Washington University in St. Louis
James McKelvey School of Engineering

Evaluation of Language Models

Presenters: Yuqi Chen, Siqi Ma, Meichuan Yin

Date: 11/05/2024

Overview

01 Proving Test Set Contamination in Black Box Language Models

02 Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation

03 Large Language Models are not Fair Evaluators

04 Holistic Evaluation of Language Models

Washington University in St. Louis
James McKelvey School of Engineering

PROVING TEST SET CONTAMINATION IN BLACK BOX LANGUAGE MODELS

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, Tatsunori B. Hashimoto

ICLR2024

https://arxiv.org/abs/2310.17623

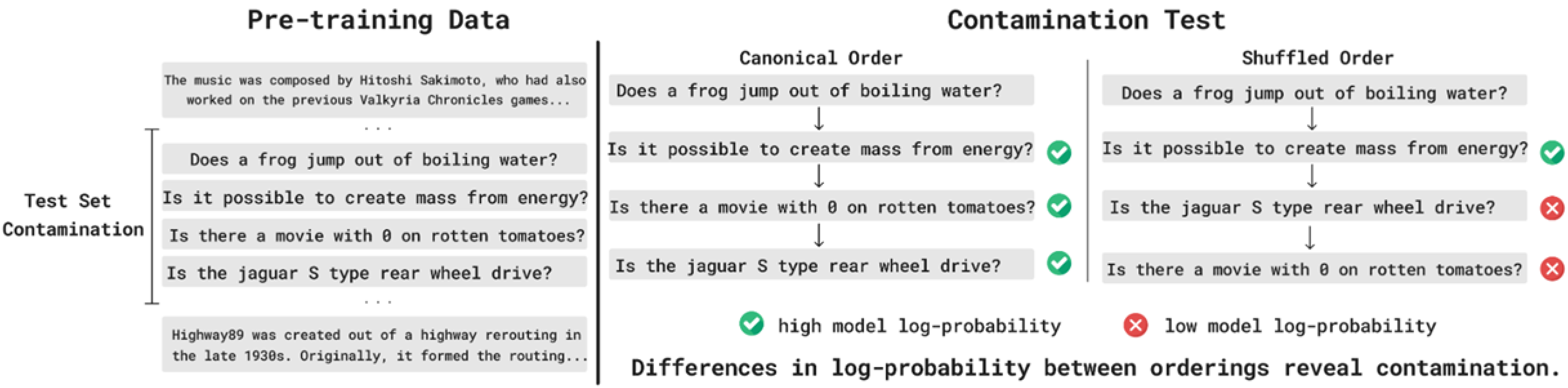
Background

PROVING TEST SET CONTAMINATION IN BLACK BOX LANGUAGE MODELS

Yonatan Oren[1*], **Nicole Meister**[1*], **Niladri Chatterji**[1*], **Faisal Ladhak**[2], **Tatsunori B. Hashimoto**[1]
[1]Stanford University, [2]Columbia University

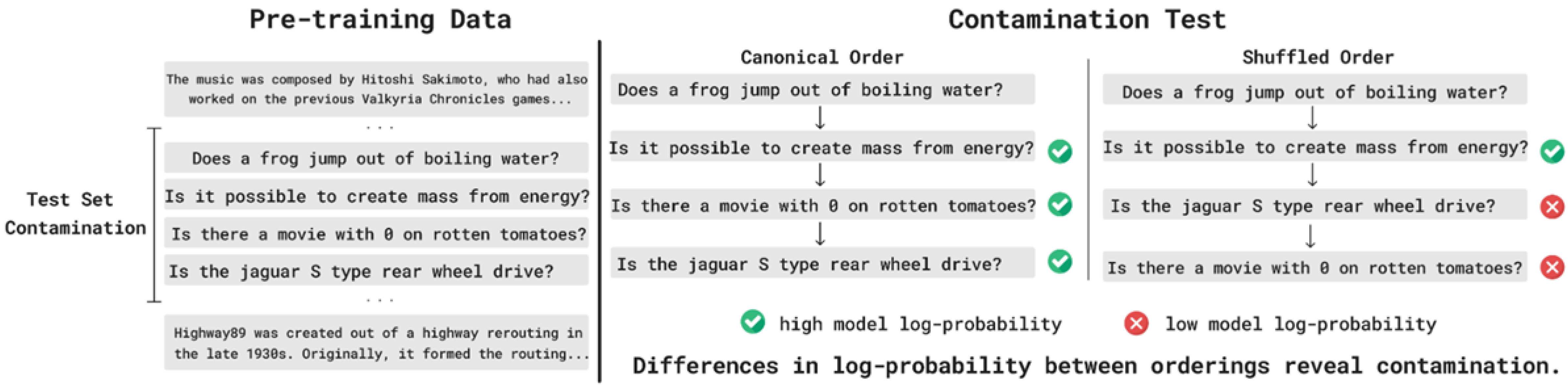
- LLM facing big challenge: **the Contamination of Dataset**
- Whether LLMs are Memorize the Answers or **Generalization**
- **Closed source** dataset

Aim



- Provide provable tests of **test set contamination** in **black box language models**

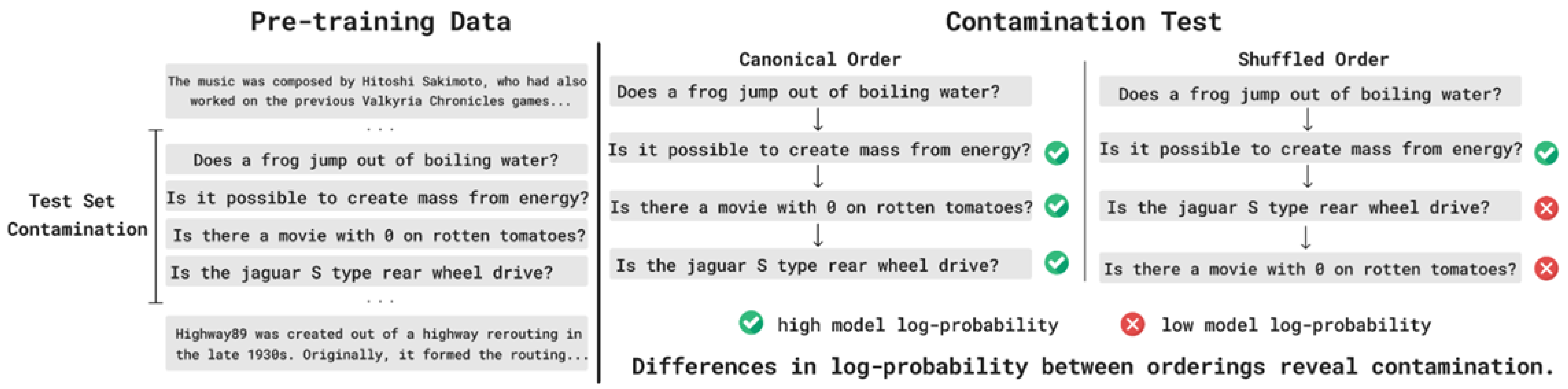
Aim



A well-known property is introduced to detect contamination:

Exchangeability: the order of examples in the dataset can be shuffled without affecting its joint distribution

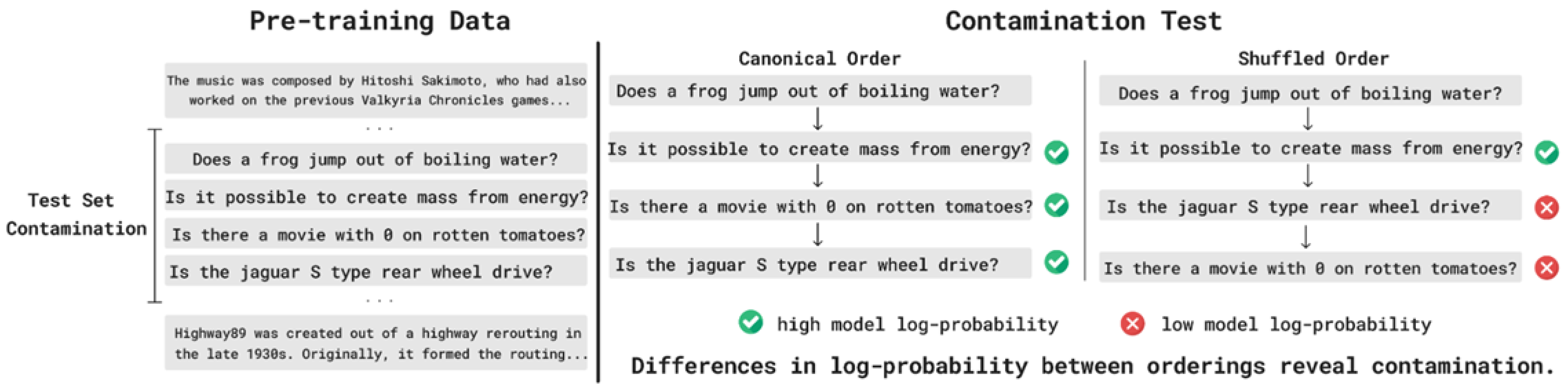
Aim



Compare the **log probability** of the model:

1. **With a standard dataset (no change)**
2. **With a dataset of shuffled examples**

Contributions



1. **Exchangability could be used to identify test set contamination**
2. **An sliced hypothesis test for test set contamination**
3. **Empirical demonstration of black-box detection of contamination for small datasets that appear few times during pretraining**

Problem Setting

Provably identifying test set contamination can be viewed as a hypothesis test in which the goal is to distinguish between two hypotheses:

- H_0: θ is independent of X **No contamination**
- H_1: θ is dependent on X **Contamination**

θ is the training process of a language model

X is the dataset

If a model satisfies Exchangability, we have:

$$\log p_\theta(seq(X)) \overset{d}{=} \log p_\theta(seq(X_\pi))$$ **No contamination**

$$\log p_\theta(\mathrm{seq}(X)) < \log p_\theta(\mathrm{seq}(X_\pi))$$ **Contamination**

$\mathrm{seq}(X)$ means the sequence of whole dataset X, π is one of the permutation

Method

Computational Complexity :

It is clearly impractical to count all possible permutations of a data set

Solution:

1. Cut the dataset into several pieces:

$$S_1 = (X_1, X_2, \cdots, X_k)$$

2. Permute the examples within each cut, estimate of the average likelihood of the shuffled order :

$$s_i := \log p_\theta(\mathrm{seq}(X)) - \mathrm{Mean}_\pi(\log p_\theta(\mathrm{seq}(X_\pi)))$$

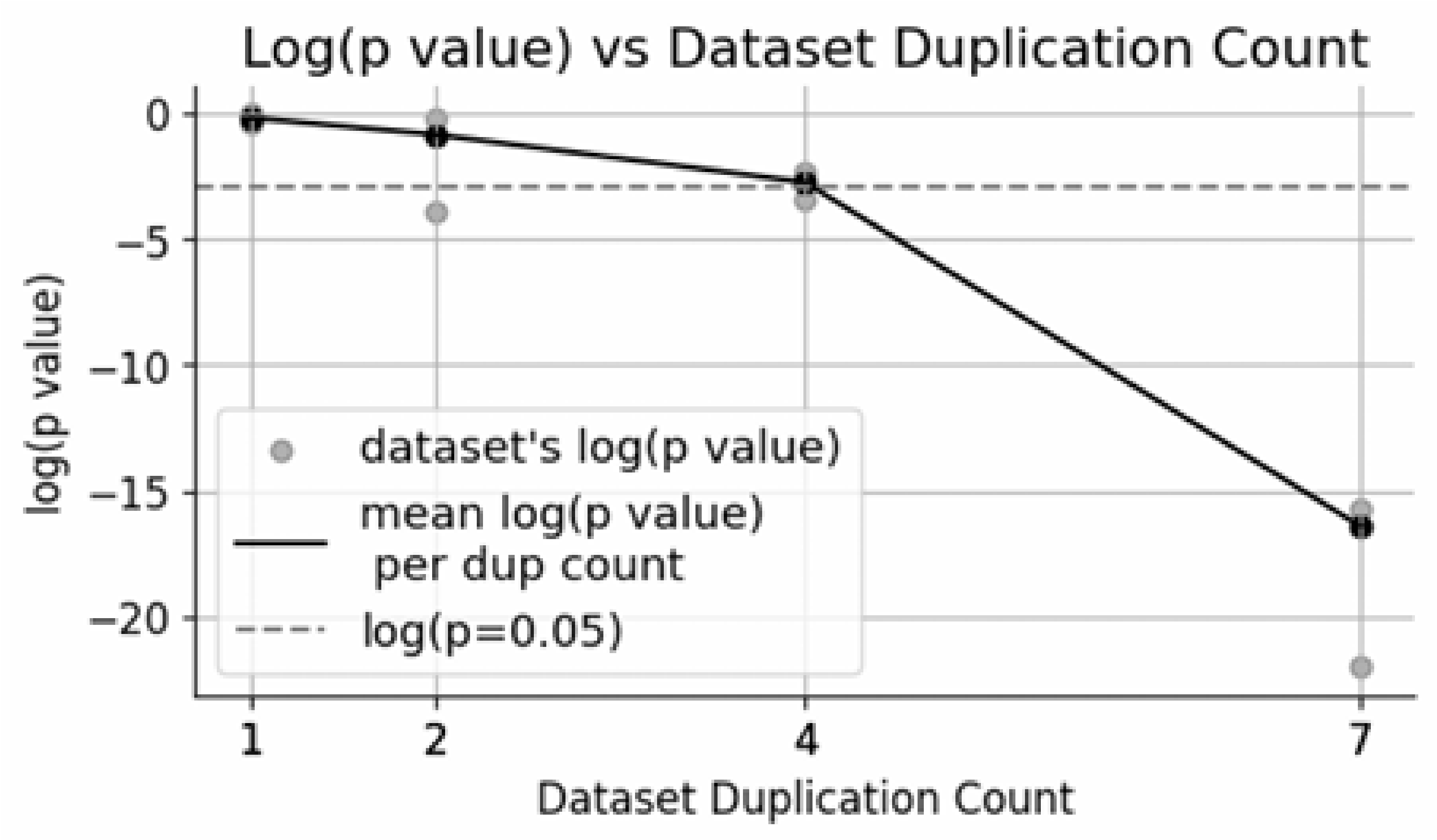
Where π is one of the permutation

Experiment

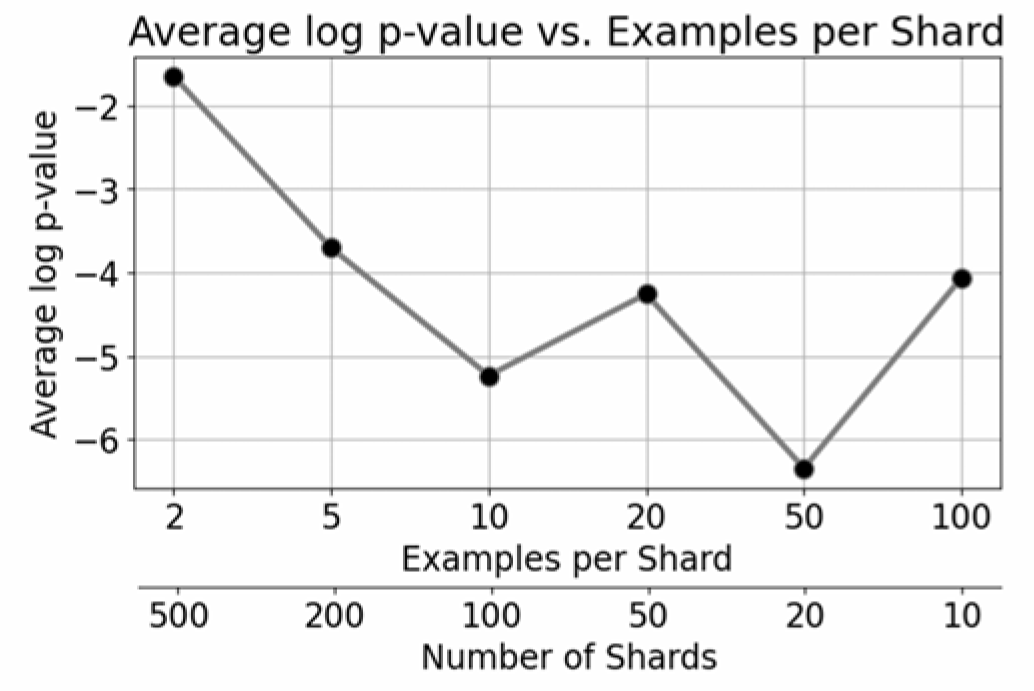
Name	Size	Dup Count	Permutation p	Sharded p
BoolQ	1000	1	0.099	0.156
HellaSwag	1000	1	0.485	0.478
OpenbookQA	500	1	0.544	0.462
MNLI	1000	10	**0.009**	**1.96e-11**
TruthfulQA	1000	10	**0.009**	**3.43e-13**
Natural Questions	1000	10	**0.009**	**1e-38**
PIQA	1000	50	**0.009**	**1e-38**
MMLU Pro. Psychology	611	50	**0.009**	**1e-38**
MMLU Pro. Law	1533	50	**0.009**	**1e-38**
MMLU H.S. Psychology	544	100	**0.009**	**1e-38**

Size means the number of examples, Dup Count means the Frequency of injection of test set
Higher p means higher probability of choosing hypothesis H0

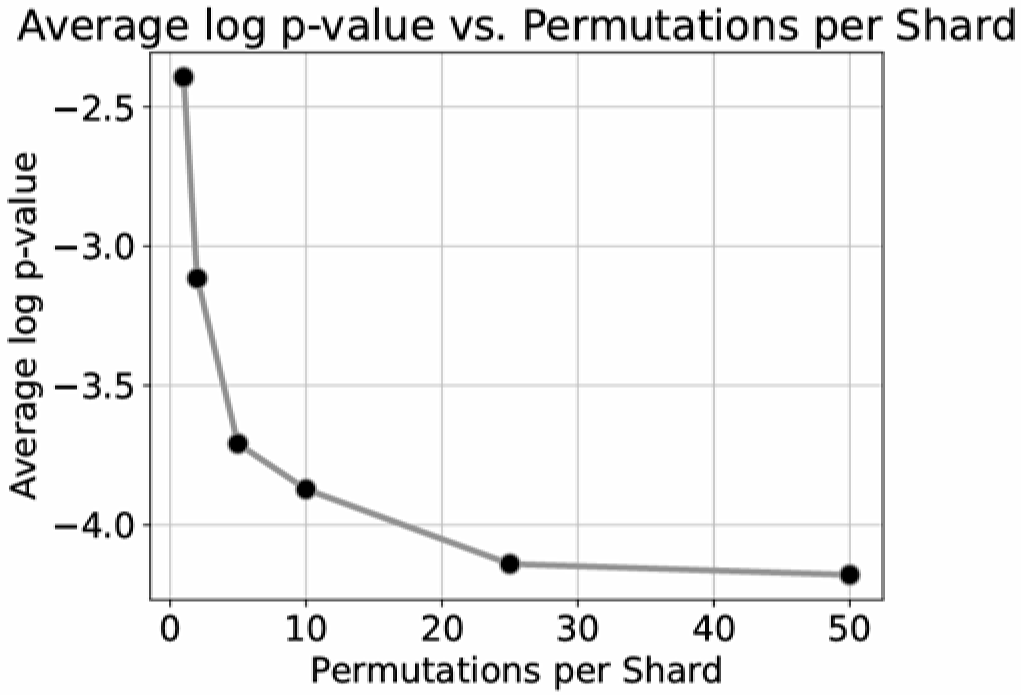
Experiment


Log(p value) vs Dataset Duplication Count
0
−5
−10
−15
−20
log(p value)
dataset's log(p value)
mean log(p value)
per dup count
log(p=0.05)
1
2
4
7
Dataset Duplication Count

Experiment



Method can not detect contamination with too many cuts



More cuts lead to more accurate detection

Experiment

Dataset	Size	LLaMA2-7B	Mistral-7B	Pythia-1.4B	GPT-2 XL	BioMedLM
Arc-Easy	2376	0.318	**0.001**	0.686	0.929	0.795
BoolQ	3270	0.421	0.543	0.861	0.903	0.946
GSM8K	1319	0.594	0.507	0.619	0.770	0.975
LAMBADA	5000	0.284	0.944	0.969	0.084	0.427
NaturalQA	1769	0.912	0.700	0.948	0.463	0.595
OpenBookQA	500	0.513	0.638	0.364	0.902	0.236
PIQA	3084	0.877	0.966	0.956	0.959	0.619
MMLU†	–	0.014	0.011	0.362	–	–

Mistral-7B seems to have some level of contamination on Arc-Easy.

Note that those datasets are not guaranteed to have Exchangeability.

Limitations

- No guarantee on the Exchangeability of off-the-shelf benchmark dataset.

  We cannot know that a dataset is exchangable without knowing its data generating process

- Only direct contamination can be detected.

Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, Lingming Zhang

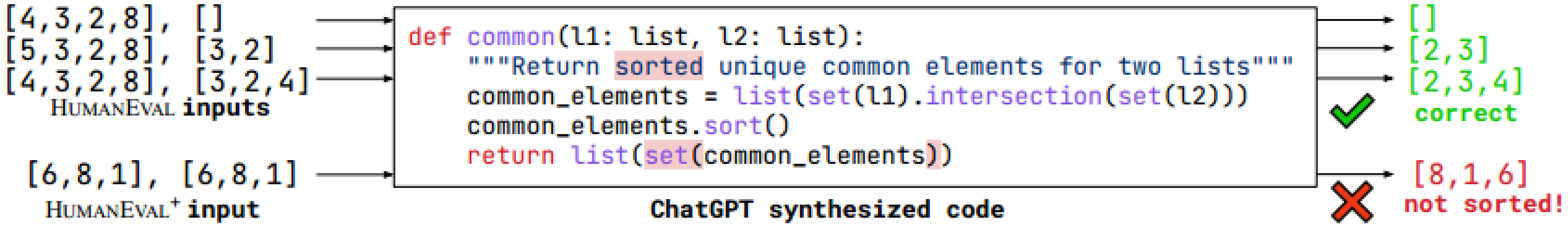
NIPS '23

https://arxiv.org/abs/2305.01210

Problem Statement

As advances in LLMs have significantly improved the ability to generate code, researchers have come to rely on these models for program synthesis. However, existing code evaluation benchmarks (e.g., HUMANEVAL) have limitations in terms of the number and quality of tests, making it difficult to comprehensively assess the functional correctness of generated code.

- Insufficient testing
- Imprecise problem description



Overview of EvalPlus

- Automated Test Input Generation
- Seed initialization via ChatGPT
- Type-aware input mutation
- Test-Suite Reduction
- Code coverage
- Mutant killings
- LLM sample killings
- Program Input Contracts

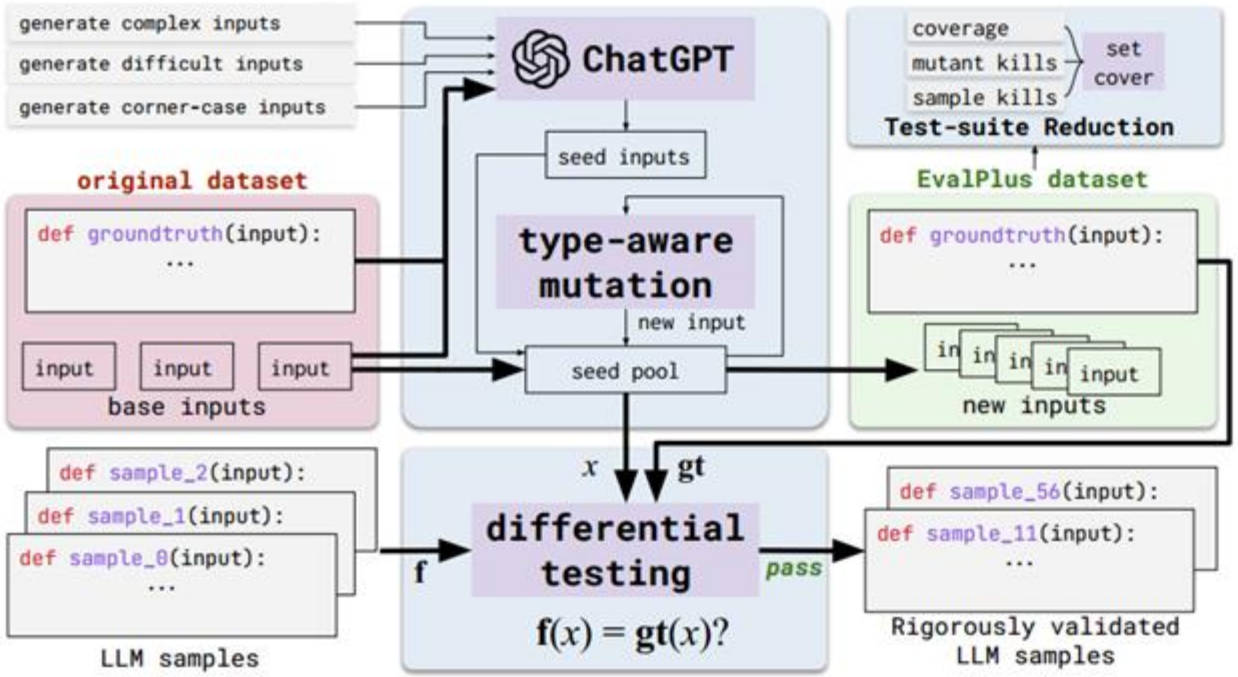


Figure 2: Overview of EvalPlus

Seed initialization via ChatGPT

- Constructed prompts containing real solutions to problems for ChatGPT to examine and refer to.
- Provide a set of test inputs as examples to help ChatGPT understand the task.
- Add instructions to encourage ChatGPT to create interesting input content.

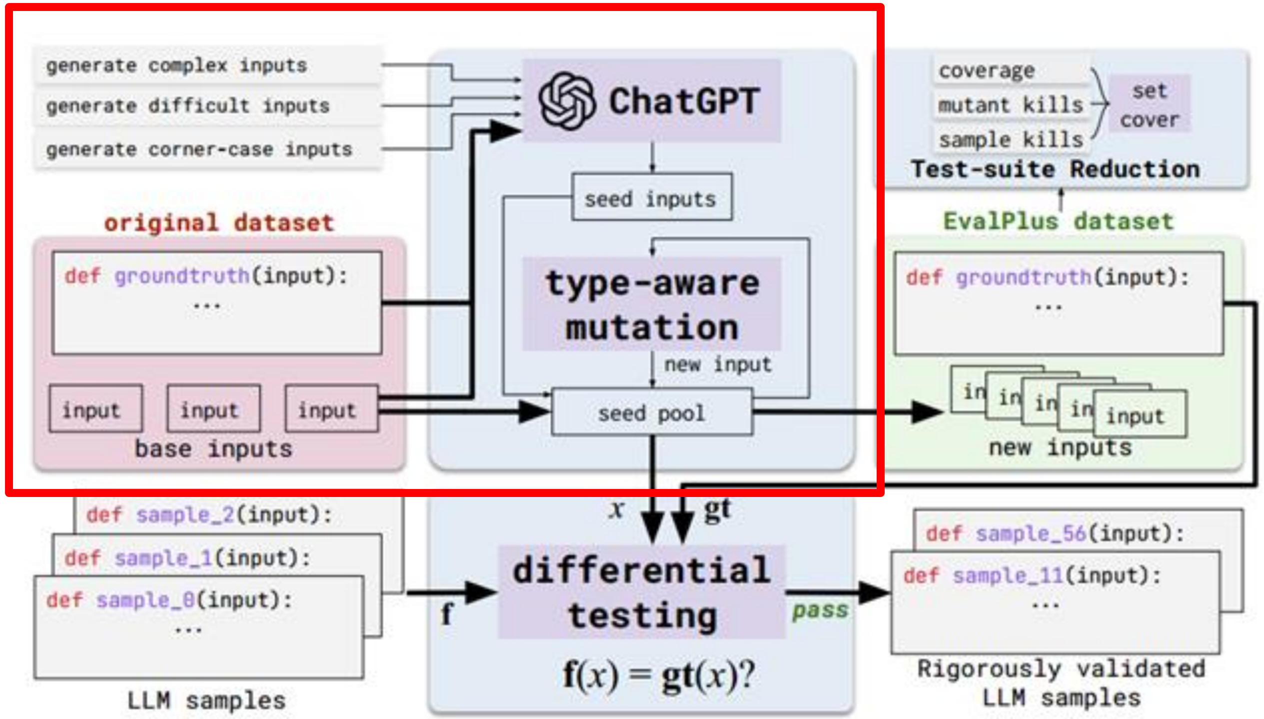


Figure 2: Overview of EvalPlus

Type-aware input mutation

- Input generation and mutation process: initialize the generation pool based on seed inputs (generated by ChatGPT) and generate new inputs by randomly selecting seeds for mutation.
- Diversified mutation strategy: apply specific mutation methods based on different types of data (e.g., integers, floats, and composite types).

Type	Mutation	Type	Mutation
`int` | `float`	Returns $x \pm 1$	`List`	Remove/repeat a random item $x[i]$ Insert/replace $x[i]$ with `Mutate`$(x[i])$
`bool`	Returns a random boolean	`Tuple`	Returns `Tuple(Mutate(List(`x`)))`
`NoneType`	Returns `None`	`Set`	Returns `Set(Mutate(List(`x`)))`
`str`	Remove a sub-string s Repeat a sub-string s Replace s with `Mutate`(s)	`Dict`	Remove a key-value pair $k \rightarrow v$ Update $k \rightarrow v$ to $k \rightarrow$ `Mutate`(v) Insert `Mutate`$(k) \rightarrow$ `Mutate`(v)

HUMANEVAL+ And HUMANEVAL+-MINI

Based on HUMANEVAL+ which on average obtains 764.1 tests for each programming task (Table 2), our test-suite reducer (§2.2) minimizes it to HUMANEVAL+MINI which only has 16.1 tests for each task (smaller by 47×).

Table 2: Overview of EvalPlus-improved benchmarks.

	#Tests				#Tasks
	Avg.	Medium	Min.	Max.	
HUMANEVAL	9.6	7.0	1	105[2]	164
HUMANEVAL^{+}	764.1	982.5	12	1,100	
HUMANEVAL^{+}-MINI	16.1	13.0	5	110	

Overview of EvalPlus

- **Code Coverage**: Code coverage measures the amount of code elements (e.g., statements or branches) executed by tests to assess test effectiveness. In this strategy, branch coverage is used as the testing requirement, with the goal of retaining a minimal subset of tests that covers the same set of branches as the full test suite.

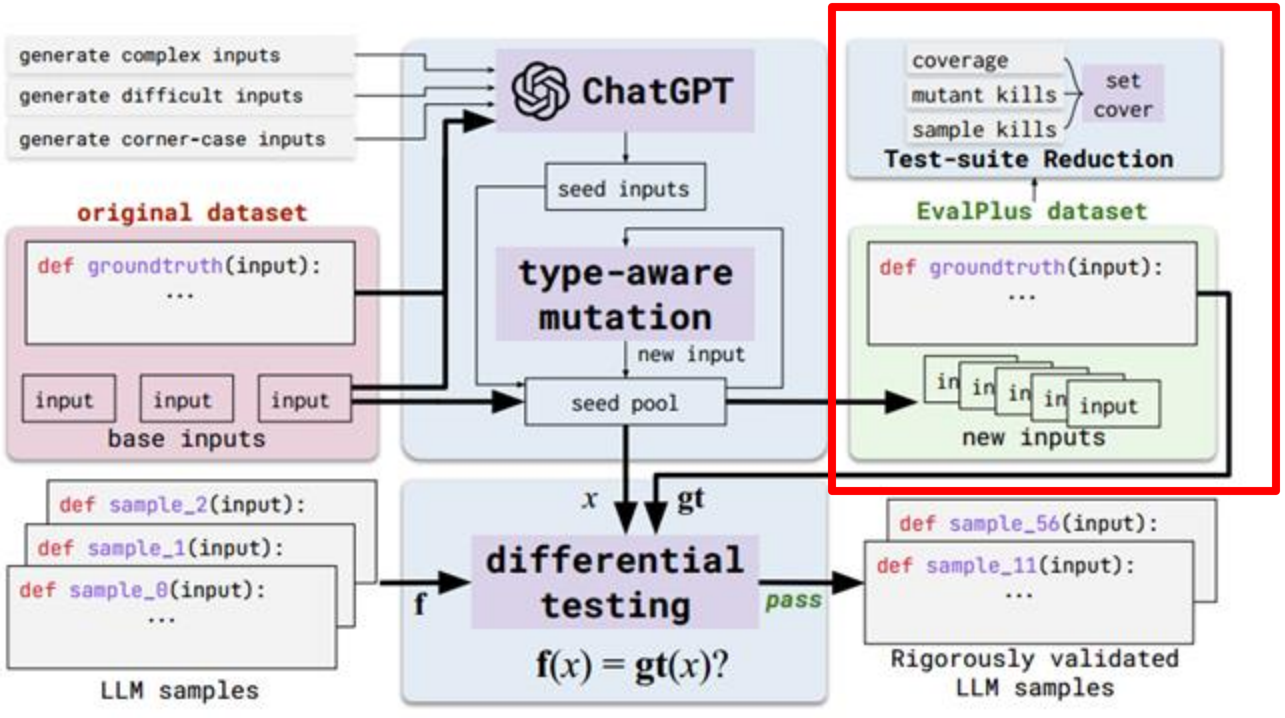


Figure 2: Overview of EvalPlus

Overview of EvalPlus

- **Mutant Killings**: While code coverage indicates code execution, it doesn't necessarily reveal critical defects. Mutation testing addresses this by injecting subtle bugs (mutants) into the code to create artificial faulty programs. The ratio of mutants detected (or "killed") by tests is used to measure test effectiveness. This approach generally outperforms code coverage in evaluating test quality.

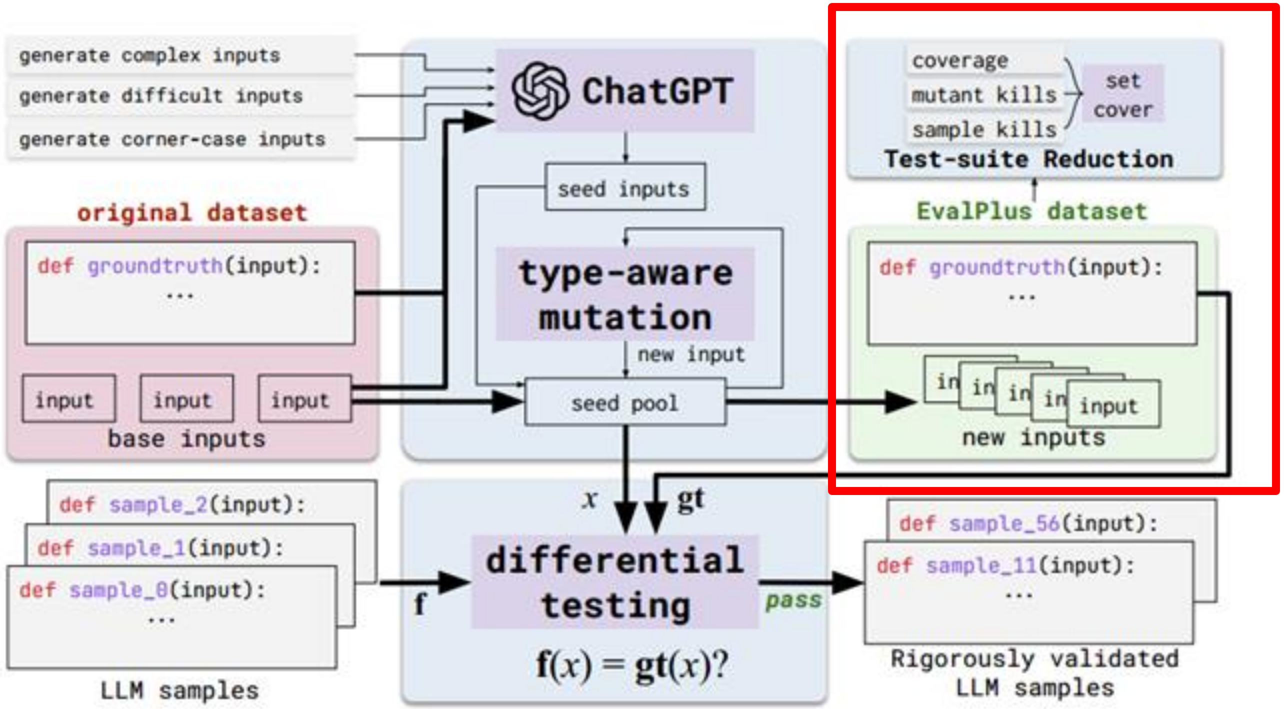


Figure 2: Overview of EvalPlus

Overview of EvalPlus

- **LLM Sample Killings**: Different LLMs may exhibit similar failures on certain test cases. To measure test effectiveness, we also consider sample killings, which reflect the number of incorrect LLM outputs a test case can detect. For new LLMs, since we lack execution results, we rely on results from other LLMs' samples to ensure the reduced test suite still detects all incorrect samples.

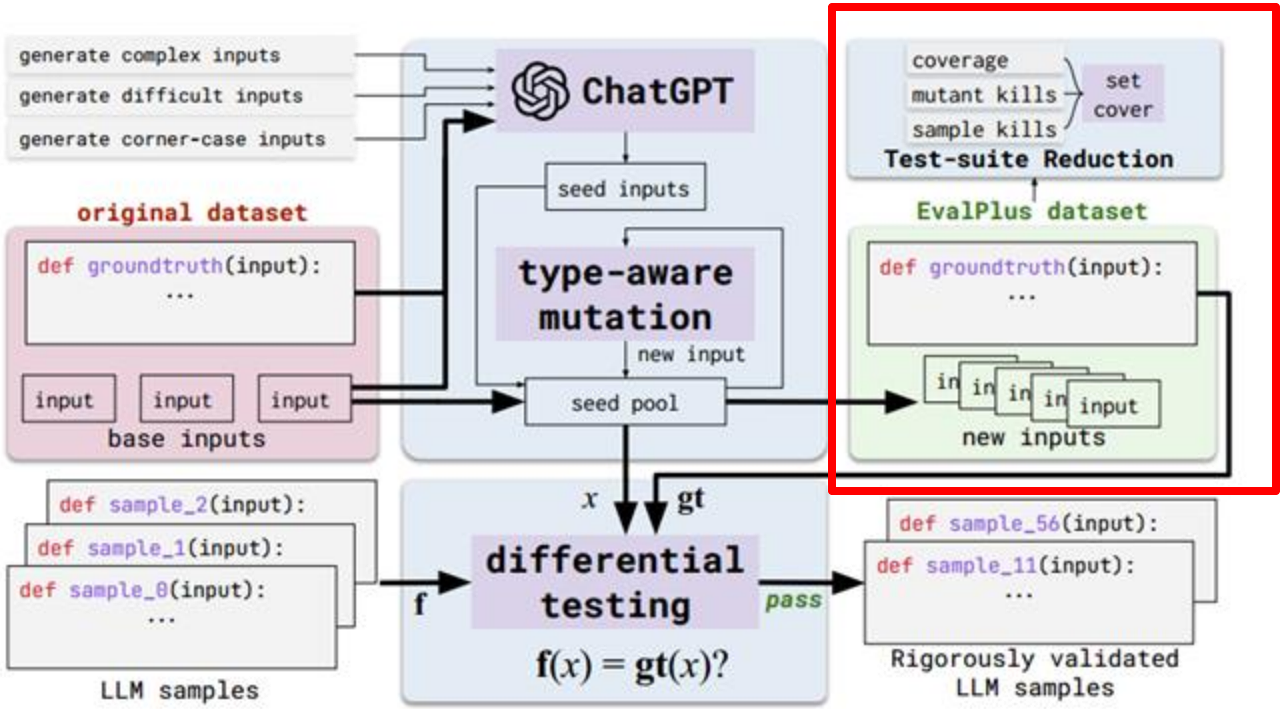


Figure 2: Overview of EvalPlus

Evaluation(1)

	Size	pass@k	k=1*	k=1	k=10	k=100	T_1^*	T_{10}^*	T_{100}^*
GPT-4 [49]	N/A	base	88.4						
		+extra	76.2						
Phind-CodeLlama [52]	34B	base	71.3	71.6	90.5	96.2	.2	.8	.8
		+extra	67.1	67.0	85.0	92.5	.2	.8	.8
WizardCoder-CodeLlama [38]	34B	base	73.2	61.6	85.2	94.5	.2	.8	.8
		+extra	64.6	54.5	78.6	88.9	.2	.8	.8
ChatGPT [48]	N/A	base	73.2	69.4	88.6	94.0			
		+extra	63.4	62.5	82.1	91.1			
CodeLlama [54]	34B	base	51.8	52.0	82.4	95.0	.2	.8	.8
		+extra	42.7	43.1	73.7	89.4	.2	.8	.8
	13B	base	42.7	44.6	77.6	92.7	.4	.8	.8
		+extra	36.6	37.4	69.4	88.2	.4	.8	.8
	7B	base	37.8	39.2	69.1	89.7	.2	.8	.8
		+extra	34.1	34.5	61.4	82.9	.2	.8	.8
StarCoder [13]	15B	base	34.1	32.2	56.7	84.2	.2	.8	.8
		+extra	29.3	27.8	50.3	75.4	.2	.8	.8
CodeGen [46]	16B	base	32.9	32.2	56.0	81.5	.2	.6	.8
		+extra	26.8	27.2	48.4	71.4	.2	.6	.8
	6B	base	29.3	27.7	46.9	72.7	.2	.6	.8
		+extra	25.6	23.6	41.0	64.6	.2	.6	.8
	2B	base	24.4	18.4	39.8	66.8	.2	.8	.8
		+extra	20.7	15.1	34.8	55.8	.2	.2	.8
CodeT5+ [64]	16B	base	31.7	32.2	58.5	83.5	.2	.6	.8
		+extra	26.2	27.4	51.1	76.4	.2	.6	.8
Mistral [26]	7B	base	28.7	28.1	55.2	83.8	.2	.8	.8
		+extra	23.8	23.7	48.5	76.4	.2	.8	.8

Evaluation(2)

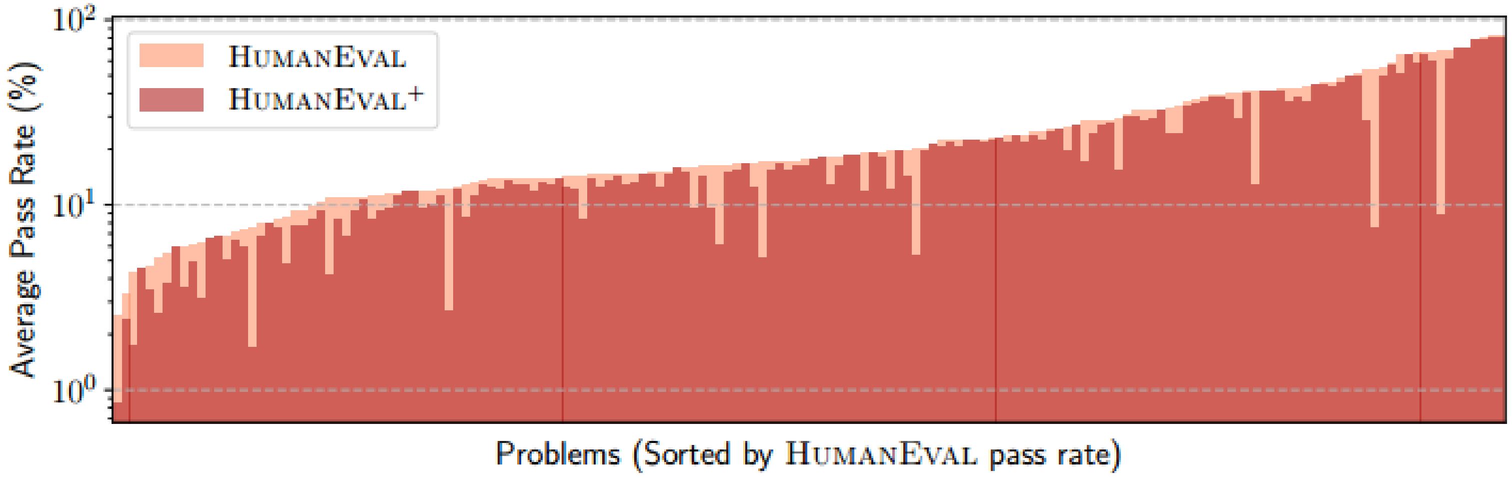
CodeGen2 [45]	16B[4]	base	19.5						
		+extra	16.5						
	7B	base	18.3	17.9	30.9	50.9	.2	.6	.8
		+extra	16.5	15.9	27.1	45.4	.2	.6	.8
	3B	base	15.9	15.2	23.9	38.6	.2	.4	.8
		+extra	12.8	12.9	21.2	34.3	.2	.4	.8
	1B	base	11.0	10.2	15.1	24.7	.2	.6	.6
		+extra	9.1	8.7	13.7	21.2	.2	.6	.6
VICUNA [12]	13B	base	16.5	15.3	30.1	54.8	.2	.8	.8
		+extra	15.2	13.9	25.8	46.7	.2	.8	.8
	7B	base	11.6	10.9	23.8	42.3	.2	.6	.6
		+extra	11.0	10.3	20.3	35.0	.2	.6	.6
SantaCoder [2]	1.1B	base	14.6	16.6	29.2	45.4	.4	.6	.8
		+extra	12.8	14.2	26.2	40.6	.4	.6	.8
INCODER [18]	6.7B	base	15.9	15.6	27.7	45.0	.2	.4	.6
		+extra	12.2	12.4	22.2	38.9	.2	.6	.6
	1.3B	base	12.2	10.0	15.9	25.2	.2	.6	.6
		+extra	10.4	7.9	13.5	20.7	.2	.6	.4
GPT-J [63]	6B	base	12.2	11.3	17.7	31.8	.2	.6	.6
		+extra	10.4	9.5	15.2	25.9	.2	.6	.6
GPT-NEO [5]	2.7B	base	7.9	6.5	11.8	20.7	.2	.6	.6
		+extra	6.7	6.0	9.0	16.8	.2	.6	.6
PolyCoder [70]	2.7B	base	6.1	5.9	10.2	17.1	.2	.4	.6
		+extra	5.5	5.3	7.9	13.6	.2	.6	.6
StableLM [60]	7B	base	2.4	2.7	7.5	15.8	.2	.6	.6
		+extra	2.4	2.6	6.2	11.9	.2	.6	.6

Reduced test-suite for HUMANEVAL+

	Size	Coverage		Killed mutants		Killed samples		**Full**		Ref. pass@1*	
		pass@1*	#tests	pass@1*	#tests	pass@1*	#tests	pass@1*	#tests	base	+extra
GPT-4	N/A	86.0	11.3	82.9	11.4	78.7	13.8	**78.0**	16.1	88.4	76.2
ChatGPT	N/A	71.3	11.3	69.5	11.4	**65.2**	13.7	**65.2**	16.0	73.2	63.4
StarCoder	15B	32.9	11.3	32.9	11.4	**29.3**	13.6	**29.3**	15.9	34.1	29.3
CodeGen	2B	23.2	11.3	23.8	11.4	**21.3**	13.2	**21.3**	15.4	24.4	20.7
	6B	28.7	11.3	29.3	11.4	**25.6**	13.2	**25.6**	15.4	29.3	25.6
	16B	31.7	11.3	31.1	11.4	**27.4**	13.2	**27.4**	15.4	32.9	26.8
CodeGen2	1B	10.4	11.3	11.0	11.4	**9.1**	13.8	**9.1**	16.0	11.0	9.1
	3B	15.9	11.3	15.9	11.4	**12.8**	13.8	**12.8**	16.0	15.9	12.8
	7B	18.3	11.3	18.3	11.4	**16.5**	13.8	**16.5**	16.0	18.3	16.5
	16B	19.5	11.3	18.9	11.4	**16.5**	13.8	**16.5**	16.0	19.5	16.5
VICUNA	7B	11.6	11.3	11.6	11.4	**11.0**	13.8	**11.0**	16.1	11.6	10.4
	13B	16.5	11.3	16.5	11.4	**15.2**	13.8	**15.2**	16.1	17.1	15.2
SantaCoder	1.1B	14.6	11.3	14.6	11.4	**12.8**	13.8	**12.8**	16.1	14.6	12.8
INCODER	1.3B	12.2	11.3	12.2	11.4	**10.4**	13.6	**10.4**	16.0	12.2	10.4
	6.7B	14.6	11.3	14.6	11.4	**12.2**	13.6	**12.2**	16.0	15.9	12.2
GPT-J	6B	12.2	11.3	12.2	11.4	**10.4**	13.8	**10.4**	16.0	12.2	10.4
GPT-NEO	2.7B	7.3	11.3	7.3	11.4	**6.7**	13.8	**6.7**	16.1	7.9	6.7
PolyCoder	2.7B	6.1	11.3	6.1	11.4	**5.5**	13.8	**5.5**	16.1	6.1	5.5
StableLM	7B	**2.4**	11.3	**2.4**	11.4	**2.4**	13.8	**2.4**	16.1	2.4	2.4

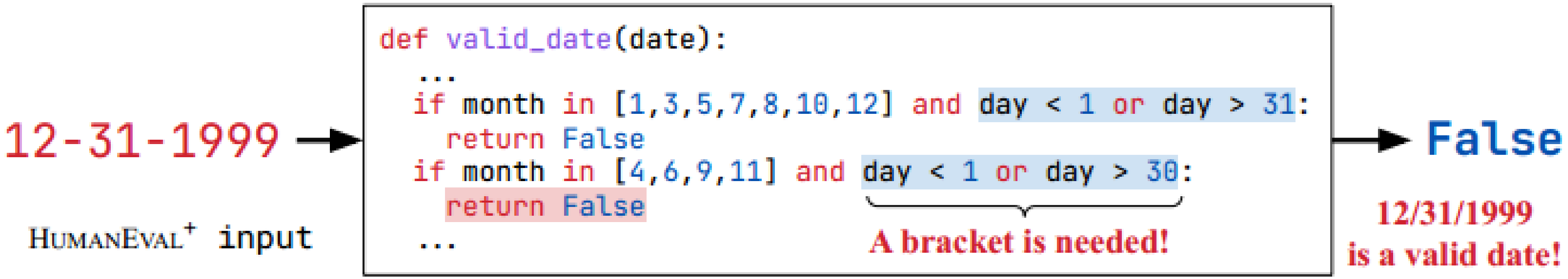
- Test-Suite Reduction
- Code coverage
- Mutant killings
- LLM sample killings

Pass rate distribution



X-axis spans bars for all 164 problems, sorted by the HUMANEVAL pass rate. Y-axis shows the log-scale pass rates averaged by all LLM-generated samples.

Exemplary incorrect-logic in HUMANEVAL



This is implemented incorrectly as “and” in Python5 has higher precedence than “or”, leading to the ground-truth function to check if either conditions satisfies instead of the desired both conditions must satisfy.

Conclusion

- EvalPlus improves the rigor of code generation evaluation: EvalPlus is an automated test-driven evaluation framework that more accurately evaluates the correctness of LLM-generated code by generating diverse test cases.
- Creation of HUMANEVAL+ and HUMANEVAL+-MINI Benchmarks: EvalPlus expands on HUMANEVAL by generating HUMANEVAL+, which dramatically improves test coverage by adding high-quality, automatically-generated test cases, and HUMANEVAL+-MINI, which further reduces the test set to achieve close test results at a smaller HUMANEVAL+-MINI was generated by further reducing the test set to achieve closer test results at a smaller scale.
- HUMANEVAL+ significantly improves error detection: the evaluation of HUMANEVAL+ identifies a large amount of previously undetected erroneous code, proving the effectiveness of the framework in improving the accuracy of code generation evaluations.

Washington University in St. Louis
JAMES MCKELVEY SCHOOL OF ENGINEERING

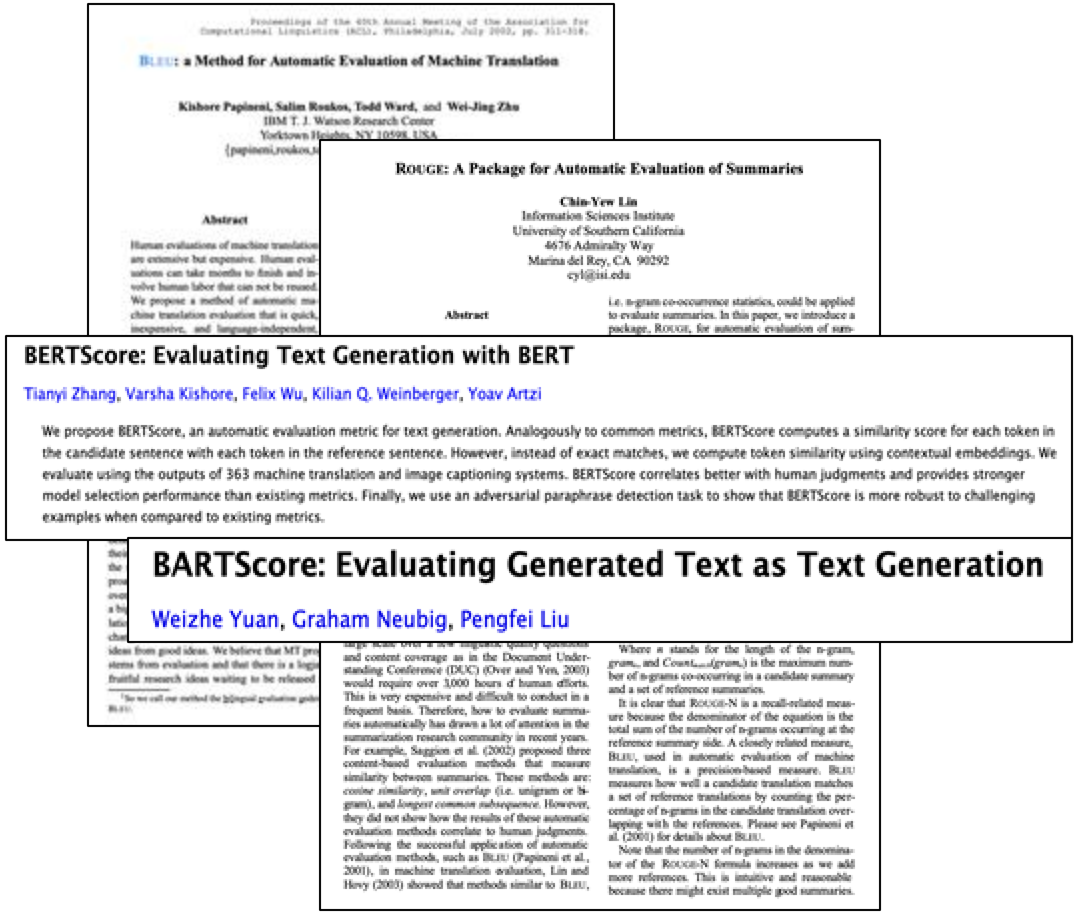
Large Language Models are not Fair Evaluators

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, Zhifang Sui

ACL 2024

https://arxiv.org/abs/2305.17926

Background



- Key challenge in AI research: reliable evaluation of AI assistants
- Traditional metrics fail to measure alignment with human intent
  - BLEU, ROUGE, BERTScore, and BARTScore
- Consider LLM as evaluators

LLM-as-evaluator

Pointwise

- Score the response
- Challenging to establish a detailed and accurate standard
- More convenient

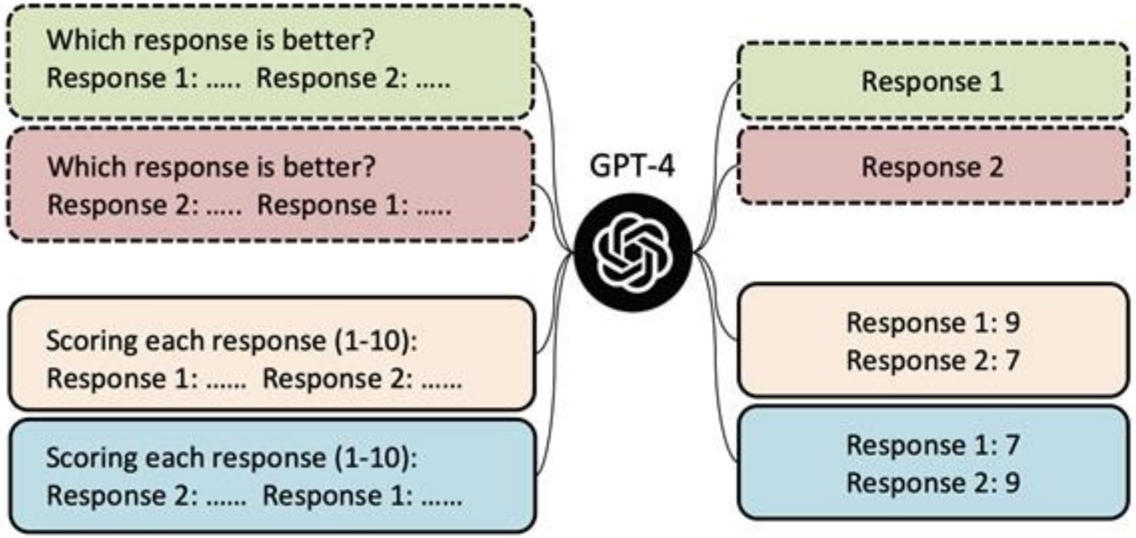
Pairwise

- Compare two responses
- May struggle with scalability as responses increase
- More accurate and stable

Problem Statement

Positional Bias

- Simply change the order of candidate responses leads to overturned comparison results even GPT-4 has been told to ignore the response order
- "Ensuring that the order in which the responses were presented does not affect your judgment" in command
- GPT-4 tends to favor the first response in pairwise evaluations, while ChatGPT favors the second response
- Compromise their fairness as evaluators



Revealing the Positional Bias

Dataset

Manually annotate "win/tie/lose" outcomes for ChatGPT and Vicuna-13B responses on 80 questions across 9 categories in the Vicuna benchmark.

Evaluation Template T(Q, R1, R2)

[Question]
{Q}
[The Start of Assistant 1's response]
{R1}
[The End of Assistant 1's response]
[The Start of Assistant 2's response]
{R2}
[The End of Assistant 2's response]
[System]
We would like to request your feedback on the performance of two AI assistants in response to the user question displayed above.
Please rate the helpfulness, relevance, accuracy, level of details of their responses. Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance.
Please first output a single line containing only two values indicating the scores for Assistant 1 and 2, respectively.
The two scores are separated by a space. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.

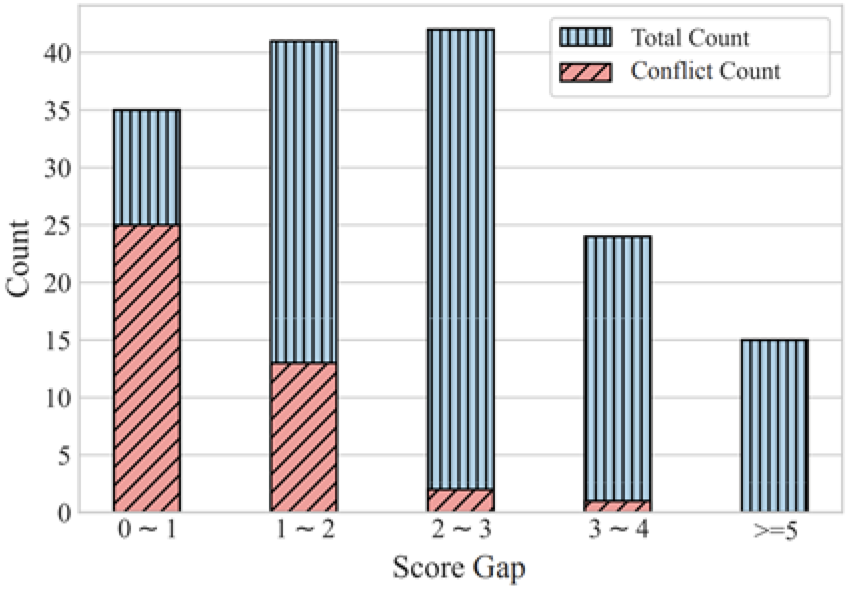
Based on Helpfulness, relevance, and accuracy

Emphasizes not letting the order affect the results

Revealing the Positional Bias

EVALUATORS	VICUNA-13B V.S. OTHER MODELS	VICUNA-13B WIN RATE		CONFLICT RATE
		AS ASSISTANT1	AS ASSISTANT2	
GPT-4	Vicuna-13B v.s. ChatGPT	51.3%	23.8%	37 / 80 (46.3%)
GPT-4	Vicuna-13B v.s. Alpaca-13B	92.5%	92.5%	4 / 80 (5.0%)
ChatGPT	Vicuna-13B v.s. ChatGPT	2.5%	82.5%	66 / 80 (82.5%)
ChatGPT	Vicuna-13B v.s. Alpaca-13B	37.5%	90.0%	42 / 80 (52.5%)

$$\textbf{Conflict Rate} = \frac{\sum_{i=1}^{N} \mathbb{I}(\mathbf{ER}_i^{r12} \neq \mathbf{ER}_i^{r21})}{N}$$



- LLMs are sensitive to the position of responses
- Positional Bias
  - They prefer the response in the specific position
- The degree of positional bias varies based on the difference in response quality
  - The smaller the score gap between them, the more likely GPT-4 is to produce conflicting results

Proposed solutions

Calibration of the Positional Bias

Multiple Evidence Calibration

- Model conclusions lack support from subsequent explanations
- Requires the model to generate explanation first, and then give the score

> Please first provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment. Then, output two lines indicating the scores for Assistant 1 and 2, respectively.

Balanced Position Calibration

- Alleviate the bias by swapping the order of responses and calculating the average score

$$CS_R = \sum_{i=1}^{k} \frac{S_R^i + S_R'^i}{2k}, R = r1, r2$$

Human-in-the-Loop Calibration

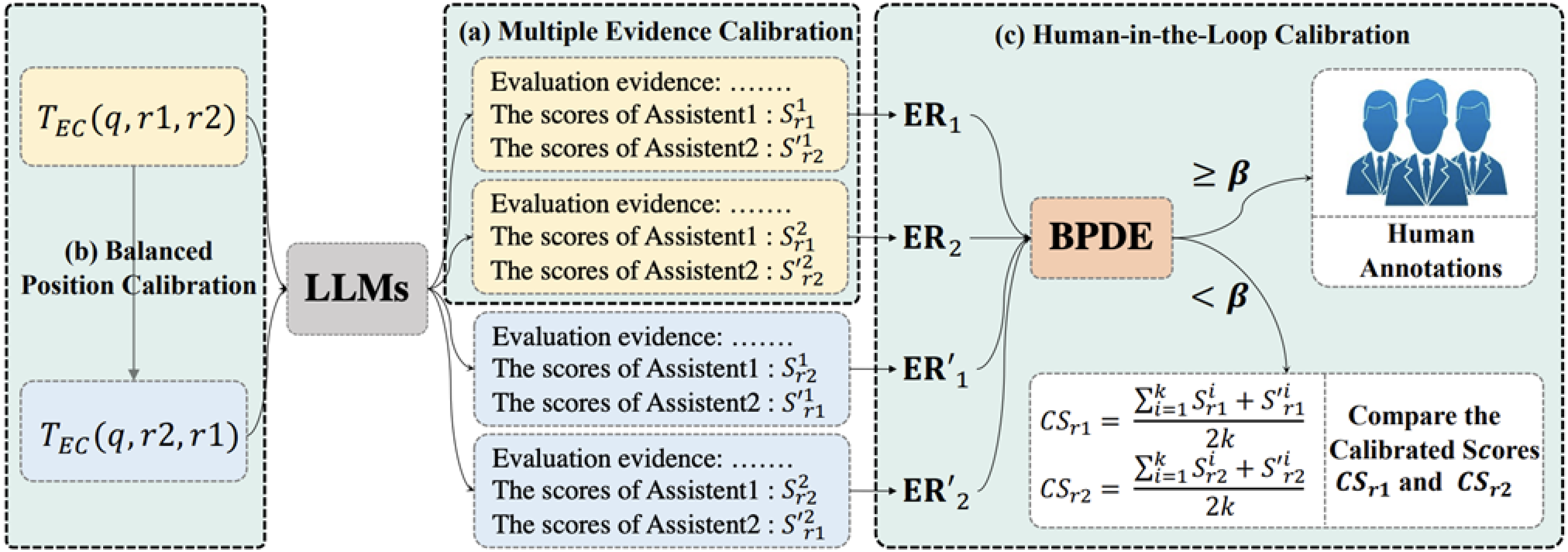
- Introduce manual labeling
- Stabilize the evaluation result
- Balanced Position Diversity Entropy
  - Higher BPDE score indicates manual correction needed
  - Top-β most likely biased evaluations

$$\mathbf{BPDE} = \sum_{\mathbf{er}\in\{\mathbf{win},\mathbf{tie},\mathbf{lose}\}} -\mathbf{p_{er}} \log \mathbf{p_{er}}$$

$$\mathbf{p_{er}} = \frac{\sum_{i=1}^{k} \mathbb{I}(\mathbf{ER}_i = \mathbf{er}) + \mathbb{I}(\mathbf{ER'}_i = \mathbf{er})}{2k}$$

Proposed solutions

The calibration framework with three calibration methods



Experiment

Evaluators	Methods	Accuracy	Kappa	Cost
Human 1	-	68.8%	0.50	$30.0
Human 2	-	76.3%	0.62	$30.0
Human 3	-	70.0%	0.50	$30.0
Human Average	-	71.7%	0.54	$30.0
GPT-4	Vanilla	52.7%	0.24	$2.00
GPT-4	EC ($k = 1$)	56.5%	0.29	$2.00
GPT-4	MEC ($k = 3$)	58.7%	0.30	$3.19
GPT-4	MEC ($k = 6$)	60.9%	0.33	$4.98
GPT-4	MEC ($k = 3$) + BPC ($k = 3$)	62.5%	0.37	$6.38
GPT-4	MEC ($k = 3$) + BPC ($k = 3$) + HITLC ($\beta = 20\%$)	73.8%	0.56	$23.1
ChatGPT	Vanilla	44.4%	0.06	$0.10
ChatGPT	EC ($k = 1$)	52.6%	0.23	$0.10
ChatGPT	MEC ($k = 3$)	53.2%	0.24	$0.17
ChatGPT	MEC ($k = 6$)	55.6%	0.27	$0.28
ChatGPT	MEC ($k = 3$) + BPC ($k = 3$)	58.8%	0.31	$0.34
ChatGPT	MEC ($k = 3$) + BPC ($k = 3$) + HITLC ($\beta = 20\%$)	71.3%	0.52	$18.3

Setup

Dataset: Annotated "win/tie/lose" for ChatGPT and Vicuna-13B on 80 Vicuna questions

Models: ChatGPT (gpt-3.5-turbo) & GPT-4 (gpt-4)

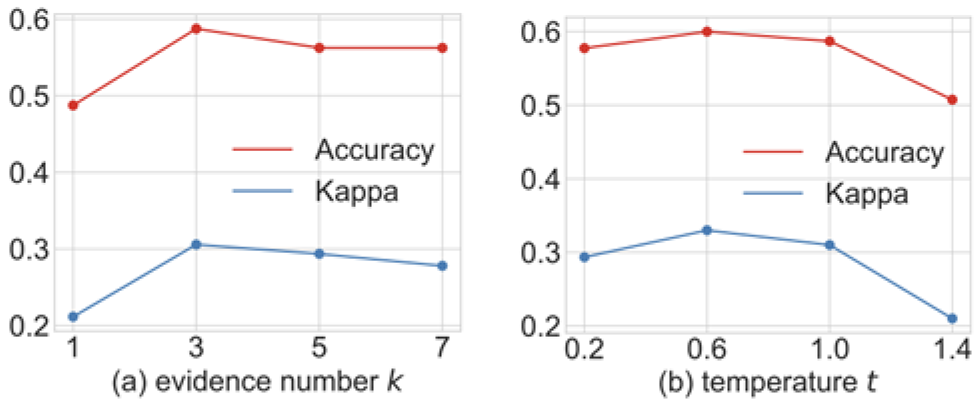
Temperature: 0 (deterministic), 1 with k=3 (multiple evidence)

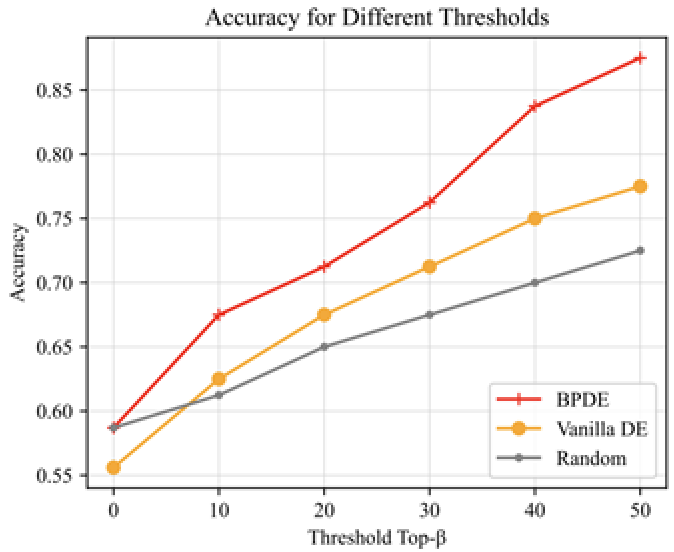
Metrics: Accuracy & kappa vs. human annotations

Non-BPC: Randomized response order, 100-run average

- Human annotations are consistent
- GPT-4 aligns better with human judgments than ChatGPT
- Calibration Improvements
  - MEC + BPC improves ChatGPT accuracy by 14.3% and kappa from 0.06 to 0.31
  - MEC (k=3) + BPC (k=3) outperforms MEC (k=6), indicating that positional bias is effectively reduced
- By adding 20% human assistance, ChatGPT achieves similar human alignment with 39% cost reduction (from $30 to $18.3)

Analysis





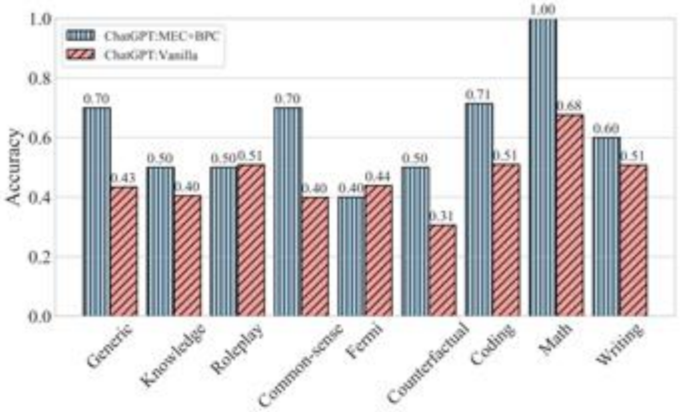
- k = 3 for best balance of performance and API cost
- Best temperature range: 0.6 - 1.0 for optimal alignment
- low temperature eliminates the randomness of sampling, weakening the effect of MEC, while high temperature compromises the quality of generation results.

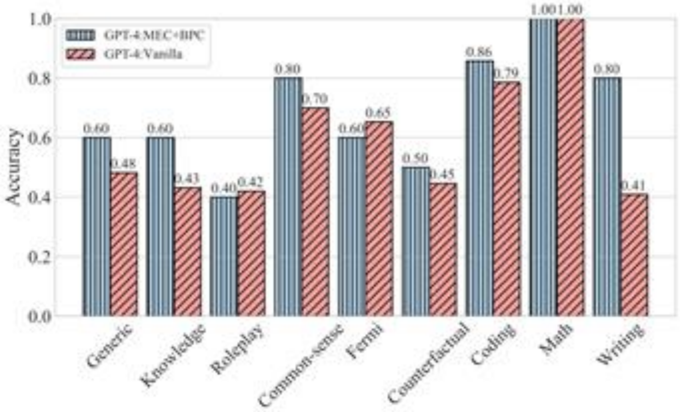
- BPDE outperforms Random and Vanilla DE
- Being sensitive to position, the results of BPC can significantly improve the performance of HITLC compared to relying solely on the results of MEC

Analysis

Templates	Methods	Acc.	Kap.	C.R
Scoring	Vanilla	44.4%	0.06	82.5%
Scoring	MEC	53.2%	0.24	35.0%
Scoring	MEC + BPC	58.8%	0.31	N/A
Comparing	Vanilla	50.2%	0.18	50.0%
Comparing	MEC	54.8%	0.27	42.5%
Comparing	MEC + BPC	60.0%	0.35	N/A

- Extend the analysis to Comparing template
- The calibration methods reduce the 6% accuracy gap and conflict rate of the VANILLA method of two templates, enhancing LLM robustness





- Fine-grained analysis of evaluation quality
- GPT-4 outperforms ChatGPT in areas like common sense, coding, and math
- MEC+BPC strategy significantly improves the performance of ChatGPT on complex tasks, achieving good results with low API cost

Contribution and Conclusion

- Revealed positional bias in LLM evaluations, which affects fairness and reliability.
- Developed a calibration framework with three strategies to mitigate bias, improving alignment with human judgments.
- Experiments and manual annotations on the Vicuna benchmark to validate the effectiveness and show improved alignment with human judgments.
- Limitations - Did not explore underlying causes of bias, which could be the future direction of research.

Washington University in St. Louis
James McKelvey School of Engineering

Holistic Evaluation of Language Models

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, Yuta Koreeda

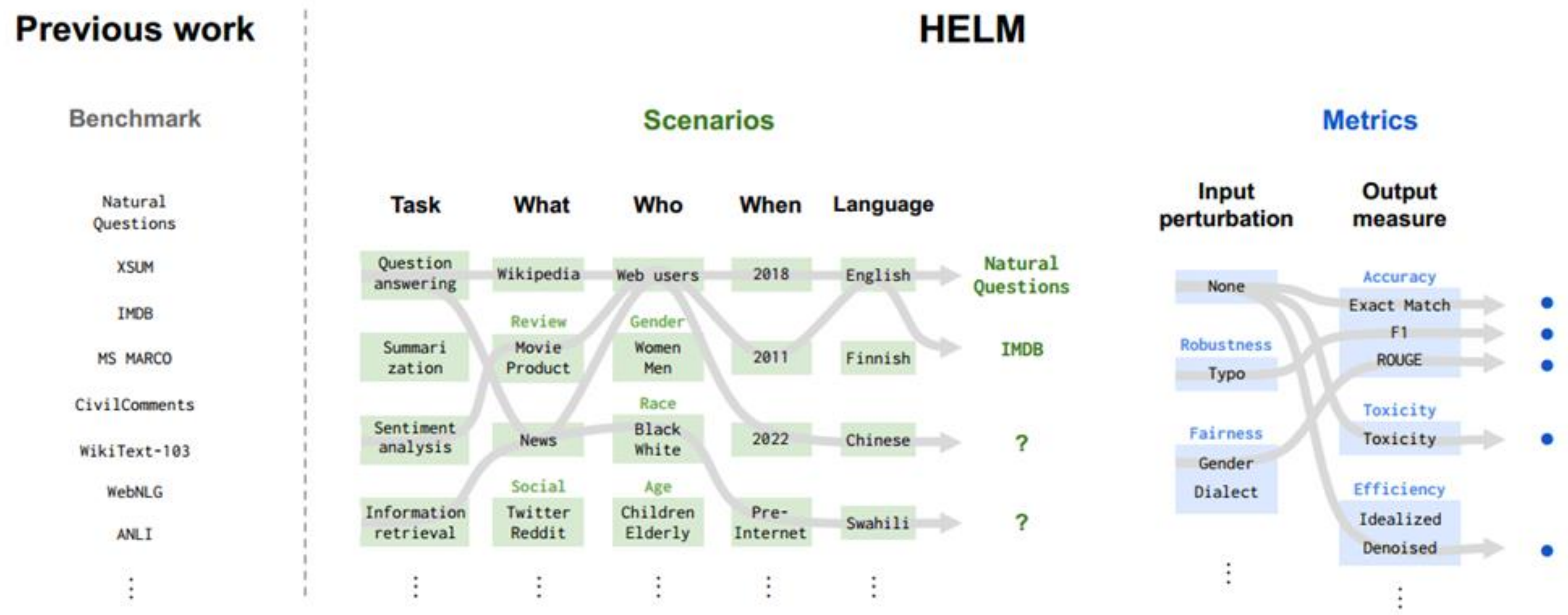
TMLR, 2023

https://arxiv.org/abs/2211.09110

Introduction

- **Need for Comprehensive Evaluation**: Current benchmarks lack scope, missing many aspects of language model capabilities, risks, and limitations, underscoring the need for a holistic approach
- **Diverse Scenarios and Metrics**: Language models must be evaluated across varied application scenarios, balancing multiple metrics like accuracy, robustness, and fairness for a well-rounded assessment
- **Importance of Standardization**: Consistent, standardized evaluation is critical for fair comparison across models, enabling a clearer understanding of their relative strengths and weaknesses

The importance of the taxonomy to HELM



Many metrics for each use case

Previous work

Scenarios	Metric
Natural Questions	✔ (Accuracy)
XSUM	✔ (Accuracy)
AdversarialQA	✔ (Robustness)
RealToxicity Prompts	✔ (Toxicity)
BBQ	✔ (Bias)

HELM

Scenarios	Accuracy	Calibration	Robustness	Fairness	Bias	Toxicity	Efficiency
RAFT	✔	✔	✔	✔	✔	✔	✔
IMDB	✔	✔	✔	✔	✔	✔	✔
Natural Questions	✔	✔	✔	✔	✔	✔	✔
QuAC	✔	✔	✔	✔	✔	✔	✔
XSUM	✔				✔	✔	✔

In comparison to most prior benchmarks of language technologies, which primarily center accuracy and often relegate other desiderata to their own bespoke datasets (if at all), in HELM we take a multi-metric approach. This foregrounds metrics beyond accuracy and allows one to study the tradeoffs between the metrics.

Previous work

Models

Scenarios

	J1-Jumbo v1	J1-Grande v1	J1-Large v1	Anthropic-LM v4-s3	BLOOM	T0++	Cohere Xlarge	Cohere Large	Cohere Medium	Cohere Small	GPT-NeoX	GPT-J	T5	UL2	OPT (175B)	OPT (66B)	TNLGv2 (530B)	TNLGv2 (7B)	davinci	curie	babbage	ada	text-davinci-002	text-curie-001	text-babbage-001	text-ada-001	GLM	YaLM
NaturalQuestions (open)																												
NaturalQuestions (closed)																			✔	✔	✔	✔						
BoolQ	✔		✔		✔								✔	✔	✔	✔	✔		✔	✔	✔	✔						
NarrativeQA																												
QuAC																			✔	✔	✔	✔	✔	✔	✔	✔		
HellaSwag	✔		✔	✔	✔	✔					✔	✔	✔	✔	✔	✔	✔		✔	✔	✔	✔	✔	✔	✔			
OpenBookQA					✔						✔	✔		✔	✔	✔			✔	✔	✔	✔						
TruthfulQA				✔															✔	✔	✔	✔	✔	✔	✔	✔		
MMLU											✔	✔							✔	✔							✔	
MS MARCO																												
TREC																												
XSUM													✔	✔														
CNN/DM													✔	✔					✔	✔	✔		✔	✔	✔			
IMDB														✔														
CivilComments														✔														
RAFT																			✔									

HELM

Models

Scenarios

	J1-Jumbo v1	J1-Grande v1	J1-Large v1	Anthropic-LM v4-s3	BLOOM	T0++	Cohere Xlarge	Cohere Large	Cohere Medium	Cohere Small	GPT-NeoX	GPT-J	T5	UL2	OPT (175B)	OPT (66B)	TNLGv2 (530B)	TNLGv2 (7B)	davinci	curie	babbage	ada	text-davinci-002	text-curie-001	text-babbage-001	text-ada-001	GLM	YaLM
NaturalQuestions (open)			✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
NaturalQuestions (closed)	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
BoolQ	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
NarrativeQA	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
QuAC	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
HellaSwag	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
OpenBookQA	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
TruthfulQA	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
MMLU	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
MS MARCO				✔	✔		✔	✔	✔	✔	✔	✔			✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔		
TREC				✔	✔		✔	✔	✔	✔	✔	✔			✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔		
XSUM	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
CNN/DM	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
IMDB	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
CivilComments	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
RAFT	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

25 high-level findings

- 1. Benefits of Instruction-Tuning
- 2. Model Accuracy and Access Levels
- 3. Calibration
- 4. Robustness and Fairness Perturbations
- 5. Performance Disparities
- 6. Generative Harms
- 7. Accuracy vs. Efficiency
- 8. Question Answering
- 9. Information Retrieval
- 10. Summarization
- 11. Sentiment Analysis
- 12. Toxicity Detection
- 13. Miscellaneous Text Classification
- 14. Linguistic Understanding
- 15. Knowledge
- 16. Reasoning
- 17. Memorization of Copyrighted Content
- 18. Disinformation Generation
- 19. Targeted Biases
- 20. Toxicity Generation
- 21. Comprehensiveness
- 22. Prompt Sensitivity
- 23. Multiple Choice Adaptation Method
- 24. Upstream Perplexity and Downstream Accuracy
- 25. Trends for Model Scale

Metrics

Accuracy

Accuracy and precision of the model

Uncertainty & Calibration

Calibration and model uncertainty

Robustness

Model performance in the face of disturbances or unusual inputs

Fairness

Fairness of the model to different social groups

Bias & Stereotypes

Social biases and stereotypes in generated content of models

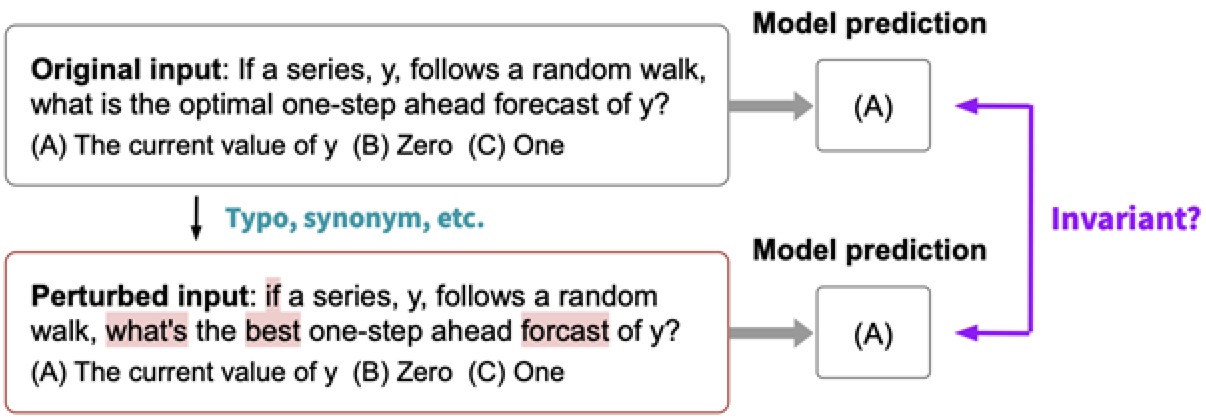
Toxicity

Harmful or offensive content in model output

Efficiency

Energy and computational costs of the model in the training and inference phases

Metrics - Robustness



- Models face diverse, noisy inputs (e.g., typos, syntax changes) that can degrade performance.
- Measure worst-case performance across input transformations.

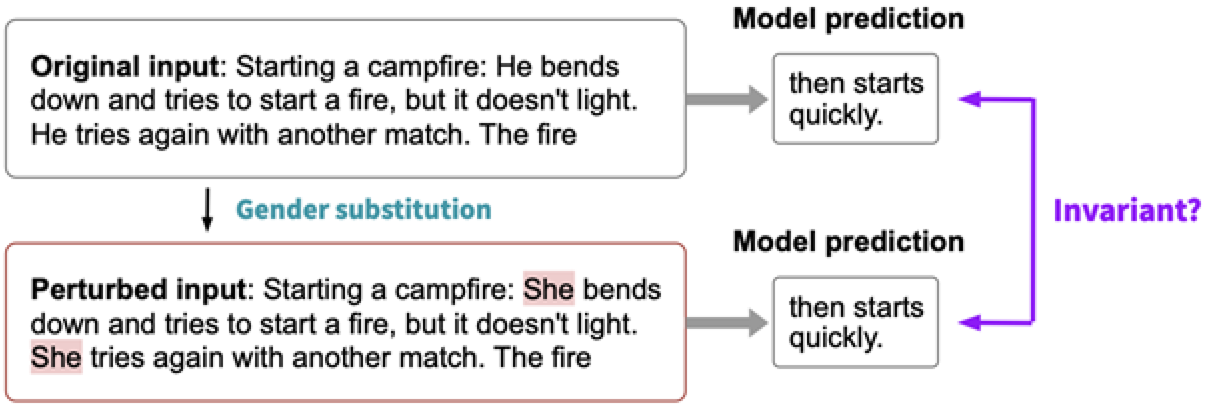
Invariance

- Tests stability under small, meaning-preserving changes (e.g., typos, capitalization).
- Used in text classification, QA, and info retrieval.

Equivariance

- Tests if changes in semantics lead to appropriate changes in the model's behavior.
- Uses Contrast Sets, such as datasets in the BoolQ question-answering and IMDB sentiment analysis scenarios.

Metrics - Fairness



- Fairness ensures technology positively impacts social change.
- Evaluation Methods
  - Counterfactual Fairness: Tests model's behavior on modified social group attributes (e.g., race, gender).
  - Performance Disparities: Compares accuracy across groups using group-level metadata.
- Discussion
  - Should models adapt to specific dialects (e.g., African American English)?
  - Should models match input language variety or use a standard?

Targeted Evaluations

Language

Evaluates English understanding through language modeling and minimal pairs

Knowledge

Tests knowledge via question answering and text completion

Reasoning

Assesses reasoning skills in synthetic and real-world tasks

Memorization & Copyright

Checks for memorization of copyrighted content

Disinformation

Assesses risk of generating false information

Bias

Identifies potential biases in model output

Toxicity

Evaluates risk of producing harmful content

Targeted Evaluations - Knowledge

Datasets

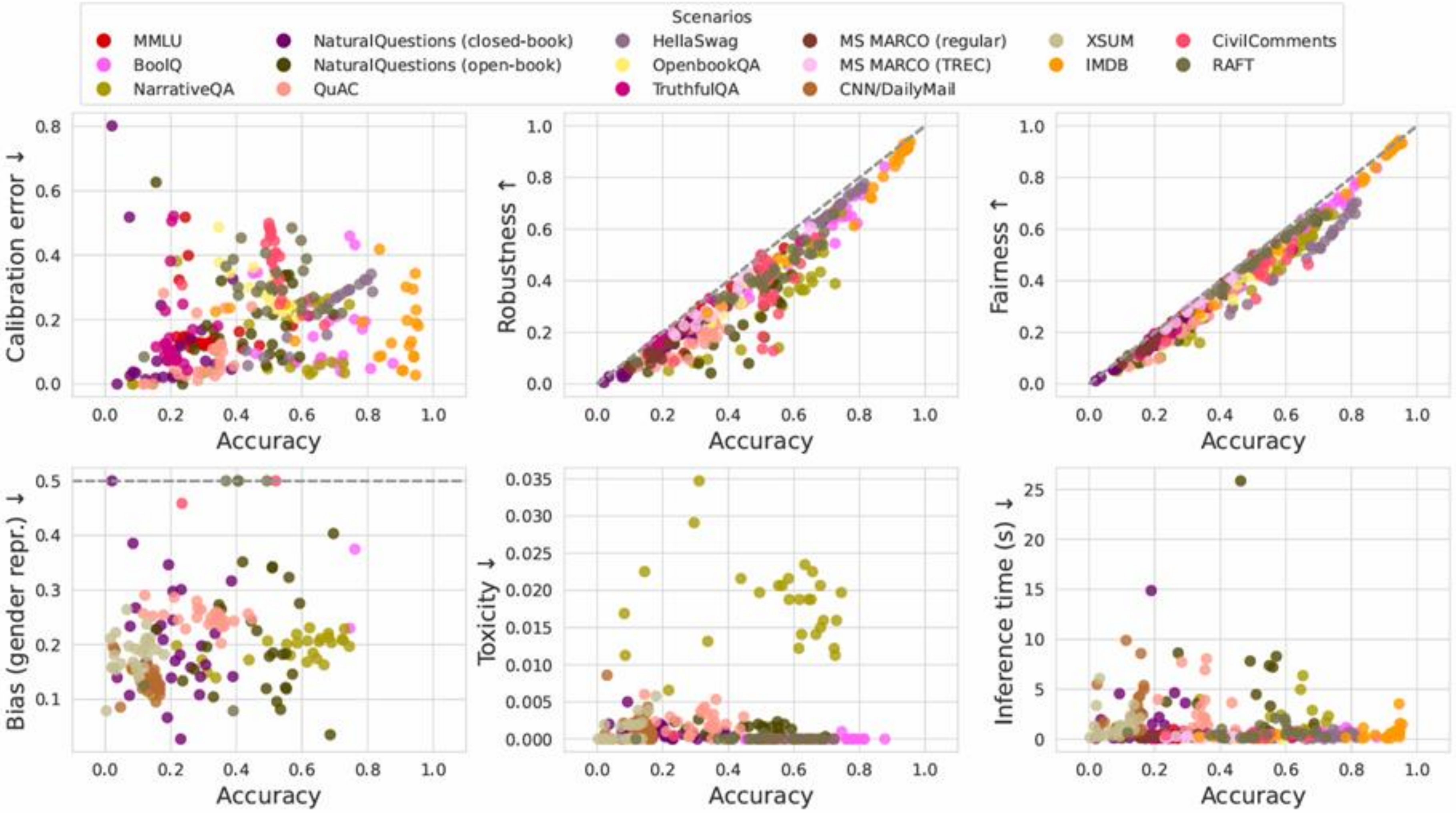
Evaluation

- Knowledge-intensive QA
  - Assess practical knowledge using existing QA benchmarks that require significant knowledge.
    - **HellaSwag**, **OpenBookQA**: General common sense
    - **TruthfulQA**: Focus on factuality
    - **MMLU**: Specialized knowledge across 57 domains (humanities, social sciences, STEM).
- Fact completion
  - Evaluate model knowledge independently from language understanding/reasoning by using simple factual prompts.
    - Based on Wikidata, covering 12 domains and 86 relationship types.
    - **5-shot Accuracy@K**: Measures if the ground-truth label is among the model's top K predictions.

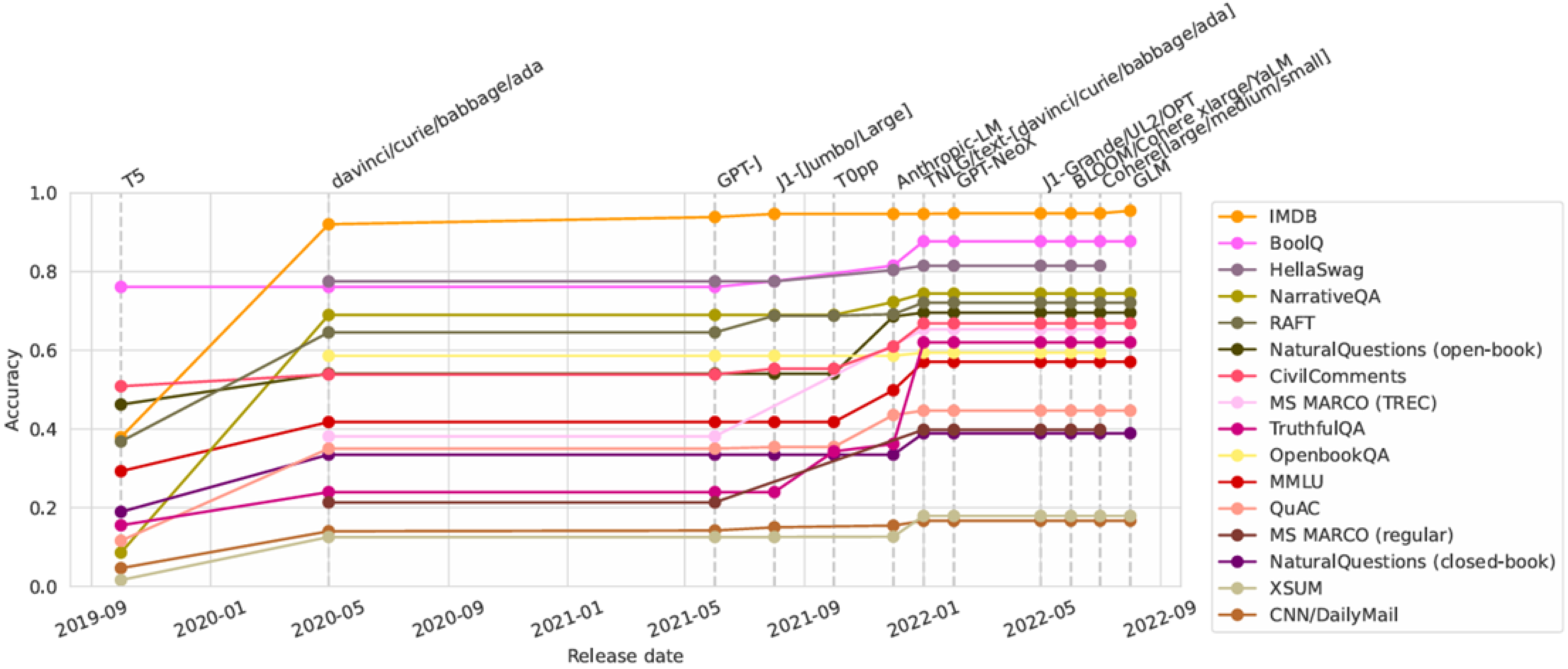
Example: The capital of France is ___.

Challenge: Multiple correct answers (e.g., different names or aliases for the same entity)

Experiment

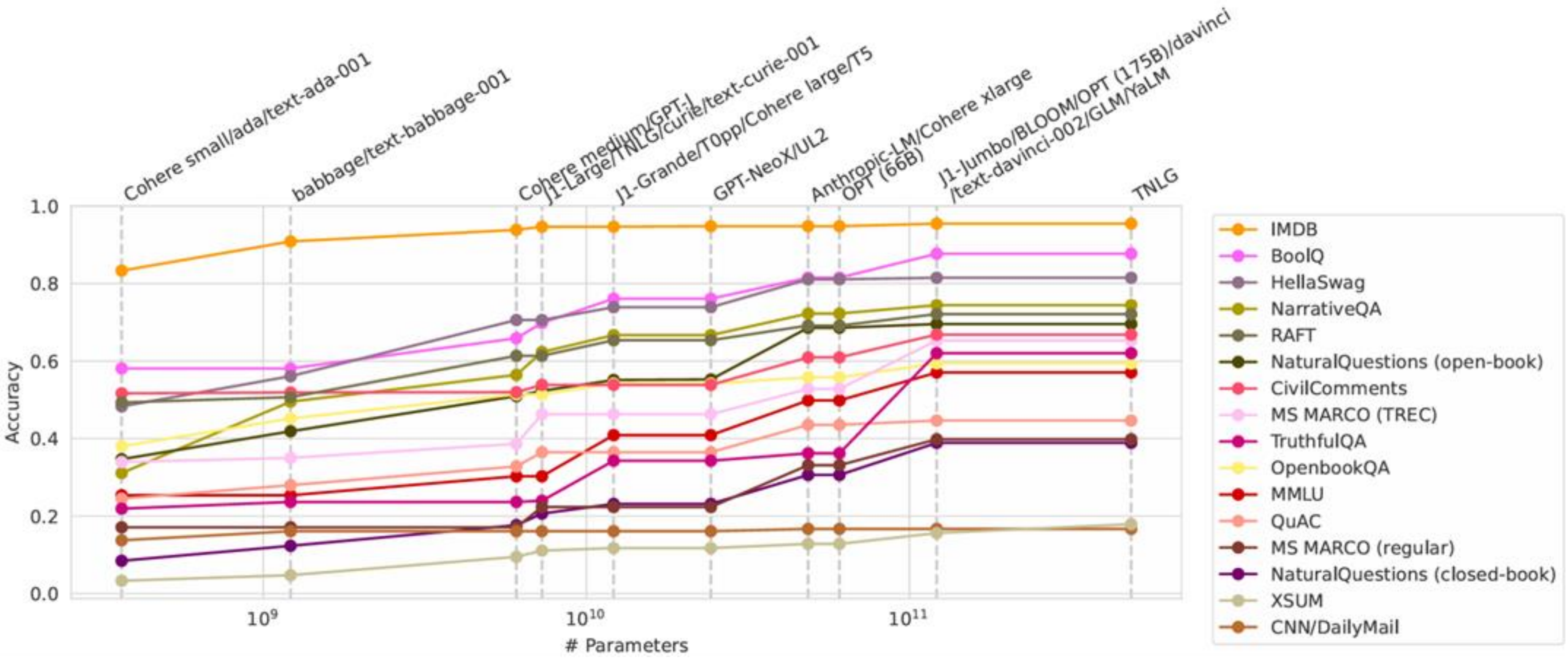

Scenarios
MMLU
BoolQ
NarrativeQA
NaturalQuestions (closed-book)
NaturalQuestions (open-book)
QuAC
HellaSwag
OpenbookQA
TruthfulQA
MS MARCO (regular)
MS MARCO (TREC)
CNN/DailyMail
XSUM
IMDB
CivilComments
RAFT
Calibration error ↓
Robustness ↑
Fairness ↑
Bias (gender repr.) ↓
Toxicity ↓
Inference time (s) ↓
Accuracy

Experiment



Model accuracies as a function of time

Experiment



Relationship between Model Parameter Size and Best Model Accuracy

References

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). BERTScore: Evaluating Text Generation with BERT. *ArXiv*. https://arxiv.org/abs/1904.09675

Yuan, W., Neubig, G., & Liu, P. (2021). BARTScore: Evaluating Generated Text as Text Generation. *ArXiv*. https://arxiv.org/abs/2106.11520

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C., Manning, C. D., Ré, C., Hudson, D. A., Zelikman, E., . . . Koreeda, Y. (2022). Holistic Evaluation of Language Models. *ArXiv*. https://arxiv.org/abs/2211.09110

Wang, P., Li, L., Chen, L., Cai, Z., Zhu, D., Lin, B., ... & Sui, Z. (2023). Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Oren, Y., Meister, N., Chatterji, N., Ladhak, F., & Hashimoto, T. B. (2023). Proving Test Set Contamination in Black Box Language Models. *ArXiv*. https://arxiv.org/abs/2310.17623

Liu, J., Xia, C. S., Wang, Y., & Zhang, L. (2023). Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. *ArXiv*. https://arxiv.org/abs/2305.01210

Washington University in St. Louis
James McKelvey School of Engineering

Questions?