94.5	95.7	95.7	95.2	$95.7_{\pm 0.0}$	**95.7**$_{\pm 0.0}$
Amazon	3	67.0	68.6	68.6	67.3	$68.6_{\pm 0.0}$	**68.6**$_{\pm 0.0}$
AGNews	3	83.7	**86.5**	84.2	83.8	$86.4_{\pm 0.0}$	$86.4_{\pm 0.0}$

Examining the importance of AMA prompt reformatting

We take the prompt-templates directly from the GPT-3 paper. We find that applying multiple prompts in these templates and aggregating the predictions is not effective:

Aggregation with no prompt-reformatting

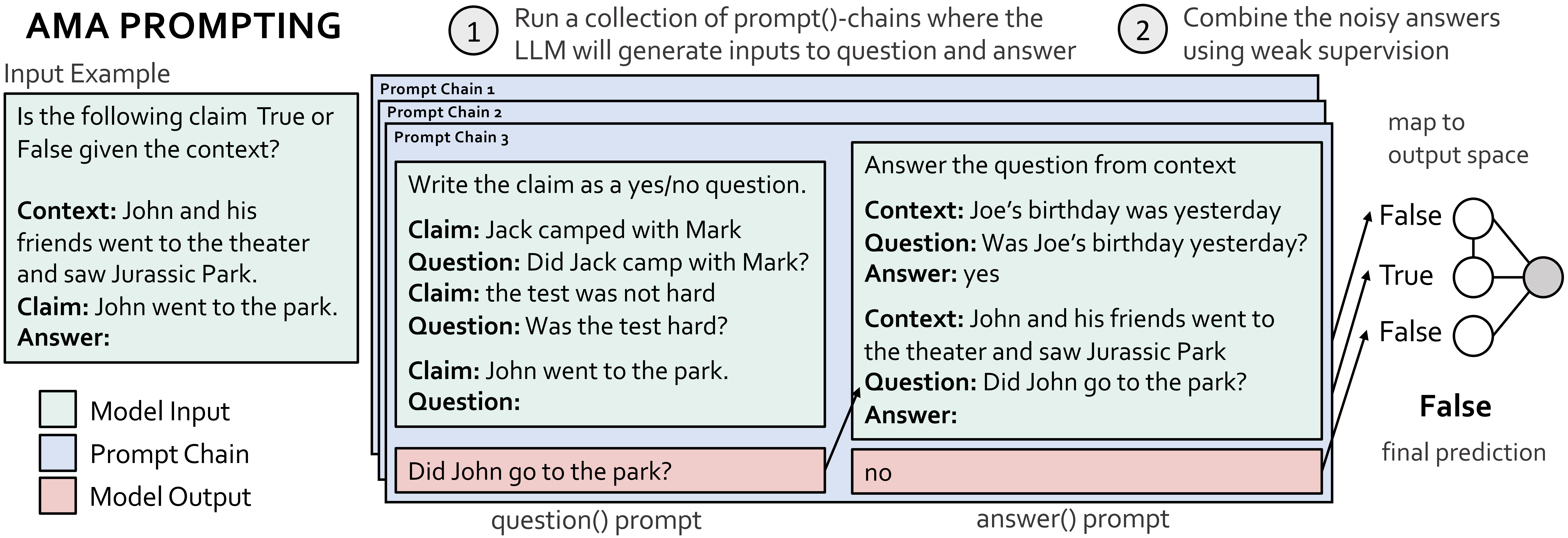
AMA: reformatting and aggregation

Model	GPT-J Few-Shot	GPT-J Few-Shot	GPT-J Few-Shot	GPT-J AMA
Aggregation	Average	Majority Vote	Weak Supervision	Weak Supervision
Natural Language Understanding				
CB	23.8	17.9	50.0	83.9
RTE	53.5	53.1	54.2	75.1
WSC	46.2	38.5	38.5	77.9
COPA	80.0	81.0	81.0	84.0
Natural Language Inference				
ANLI R1	33.4	33.5	33.5	37.8
ANLI R2	33.2	32.9	32.2	37.9
ANLI R3	35.4	36.5	34.6	40.2
Classification				
AGNews	70.3	70.7	75.0	86.4
Amazon	61.9	62.4	62.5	68.2

+28 %

+39 %

Ask Me Anything (AMA)

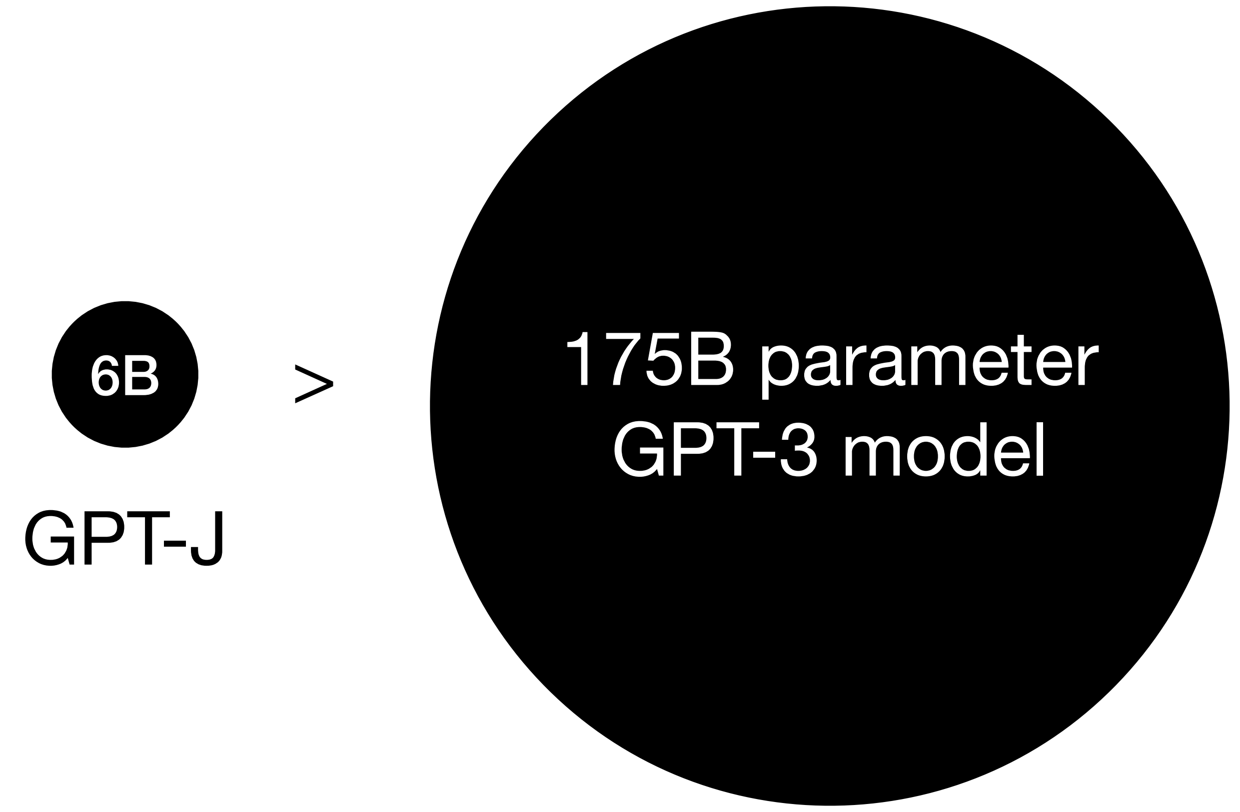


Evaluations

We Beat GPT-3 on their Benchmarks!

Model	Neo Few-Shot	Neo (QA)	Neo (QA + WS)	GPT-3 Few-Shot
# Params	6B	6B	6B	175B
Natural Language Understanding				
BoolQ	66.5$_{(3)}$	64.9	67.2$_{\pm 0.0}$	**77.5**$_{(32)}$
CB	25.0$_{(3)}$	83.3	**83.9**$_{\pm 0.0}$	82.1$_{(32)}$
COPA	77.0$_{(3)}$	58.2	84.0$_{\pm 0.0}$	**92.0**$_{(32)}$
MultiRC	60.8$_{(3)}$	58.8	63.8$_{\pm 0.0}$	**74.8**$_{(32)}$
ReCoRD	75.6$_{(3)}$	74.5	74.4$_{\pm 0.0}$	**89.0**$_{(32)}$
RTE	58.8$_{(3)}$	61.7	**75.1**$_{\pm 0.0}$	72.9$_{(32)}$
WSC	36.5$_{(3)}$	74.7	**77.9**$_{\pm 0.0}$	75.0$_{(32)}$
WiC	53.3$_{(3)}$	59.0	**61.0**$_{\pm 0.2}$	55.3$_{(32)}$
Natural Language Inference				
ANLI R1	32.3$_{(3)}$	34.6	**37.8**$_{\pm 0.2}$	36.8$_{(50)}$
ANLI R2	33.5$_{(3)}$	35.4	**37.9**$_{\pm 0.2}$	34.0$_{(50)}$
ANLI R3	33.8$_{(3)}$	37.0	**40.9**$_{\pm 0.5}$	40.2$_{(50)}$
StoryCloze	51.0$_{(3)}$	76.3	**87.8**$_{\pm 0.0}$	87.7$_{(70)}$
Classification				
AGNews	74.5$_{(3)}$	83.7	**86.4**$_{\pm 0.0}$	79.1$_{(8)}$
Amazon	62.5$_{(3)}$	66.8	**68.2**$_{\pm 0.0}$	41.9$_{(8)}$
DBPedia	50.7$_{(3)}$	81.4	**83.9**$_{\pm 0.0}$	83.2$_{(8)}$
SST	64.9$_{(3)}$	94.5	**95.7**$_{\pm 0.0}$	95.6$_{(8)}$
Question-Answering				
DROP	32.3$_{(3)}$	51.0	**51.6**$_{\pm 0.0}$	36.5$_{(20)}$
NQ	13.7$_{(3)}$	19.7	19.6$_{\pm 0.0}$	**29.9**$_{(64)}$
RealTimeQA	34.7$_{(3)}$	34.7	**36.0**$_{\pm 0.0}$	35.4$_{(1)}$
WebQs	29.1$_{(3)}$	44.2	**44.1**$_{\pm 0.0}$	41.5$_{(64)}$

With an open-source model that's 1/30th the size!



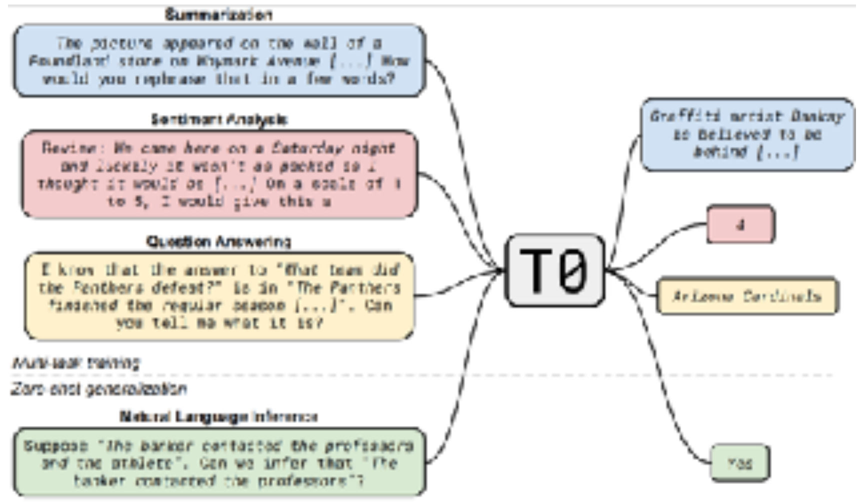
Small models still struggle with long, noisy contexts and factual knowledge

Results Generalize Across Model Types and Sizes

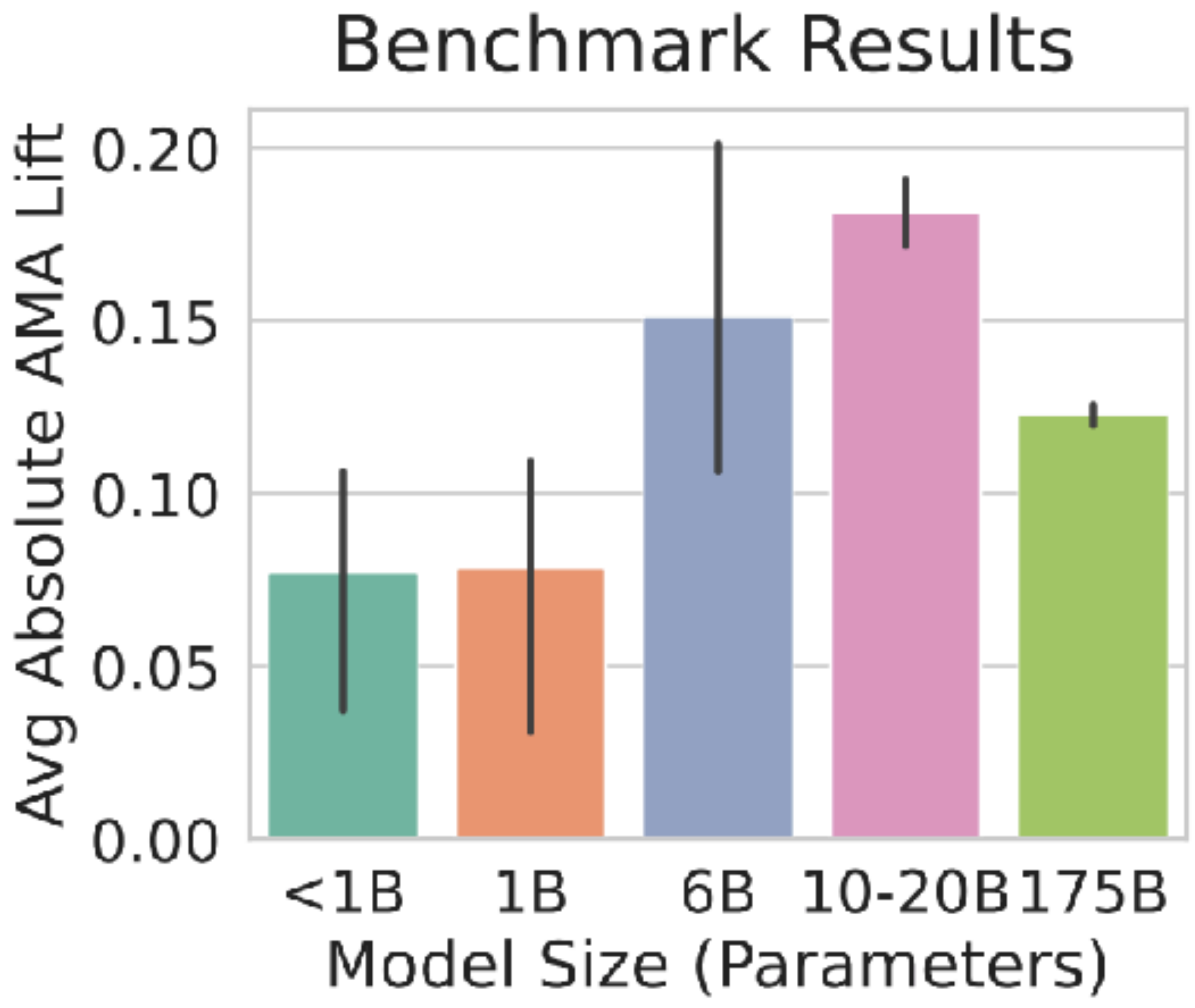
a BigScience initiative

BLOOM

176B params · 59 languages · Open-access



We see lift across model sizes (125M-176B) and type (BLOOM, OPT, Neo) for autoregressive models!



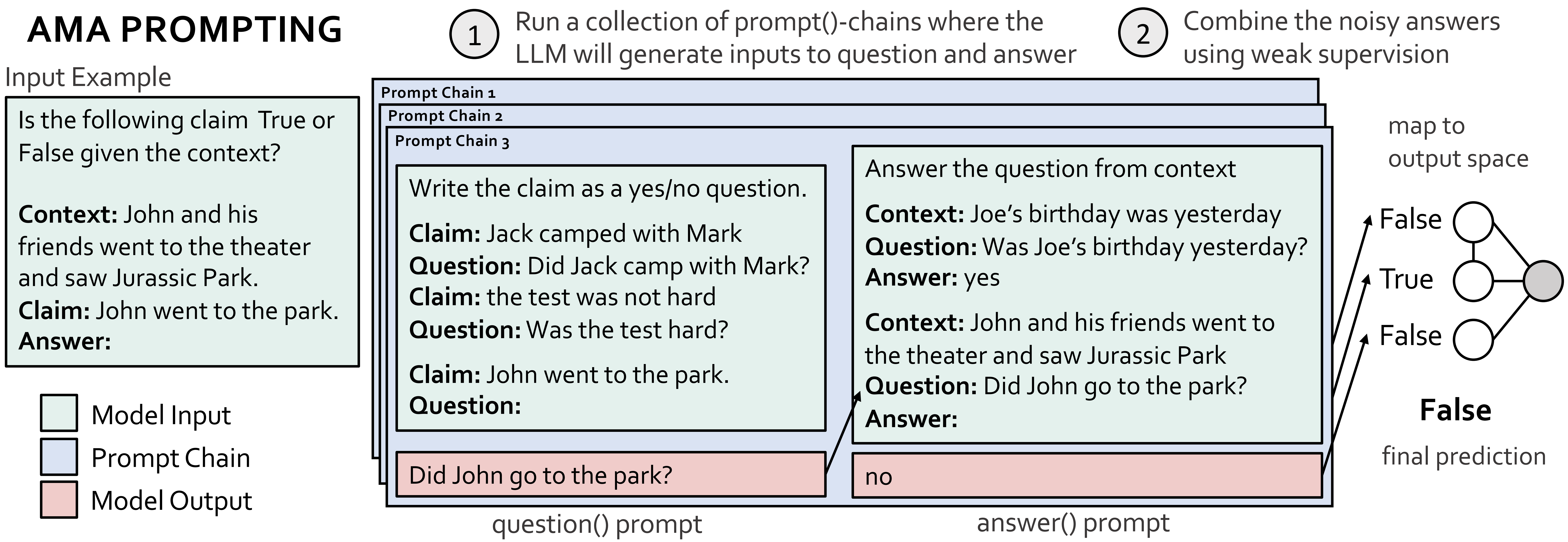
Average

10.2 ± 6.1 (absolute)

21.4 ± 11.2 (relative)

across 14 foundation models

Ask Me Anything (AMA)



Conclusion

Paper: https://arxiv.org/abs/2210.02441
Code: https://github.com/HazyResearch/ama_prompting
Blog: https://www.numbersstation.ai/post/ask-me-anything

Contact: simran@cs.stanford.edu

Thanks to my amazing lab mates & advisor & collaborators!


Avanika


Ines


Laurel


Mayee


Neel


Kush

Snorkel

Center for Research on Foundation Models

Numbers Station

TOGETHER

Thank you!

Contact: simran@cs.stanford.edu

Find additional resources at:

Code: https://github.com/HazyResearch/ama_prompting
Paper: https://arxiv.org/abs/2210.02441
Blogs: https://hazyresearch.stanford.edu/blog