BenchMarker: An Education-Inspired Toolkit for Highlighting Flaws in Multiple-Choice Benchmarks

Nishant Balepur[1,2] **Bhavya Rajasekaran**[1] **Jane Oh**[1] **Michael Xie**[1] **Atrey Desai**[1]
Vipul Gupta[3] **Steven Moore**[4] **Eunsol Choi**[2] **Rachel Rudinger**[1] **Jordan Boyd-Graber**[1]

[1]University of Maryland [2]New York University [3]Scale AI [4]George Mason University

nbalepur@umd.edu jbg@umiacs.umd.edu

Abstract

Multiple-choice question answering (MCQA) is standard in NLP, but benchmarks lack rigorous quality control. We present BenchMarker, an education-inspired toolkit using LLM judges to flag three common MCQ flaws: 1) *contamination*—items appearing exactly online; 2) *shortcuts*—cues in the choices that enable guessing; and 3) *writing errors*—structural/grammatical issues based on a 19-rule education rubric. We validate BenchMarker with human annotations, then run the tool to audit 12 benchmarks, revealing: 1) flaws persist in MCQA benchmarks, especially automatically-made and crowdsourced data—we detect 47% of TruthfulQA appears online and 100% of HellaSwag violates multiple writing rules; 2) contaminated MCQs tend to inflate accuracy, while writing errors tend to lower it and change rankings beyond random; and 3) prior benchmark repairs address their targeted issues (i.e., lowering accuracy with LLM-written distractors), but inadvertently add new flaws (i.e. implausible distractors, many correct answers). Overall, flaws in MCQs degrade NLP evaluation, but education research offers a path forward. We release BenchMarker to bridge the fields and improve MCQA benchmark design.[1]

1 Grading Benchmarks like Educators

Progress in NLP relies on benchmarks (Voorhees, 2001) that are largely multiple-choice question answering (MCQA) tasks, where models must pick the answer to a question from input choices (Robinson and Wingate, 2023). MCQs test whether models can understand the question, recall related facts, and use said knowledge to deduce the answer (Richardson et al., 2013), but only do so reliably when error-free. MCQ flaws add noise unrelated to these skills, undermining their construct validity (Smith, 2005).

MCQA datasets are notoriously flawed: MCQs exist in LLM training data (Deng et al., 2024), have exploitable shortcuts (Gupta et al., 2025), and are rife with writing issues (Palta et al., 2024, e.g., poor grammar). *Educators* detect, cull, and remedy these errors when writing MCQs to ensure students rarely face them (Campbell, 2011), but while NLP *researchers* value MCQA for its similarity to human testing (Zhuang et al., 2025), they rarely adopt education's standards: for 39 surveyed MCQA datasets, 23% report no quality control (Appendix A.1).

To address this, prior work has devoted extensive resources to "correct" MCQA benchmarks (Wang et al., 2024c; Chizhov et al., 2025). However, they do not propose reusable metrics to check whether rewritten MCQA benchmarks truly reduce errors.

We present **BenchMarker**, a toolkit for detecting MCQ errors based on three metrics educators value (Fig 1): *1) contamination*—whether MCQs appear online, a proxy for dataset leakage, similar to ensuring students cannot cheat on exams (Taylor et al., 2020, §2.1); *2) shortcuts*—whether MCQs have shallow cues that let strong LLMs answer without the question, like students guessing via partial knowledge (Lau et al., 2011, §2.2); and *3) writing errors*—grammar and structure violations on a 19-rule rubric of educational MCQ writing guidelines (Costello et al., 2018b, §2.3). We distil these insights from decades of education research via NLP advances in LLM-as-a-judge (Zheng et al., 2023), validated across 23 models, six web search APIs, 13 MCQA datasets, and 8042 expert judgments (§3).

BenchMarker audits 12 MCQA datasets and predicts pervasive issues, especially with MCQs from NLP annotation protocols, versus exams that educators design: 47% of TruthfulQA appears online, 21% of ScholarIQA has shortcuts, and HellaSwag always violates at least two writing rules (§4.1). Educational theory informs that these flaws degrade MCQA (Cronbach and Meehl, 1955), so we empirically assess their impact on LLM evaluation. LLMs have higher accuracy on contaminated data splits and lower accuracy on splits with multiple writing

[1]Our code and data are available at: https://github.com/nbalepur/BenchMarker

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15793–15824
July 2-7, 2026

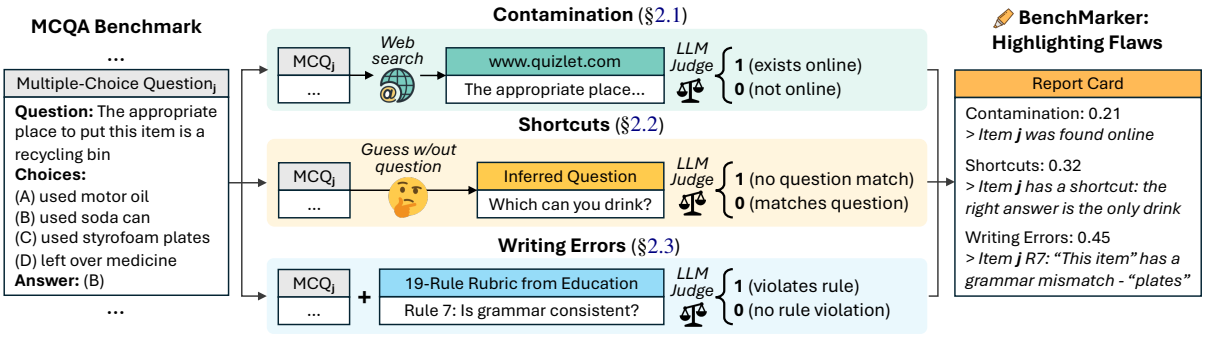

Figure 1: **BenchMarker** scores MCQA benchmark items across three axes with LLM judges: 1) *contamination*—whether the item appears on the Internet; 2) *shortcuts*—whether models can use shallow shortcuts in choices to solve the item without the question; and 3) *writing errors*—grammatical and structural issues based on a 19-rule rubric derived from education research. We aggregate scores to audit datasets and return judge feedback on items.

errors; discarding MCQs with writing errors shifts LLM rankings beyond random (§4.3), muddying how researchers select models to train and deploy. The writing errors from NLP and educational assessments overlap—unclear wording and distractor quality—motivating these fields to collaborate and jointly advance evaluations and student testing (§4.4). Lastly, while past benchmark fixes resolve targeted issues, they can add new ones: MMLU-Pro uses LLM-written distractors to lower accuracy, but this produces faulty distractors and more than one correct answer (§4.5). Thus, MCQ correction requires iterative refinement, which BenchMarker can support via repeatable, automated scoring runs.

The quality of MCQA benchmarks is often overlooked, but this erodes evaluation validity. Education research can guide the design quality control tools like BenchMarker and offer lessons to NLP from student testing, so we champion more collaborations between these fields. We contribute:

1. BenchMarker, a toolkit to predict contamination, shortcuts, and writing flaws in MCQs. We wrap it in InspectAI, a library with judge logs, standard prompts, and a UI to track runs (Appendix A.2).

2. A rigorous audit of 12 NLP benchmarks, showing MCQA flaws are pervasive and impact LLM accuracy and rankings, especially writing errors on crowdsourced and automatically written MCQs.

3. In-depth analysis on common writing flaws in NLP's and education's MCQs, and evidence that prior solutions do not fully address these issues.

4. A validation dataset of 8042 human judgments to test LLM judge reliability in MCQ flaw detection.

2 BenchMarker: Marking MCQA Flaws

A multiple-choice question (MCQ) has a question stem q and choices $\mathcal{C}$. Test-takers must pick the best answer $a \in \mathcal{C}$ and avoid distractors $\mathcal{D} = \mathcal{C} \setminus \{a\}$. MCQs assess if test-takers can understand questions, recall facts, and deduce choice correctness (Simkin and Kuechler, 2005), but issues in MCQA datasets undermine this goal (Chizhov et al., 2025). We thus introduce **BenchMarker** (Figure 1), which builds upon education research to detect benchmark flaws.

We design BenchMarker to detect flaws *within* MCQs, leaving global issues like saturation (Vania et al., 2021) and diversity (Uzunoglu et al., 2025) as future work. For our blueprint, we review MCQA work in education and NLP (Haladyna et al., 2002; Balepur et al., 2025c), informing us of three key issues: contamination (§2.1), shortcuts (§2.2), and writing errors (§2.3). We use LLM judges to predict each issue, later shown to agree with humans (§3).

2.1 Contamination Detection

Educators typically write MCQs that students have not seen before, testing the transfer of learning beyond rote memorization (Roediger and Butler, 2011). Equivalently, NLP ensures that training and testing dataset splits do not overlap (Larson, 1931), assessing model generalization over memorization.

To score whether NLP systems can easily memorize an MCQ, we check whether it exists online—a practice educators use (Taylor et al., 2020). The internet is a reasonable proxy for pre-training[2] (Li et al., 2024b)—such data is not fully disclosed but likely on the web (Soldaini et al., 2024)—and tests whether often-benchmarked retrieval-augmented

[2]Appendix A.9 compares to other contamination methods.

LLMs (Bragg et al., 2025) can cheat by finding the MCQ and echoing its answer (Paleka et al., 2025).

We adapt Li et al. (2024b)'s web search check: 1) query a web API with question stem q and gold answer a as input; and 2) prompt an LLM to judge whether the item is exactly/nearly-exactly in search results, meaning it is contaminated. We do not flag MCQs when the web pages have only the *knowledge* linked to the answer (i.e. no explicit MCQA format), since this would not lead to *exact* memorization.

2.2 Shortcuts Detection

MCQA is easy to score, as test-takers choose from a list, but this can enable informed guessing (Frary, 1988): partial knowledge students can exploit shortcuts in choices via meta-strategies to answer MCQs (Lau et al., 2011, e.g., pick the "odd one out", pattern matching), subverting the MCQ's validity. Likewise, NLP models reach above-random MCQA accuracy when answering using choices alone (Richardson and Sabharwal, 2020), spurring concerns that MCQA benchmarks have superficial cues that fail to assess true model abilities (Chandak et al., 2025).

A standard way to identify shortcuts is *partial-input studies* (Poliak et al., 2018): isolating MCQs that LLMs can solve with just the choices. Choices-only accuracy is often equated with shortcuts (Chandak et al., 2025), but this can be problematic or benign. For example, in Figure 1 (middle), when answering with the choices alone, the LLM can guess which choice is the "odd one out" via shallow cues (*"Soda" is the only beverage in the list, so it must be right*), or more strategically infer the original question (*Styrofoam plates, motor oil, and medicine are non-degradable, so the question likely links to recycling and "soda can" is right*). The former is a concerning shortcut that lets models bypass the MCQ, undermining benchmark validity, but the latter is less problematic, since the model effectively answers the original MCQ as intended.

To draw this line and see if models can shortcut the MCQ in problematic ways, we prompt LLMs to "think step by step" (Wei et al., 2022) to answer with the choices and then guess the original question, following Balepur et al. (2024b). An LLM judge then evaluates whether the inferred question is semantically equivalent to the true question. To limit model variance when answering without the question, we take a majority vote of three LLMs with high choices-only accuracy from Balepur et al. (2025a): GPT-5, Gemini 2.5 Pro, and Claude 4.5 Sonnet.[3] We predict the MCQ as having shortcuts if: 1) LLMs solve the MCQ with just choices; and 2) the inferred question does not match the original, indicating that the MCQ has exploitable shortcuts.

2.3 Writing Error Detection

MCQs lose validity if authors add structural, semantic, and grammatical errors while writing MCQs, as they render the item misleading or unanswerable (Haladyna et al., 2002). Analogously, if NLP models fail here, we cannot discern whether they lacked skills MCQA tests or misinterpreted poor writing.

Educators have long recognized such errors and have thus curated rubrics to rigorously assess MCQs (Haladyna and Downing, 1989). Inspired by this, we combine them with LLM judges (Hashemi et al., 2024), which can follow rules to score outputs with high human agreement (Kim et al., 2024). We take the 19-rule Item-Writing Flaws rubric from Tarrant et al. (2006), which avoids subjective rules in others (Haladyna and Downing, 1989, e.g., "*avoid trivial material*") and has been used for over two decades across domains in education—mostly higher education (Schmucker and Moore, 2025). We prompt LLMs to judge which rules an MCQ breaks, via 19 prompts with each flaw name, definition, and six examples (three flawed MCQs, three flawless MCQs).

The 19 rules ensure MCQs are clear (e.g., no ambiguous terms like "mostly", state questions clearly/concisely in the stem), adhere to MCQA's format (e.g., have one right answer, make distractors plausible), curb giveaways (e.g., the stem and choices must be grammatically consistent), and are not misleading (e.g., sort numerical options, avoid "none of the above"); Table 8 has the full list. Some rules may not always apply—work uses longer question stems to test long-context abilities (Dua et al., 2019) or "none of the above" for abstention (Elhady et al., 2025)—but we use all 19, since our audited benchmarks are more general (§4). We eventually reveal the most pervasive issues are clarity and distractor quality (§4.4), which apply to all MCQA datasets.

3 Validating BenchMarker's Reliability

Before using BenchMarker, we ensure it reliably predicts flaws. This section outlines datasets (§3.1), annotations (§3.2), and baselines (§3.3) to confirm BenchMarker agrees with human judgments (§3.4).

[3] We report per-model shortcut detection in Appendix A.10.

Dataset	Domain	Difficulty	Creation Strategy
Algebra Question Answering (Zhong et al., 2024, AQuA)	Math	Graduate	Student Exams
AI2 Reasoning Challenge (Clark et al., 2018, ARC)	Science	Elementary	Student Exams
CommonsenseQA (Talmor et al., 2019, CQA)	Commonsense	General	Crowdworkers
HellaSwag (Zellers et al., 2019, HS)	Commonsense	General	Model Generated
Multitask Language Understanding (Hendrycks et al., 2021, MMLU)	Multi-Subject	College	Student Exams
Open Book Question Answering (Mihaylov et al., 2018, OBQA)	Science	Elementary	Crowdworkers
Physical Interaction Question Answering (Bisk et al., 2020, PIQA)	Commonsense	General	Crowdworkers
Question Answering via Sentence Composition (Khot et al., 2020, QASC)	Science	Elementary	Crowdworkers
Scholastic Aptitude Test (Zhong et al., 2024, SAT)	Math	High School	Student Exams
Social Intelligence Question Answering (Sap et al., 2019, SIQA)	Commonsense	General	Crowdworkers
Super Google Proof Question Answering (Du et al., 2025, SGPQA)	Multi-Subject	Graduate	Expert+Model
Truthful Question Answering (Lin et al., 2021, TQA)	Commonsense	General	Author-Written

Table 1: The MCQA benchmarks we mainly explore. We audit MCQs across varied domains, difficulties, and creation strategies.

3.1 Collecting MCQs for Human Review

We collect MCQs to assess BenchMarker from 12 MCQA datasets of varied domains, difficulties, and creation strategies for dataset design (Table 1). We could randomly sample MCQs, but this might omit rare flaws. We instead run BenchMarker via GPT-5 as the judge on each benchmark's training set, then sample with stratification (Cochran, 1977; Zouhar et al., 2025): for each metric, we sample up to ten MCQs per dataset where GPT predicts the item is flawed and up to ten GPT predicts are not flawed. This sampling better surfaces positive and negative cases, ensuring we have both labels for each metric.

BenchMarker is tailored for NLP datasets, but our scores parallel what educators value when writing MCQs (Haladyna et al., 2002), so we also test how well LLMs judge MCQs designed for students. We use Costello et al. (2018a)'s 4123 labels on the 19 writing rules from §2.3 on higher education exams in computer science, humanities, health sciences, psychology and math. This yields in-domain (for NLP) and out-of-domain (for students) data for writing error detection. In total, we gather 229 MCQs to label for contamination, 271 to label for shortcuts, 3419 to label for writing errors on NLP data, and 4123 human-written MCQs with existing writing error labels, resulting in 8042 entries for validation.

3.2 Human Annotations on MCQs

We now collect human annotations on our MCQs to validate judges. For contamination, annotators look for the MCQ across four search engines (Google, Bing, DuckDuckGo, and Brave), labeling it as 1 (flawed) if it exists exactly in any web page, 0 (not flawed) otherwise. For shortcuts, annotators compare GPT's inferred question from just the choices to the original question stem. They label the item as 1 if the two do not semantically match—not testing the same concepts—following Balepur et al. (2024b), and 0 otherwise. For writing errors, annotators see a rule from the 19-item rubric and mark whether the MCQ violates it via its definition and examples, following Moore et al. (2023a).

We do not use crowdworkers to rate MCQ quality, since Moore et al. (2023a) show they have 25% disagreement with experts. Thus, we use a protocol based on qualitative coding in HCI (Bingham, 2023): Author A—an English-speaking CS graduate researcher with papers and annotation experience in MCQA—labels each MCQ, and Author B with the same background labels 50 random items for each metric (50 labels for each writing rule, for $19 \cdot 50 = 950$ in all) to evaluate reliability. They have >80% agreement per metric (Appendix A.5).

3.3 Baseline and Metric Selection

We now assess how well LLM judge predictions ($\hat{l}$) match humans (l) on our validation set. For shortcut/writing errors, we use 23 open/closed LLMs across seven families: 1) Gemini 2.5 (Comanici et al., 2025, Lite, Flash, Pro); 2) GPT-5 (OpenAI, 2025, Nano, Mini, Base); 3) Claude 4.5 (Anthropic, 2025, Haiku, Sonnet); 4) Command (Gomez, 2024, R, R+); 5) Qwen-3 (Yang et al., 2025, 0.6, 1.7, 4, 8, 14, 32B); 6) Gemma-3 (Team et al., 2025, 4, 12, 7B); and 7) LLaMA-3 (Dubey et al., 2024, 1, 3, 8, 70B). We prompt LLMs with default parameters and request a JSON with a prediction and explanation.

We also assess the Scalable Automatic Question Usability Evaluation Toolkit (Moore et al., 2024, SAQUET) for writing errors, which uses heuristics and GPT-5 judges to detect the 19 writing errors we study, but optimized on the out-of-domain validation set of student exams (§3.1). Contamination detection relies on web search quality (§2.1), so we only use our three best LLMs (Gemini Pro, GPT-5,

	Shortcuts			*Writing (In Domain, NLP)*			*Writing (Out of Domain, Human)*		
Method	Accuracy	F1 Score	Cohen's κ	Accuracy	F1 Score	Cohen's κ	Accuracy	F1 Score	Cohen's κ
Gemini 2.5 Lite	0.76	0.74	0.51	0.68	0.56	0.35	0.71	0.28	0.18
Gemini 2.5 Flash	0.69	0.68	0.41	0.79∗	0.62	0.47	0.85	0.38	0.31
Gemini 2.5 Pro	0.70	0.69	0.43	**0.82**∗	**0.66**	**0.53**	0.86	0.39	0.33
GPT-5 Nano	0.68	0.65	0.38	0.74	0.39	0.22	0.88	0.21	0.14
GPT-5 Mini	0.77	0.72	0.53	0.75	0.55	0.38	0.84	0.32	0.25
GPT-5	**0.82**∗	**0.75**	**0.61**	0.81 ∗	0.63	0.50	0.87	0.37	0.30
Claude 4.5 Haiku	0.78∗	0.73	0.55	0.72	0.58	0.38	0.76	0.31	0.22
Claude 4.5 Sonnet	0.81∗	0.75	0.59	0.79∗	0.63	0.48	0.83	0.36	0.28
Command R	0.74	0.72	0.50	0.77	0.53	0.38	0.89	0.37	0.31
Command R+	0.72	0.70	0.46	0.76	0.53	0.36	0.88	0.36	0.30
Qwen-3 0.6B	0.71	0.64	0.39	0.73	0.05	0.00	0.90	0.06	0.02
Qwen-3 1.7B	0.62	0.64	0.31	0.76	0.25	0.16	0.91	0.16	0.12
Qwen-3 4B	0.42	0.55	0.05	0.73	0.49	0.32	0.88	0.32	0.26
Qwen-3 8B	0.57	0.61	0.24	0.70	0.54	0.33	0.79	0.29	0.20
Qwen-3 14B	0.61	0.64	0.30	0.71	0.55	0.34	0.80	0.32	0.23
Qwen-3 32B	0.73	0.71	0.48	0.71	0.55	0.35	0.80	0.33	0.25
Gemma-3 4B	0.50	0.59	0.16	0.63	0.46	0.20	0.74	0.25	0.15
Gemma-3 12B	0.61	0.64	0.31	0.75	0.04	0.03	0.92	0.11	0.09
Gemma-3 27B	0.74	0.72	0.50	0.75	0.03	0.02	0.93	0.03	0.02
LLaMA-3.2 1B	0.56	0.30	0.00	0.47	0.38	0.03	0.42	0.14	0.01
LLaMA-3.2 3B	0.58	0.62	0.26	0.68	0.37	0.16	0.81	0.19	0.09
LLaMA-3.1 8B	0.49	0.58	0.14	0.71	0.38	0.19	0.86	0.22	0.14
LLaMA-3.1 70B	0.61	0.62	0.28	0.64	0.47	0.22	0.73	0.23	0.12
SAQUET	—	—	—	0.78	0.40	0.28	**0.93**	**0.52**	**0.48**
Random (50/50)	0.50	0.42	0.00	0.50	0.34	0.00	0.50	0.13	0.00
Always Not Flawed	0.64	0.00	0.00	0.74	0.00	0.00	0.93	0.00	0.00
Always Flawed	0.37	0.54	0.00	0.26	0.41	0.00	0.08	0.14	0.00

Table 2: Human–judge agreement for shortcuts and writing flaw detection. * on accuracy means the method has significantly better predictions than all trivial baselines (McNemar, 1947, Mcnemar's test, $p < 0.05$ with Bonferroni correction). Appendix A.6 has results grouped by each of the 19 writing flaws. The most reliable LLM judges are **bold**, informing BenchMarker's design.

Method	Accuracy	F1 Score	Cohen's κ
Google + Gemini Pro	0.70*	0.65	0.41
Google + GPT-5	**0.71***	**0.68**	**0.44**
Google + Claude Sonnet	0.69*	0.64	0.40
Brave + GPT-5	0.54	0.28	0.14
Perplexity + GPT-5	0.64	0.53	0.31
Exa + GPT-5	0.59	0.43	0.22
Tavily + GPT-5	0.58	0.41	0.20
Serper + GPT-5	0.68	0.65	0.36
Random (50/50)	0.50	0.52	0.00
Always Not Flawed	0.46	0.00	0.00
Always Flawed	0.54	0.70	0.00

Table 3: Human–judge agreement for contamination detection. * means the method is significantly better than trivial baselines (McNemar, 1947, Mcnemar's test, $p < 0.05$, Bonferroni correction). Appendix A.7 shows all LLM/API combinations. Google with GPT-5 is the most reliable LLM judge.

Sonnet), and focus on testing six web search APIs: Google, Bing, Perplexity, Exa, Tavily, and Serper.

We report *accuracy* (the proportion where $\hat{l} = l$), *F1 Score* (harmonic mean of precision/recall, measuring how well $\hat{l}$ predicts l with class imbalance), and *Cohen's* κ (Cohen, 1960, non-random agreement of $\hat{l}$ and l). To interpret scores, we add trivial baselines for each flaw type: randomly (50/50) predict 0 (not flawed) or 1 (flawed), always predict 0, and always predict 1. Beating these baselines gives confidence that judges informatively detect flaws.

3.4 BenchMarker Agrees with Human Judges

We now evaluate how well judges agree with humans. In contamination, Google is the best search API, and GPT-5 surpasses Gemini and Claude at using its web pages (Table 3). F1 is similar to the "Always Flawed" baseline, but accuracy/Cohen's κ are much higher, and Google+GPT-5's low F1 mainly stems from low recall; precision is 0.86. Manual analysis shows Google's API gives a subset of public search engine results—our annotations flag contaminated MCQs APIs may not surface—so we still deem our contamination detection strong.

In shortcuts/writing errors, many LLMs beat trivial baselines (Table 2); the best models are closed-source (GPT-5, Gemini Pro), but open-weight ones compete (Command R, Qwen 32B), so future work can study tuning smaller LLMs to close this gap for efficiency/reproducibility. GPT-5 has higher mean F1/Cohen's κ than SAQUET on in-domain and out-of-domain data, making BenchMarker state-of-the-art at detecting writing flaws versus existing tools. BenchMarker's competitive scores on educator-written exams suggest it could also help educators.

We want BenchMarker to be useful across MCQA datasets, so we also report its generalization. For the best LLMs in Tables 2 and 3, accuracy's standard deviation is 0.07 for shortcuts, 0.15 for con-

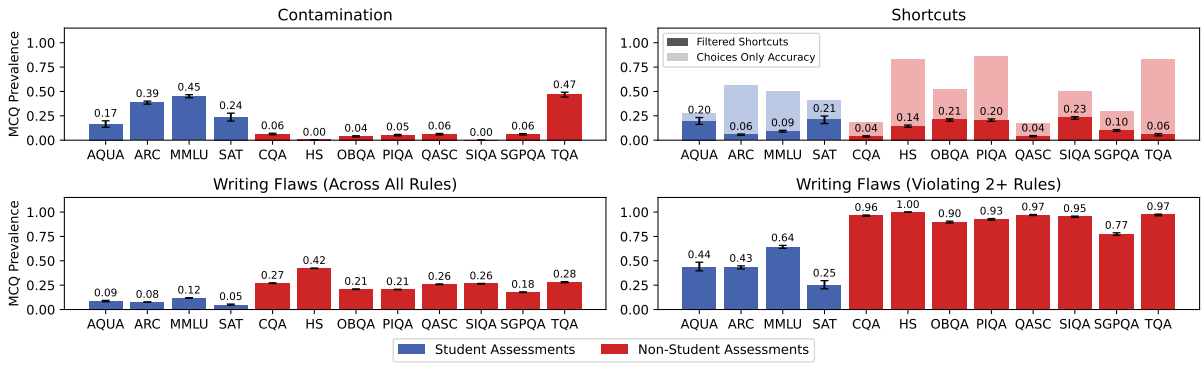

Figure 2: Prevalence of flaws in MCQA benchmarks, grouped by whether the MCQs originate from student assessments. While MCQs from exam-based benchmarks are more commonly found online (top left), they contain far fewer writing flaws (bottom).

	Contamination			*Shortcuts*			*2+ Writing Flaws*			*Any Flaw*		
Dataset	Flaw	No Flaw	ΔAcc	Flaw	No Flaw	ΔAcc	Flaw	No Flaw	ΔAcc	Flaw	No Flaw	ΔAcc
AQUA	0.74 ± 0.07	0.76 ± 0.02	+2.89	0.75 ± 0.05	0.76 ± 0.02	+0.65	0.68 ± 0.03	0.81 ± 0.02	+19.34	0.72 ± 0.03	0.81 ± 0.02	+11.52
ARC	0.90 ± 0.01	0.87 ± 0.01	-3.31	0.85 ± 0.02	0.88 ± 0.00	+4.03	0.86 ± 0.01	0.89 ± 0.01	+3.62	0.88 ± 0.01	0.88 ± 0.01	+0.08
CQA	0.74 ± 0.04	0.78 ± 0.01	+4.71	0.70 ± 0.05	0.78 ± 0.01	+11.49	0.77 ± 0.01	0.89 ± 0.03	+14.60	0.77 ± 0.01	0.88 ± 0.04	+14.12
HS	0.87 ± 0.03	0.78 ± 0.01	-9.84	0.74 ± 0.02	0.79 ± 0.01	+6.75	0.78 ± 0.01	—	—	0.78 ± 0.01	—	—
MMLU	0.85 ± 0.01	0.75 ± 0.01	-11.20	0.76 ± 0.03	0.80 ± 0.01	+4.95	0.77 ± 0.01	0.83 ± 0.01	+7.83	0.79 ± 0.01	0.79 ± 0.02	-0.83
OBQA	0.86 ± 0.04	0.87 ± 0.01	+1.69	0.86 ± 0.01	0.87 ± 0.01	+1.45	0.86 ± 0.01	0.92 ± 0.01	+6.33	0.87 ± 0.01	0.92 ± 0.01	+6.71
PIQA	0.90 ± 0.03	0.91 ± 0.01	+0.73	0.86 ± 0.01	0.92 ± 0.01	+6.16	0.90 ± 0.01	0.93 ± 0.01	+3.40	0.90 ± 0.01	0.94 ± 0.01	+3.95
QASC	0.59 ± 0.05	0.60 ± 0.01	+2.96	0.55 ± 0.05	0.61 ± 0.01	+9.33	0.60 ± 0.01	0.70 ± 0.06	+17.20	0.60 ± 0.01	0.66 ± 0.07	+9.00
SAT	0.74 ± 0.04	0.79 ± 0.01	+6.65	0.77 ± 0.04	0.78 ± 0.02	+0.99	0.74 ± 0.05	0.79 ± 0.01	+6.71	0.77 ± 0.03	0.80 ± 0.01	+3.91
SIQA	1.00 ± 0.00	0.80 ± 0.01	-19.86	0.80 ± 0.02	0.80 ± 0.01	+0.12	0.79 ± 0.01	0.97 ± 0.01	+22.42	0.79 ± 0.01	0.97 ± 0.01	+22.74
SGPQA	0.47 ± 0.03	0.48 ± 0.01	+2.49	0.53 ± 0.02	0.47 ± 0.01	-9.71	0.45 ± 0.01	0.57 ± 0.01	+25.85	0.46 ± 0.01	0.57 ± 0.01	+24.02
TQA	0.76 ± 0.02	0.78 ± 0.02	+2.47	0.60 ± 0.06	0.78 ± 0.01	+31.52	0.77 ± 0.01	0.93 ± 0.03	+21.53	0.77 ± 0.01	0.94 ± 0.04	+21.95
Micro μ	0.81 ± 0.01	0.76 ± 0.00	-6.90	0.77 ± 0.01	0.77 ± 0.00	-1.03	0.75 ± 0.00	0.83 ± 0.00	+9.83	0.76 ± 0.00	0.80 ± 0.01	+5.39
Macro μ	0.78 ± 0.04	0.76 ± 0.03	-2.48	0.73 ± 0.03	0.77 ± 0.04	+5.35	0.75 ± 0.04	0.84 ± 0.04	+12.17	0.76 ± 0.04	0.83 ± 0.04	+9.61

Table 4: LLM accuracy on flawed and not flawed MCQA dataset splits across each flaw type. Micro/macro averages show that contaminated splits tend to have higher accuracy, and splits with two or more writing errors tend to have lower accuracy.

tamination, and 0.06 for writing errors; we later reveal writing errors are crucial for NLP to fix (§4.2) and encouragingly, standard deviation is low.

3.5 Recommendation: LLMs in BenchMarker

Overall, LLMs predict MCQ contamination, shortcuts, and each of the 19 writing errors with Cohen's κ and accuracy matching standard LLM judge protocols (Zheng et al., 2023; Bavaresco et al., 2025). When using BenchMarker for the rest of our analyses (§4), we use the LLMs with the best Cohen's κ in Tables 2 and 3 for the three flaw types. For writing errors, we pick the LLM with the best Cohen's κ for each of the 19 errors types (Appendix A.4).

Most of BenchMarker's cost comes from writing flaw detection, which uses 19 LLM judges per item. For researchers with limited resources, Gemini-2.5 Pro is likely too expensive; we recommend Gemini-2.5 Flash, which is only 0.05 below Gemini-2.5 Pro in Cohen's κ and is $\sim \frac{1}{4}$ of the cost.[4] We provide a more detailed cost breakdown in Appendix A.11.

Finally, for researchers without access to closed-source LLMs, we recommend Cohere Command-R for writing flaw detection; it has the best Cohen's κ out of all open-weight LLMs. In Appendix A.12, we run follow-up experiments on judge ensembling, confidence calibration, and writing flaw detection failures to further support this recommendation.

4 A Report Card for MCQA Benchmarks

Having validated BenchMarker, we now use it to audit benchmarks (§4.1) and study how flaws impact evaluation (§4.2, §4.3). We reveal writing errors are rife (§4.4), unaffected by prior fixes (§4.5).

4.1 MCQA Benchmarks are Rife with Flaws

We run BenchMarker on up to 1000 sampled MCQs from the test sets of 12 benchmarks in Table 1, predicting contamination, shortcut, and writing errors. All datasets have flaws (Figure 2): we detect 47% of TruthfulQA exists online, 23% of SocialIQA has shortcuts, and on average, each item in HellaSwag violates 44% of the 19 writing rules. In education, MCQs violating 2+ writing rules are "unacceptable" (Tarrant et al., 2006), but we suspect 7/12 datasets

[4] https://ai.google.dev/gemini-api/docs/pricing

	Contamination			*Shortcuts*			*2+ Writing Errors*			*Any Flaw*		
Model	All	No Flaw	Random	All	No Flaw	Random	All	No Flaw	Random	All	No Flaw	Random
Gemini-2.5 Lite	0.530 (10)	0.525 (10)	0.530 (10)	0.530 (10)	0.525 (10)	0.520 (10)	0.530 (10)	0.542 (10)	0.489 (10)	0.530 (10)	0.505 (10)	0.469 (10)
Gemini-2.5 Flash	0.562 (9)	0.560 (9)	0.568 (9)	0.562 (9)	0.559 (9)	0.554 (9)	0.562 (9)	0.585 (9)	0.526 (9)	0.562 (9)	0.557 (9)	0.514 (9)
Gemini-2.5 Pro	0.865 (3)	0.857 (3)	0.856 (3)	0.865 (3)	0.862 (3)	0.862 (3)	0.865 (3)	0.961 (1)	0.909 (2)	0.865 (3)	0.955 (1)	0.894 (2)
GPT-5 Nano	0.825 (6)	0.815 (6)	0.816 (6)	0.825 (6)	0.823 (6)	0.822 (6)	0.825 (6)	0.927 (5)	0.868 (6)	0.825 (6)	0.911 (5)	0.850 (6)
GPT-5 Mini	0.858 (4)	0.848 (4)	0.849 (4)	0.858 (4)	0.858 (4)	0.855 (4)	0.858 (4)	0.953 (3)	0.899 (4)	0.858 (4)	0.942 (3)	0.883 (4)
GPT-5	0.878 (1)	0.870 (1)	0.871 (1)	0.878 (1)	0.877 (1)	0.875 (1)	0.878 (1)	0.958 (2)	0.916 (1)	0.878 (1)	0.950 (2)	0.902 (1)
Claude 4.5 Haiku	0.832 (5)	0.821 (5)	0.823 (5)	0.832 (5)	0.829 (5)	0.828 (5)	0.832 (5)	0.919 (6)	0.872 (5)	0.832 (5)	0.898 (6)	0.852 (5)
Claude 4.5 Sonnet	0.874 (2)	0.863 (2)	0.864 (2)	0.874 (2)	0.872 (2)	0.871 (2)	0.874 (2)	0.937 (4)	0.908 (3)	0.874 (2)	0.916 (4)	0.890 (3)
Command R	0.699 (8)	0.686 (8)	0.694 (8)	0.699 (8)	0.699 (8)	0.694 (8)	0.699 (8)	0.730 (8)	0.705 (8)	0.699 (8)	0.676 (8)	0.678 (8)
Command R+	0.720 (7)	0.706 (7)	0.712 (7)	0.720 (7)	0.719 (7)	0.716 (7)	0.720 (7)	0.766 (7)	0.738 (7)	0.720 (7)	0.719 (7)	0.708 (7)
Spearman's ρ	—	1.000	1.000	—	1.000	1.000		0.927	0.986	—	0.927	0.978

Table 5: LLM rank (in parentheses) correlation between full vs no flaw/random splits. Removing contamination and shortcuts does not shift rankings, but filtering writing errors or any flaw does beyond random, confirmed via permutation tests ($\alpha = 0.01$).

have over 90% of items with 2+ writing violations.

Grouping items by their origin shows those from educator-written student exams (blue, not hatched) have fewer writing flaws than those written automatically or by crowdworkers (red, hatched), showing education's value in MCQA design. Out of the non-educator MCQs, SGPQA has the fewest writing flaws; experts wrote them with LLMs, so human-AI writing is a promising path to improve MCQs. Despite this benefit, educator-written MCQs are more contaminated; annotation (§3.2) found many items online as study aids (e.g., flashcards, tutor sites), so LLM developers could filter websites linked to test preparation to stop this (Soldaini et al., 2024). One may expect contamination to link to release dates, but older MCQs (HellaSwag) have 0 contamination score while more recent ones (TQA) can be higher.

Lastly, choices-only accuracy can overestimate shortcut prevalence (Fig 2, top right). While ARC, TQA, and OBQA have high choices-only accuracy, it often stems from non-problematic strategies—the inferred question often matches the original. After filtering these cases, shortcut prevalence drops (e.g., $83\% \rightarrow 6\%$ on TQA). Past work cites high choices-only accuracy as evidence of dataset flaws (Chandak et al., 2025), but without considering *why* models succeed, this metric alone overestimates them.

4.2 MCQA Flaws Degrade LLM Evaluation

Having exposed benchmark flaws, we now test their evaluation impact, informing which issues to prioritize fixing. We evaluate 10 LLMs—GPT, Gemini, Claude, and Cohere §3.3—on our MCQs. Given the brittleness of LLMs' first-token probabilities (Wang et al., 2024a,b; Molfese et al., 2025), we instead use a prompt that requests a structured response with the model's selected choice, implemented in InspectAI (Appendix A.2). We report mean LLM accuracy[5] on the "Flaw" vs "No Flaw" splits (§4.1) to analyze how each flaw type relates to accuracy.

Scores differ per-dataset (Table 4), but: **1) Contaminated splits have higher accuracy**, similar to how memorized items are often easier for students and LLMs (Ebbinghaus, 1913; Sainz et al., 2023); **(2) Splits violating 2+ writing rules have lower accuracy**, aligning with research revealing poorly written items mislead test-takers (Schmucker and Moore, 2025; Nahum et al., 2025); and while Gupta et al. (2025) find filtering MCQs with choices-only success lowers MCQA accuracy,[6] they do not consider *how* success arises. But **(3) Splits with shortcuts have similar/mixed accuracy after correcting for strategy**, backing claims that choices-only success alone overstates shortcut issues (Balepur et al., 2025a). Viewing all flaws, writing errors tend to dominate (Table 4, right), so lower MCQA scores could stem from model failures in solving poorly-written items, rather than what MCQA aims to test.

Across every flaw, we argue that writing errors are most critical to fix, given their accuracy drops (Table 4) and prevalence (Fig 2). We study the most common writing flaws in §4.4 to guide future work.

4.3 MCQA Flaws Can Shift LLM Rankings

Users look at MCQA benchmark rankings to decide which LLMs to use daily (Liang et al., 2023), and researchers use them to select model checkpoints for further training (Walsh et al., 2025). To study if MCQ flaws could change these decisions, we run the 10 LLMs in §4.2 over all 12 benchmarks using the "Full" and "No Flaw" data splits. We also make a "Random" split by uniformly sampling the same

[5] Mean accuracy is used in NLP (Hofmann et al., 2025), but using difficulty from Item Response Theory (Lord and Novick, 2008), an education tool, maintains claims (Appendix A.8).

[6] We reproduce this result in Appendix A.10.

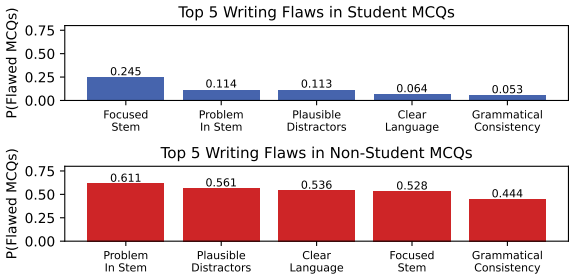

Figure 3: The five most common writing errors BenchMarker predicts in MCQA benchmarks, grouped by whether they stem from student exams. Most flaws relate to clarity and distractor difficulty. Appendix A.6 has the full distribution of 19 flaws.

number of MCQs as in the "No Flaw" split,[7] testing whether changes are just due to sampling variation.

LLM ranks on the "Full" and "No Flaw" splits are identical for contamination and shortcuts (Table 5), but ranks shift by up to two positions for writing errors. These shifts exceed random sampling: writing errors can change the models users and researchers select. We speculate these changes are starker when comparing LLMs of similar abilities (e.g., LLM pre-training checkpoints). This further backs our suggestion for NLP to devote more time toward reducing writing errors in benchmarks.

4.4 The Writing Errors for NLP to Address

Writing errors are pervasive (§4.1) and shift LLM ranks (§4.2), so we advise researchers to focus on fixing them. To inform these efforts, we now study the most common writing errors in our datasets. On student-based (standardized tests) and non-student-based MCQs, five flaws emerge: ambiguous question stems, indirectly asking questions, unclear language, grammatical inconsistency in questions and choices, and implausible distractors (Figure 3). The first four lead to unclear MCQs, an issue text simplification research has been working to remedy (Chandrasekar et al., 1996). The last links to distractor difficulty, matching recent education efforts in LLM distractor generation (McNichols et al., 2023; Lee et al., 2025). As similar writing errors plague MCQs in NLP and education—and both are building solutions—there is a clear opportunity for the fields to collaborate on tools to fix these flaws.

4.5 The Trade-Offs of Benchmark Fixes

Prior work also finds writing errors in MCQA benchmarks, spurring new versions. MMLU-Pro (Wang et al., 2024c) and MMLU-Redux (Gema et al., 2025) adjust MMLU to fix ambiguity and gold answers; GoldenSwag revises HellaSwag to improve grammar and distractors (Chizhov et al., 2025). We now run BenchMarker to study these revisions.

We first verify each revision works: GoldenSwag and MMLU-Redux fix labels, boosting accuracy, while MMLU-Pro drops accuracy via LLM-written distractors (Table 6). But GoldenSwag and MMLU-Pro add writing errors, which can impact evaluation (§4.2). Analyzing writing rules helps explain why. GoldenSwag removes MCQs with multiple correct answers ($18\% \rightarrow 4\%$) and differences in choice length ($52\% \rightarrow 43\%$) as intended, but adds grammar inconsistencies ($68\% \rightarrow 79\%$), perhaps due to its automated filtering. MMLU-Pro's LLM-written distractors lower accuracy, but we detect they are less plausible ($7\% \rightarrow 17\%$) and correct when they should be incorrect ($10\% \rightarrow 22\%$). For example, one MMLU-Pro MCQ asks for an element's change in atomic number after emitting particles, with answer "zero". LLMs incorrectly create the distractor "does not change"; this lowers accuracy, but only because models are split on which answer to select.

Finally, we test whether LLMs can rewrite MCQs to reduce the writing errors BenchMarker detects. We prompt GPT-5.2, Claude Sonnet, and Qwen-235B with MCQs from TruthfulQA and the writing errors detected by BenchMarker, and ask each model to return an MCQ that corrects these flaws. Across LLMs, this does not eliminate errors and can even introduce new ones (Table 7), motivating the

Dataset	Contam.	Shortcuts	Writing	Accuracy
HellaSwag	0.00	0.31	0.42	0.78
GoldenSwag	0.00	0.34	0.44	0.84
MMLU	0.45	0.12	0.12	0.79
MMLU Pro	0.24	0.12	0.15	0.63
MMLU Redux	0.44	0.13	0.12	0.81

Table 6: BenchMarker scores on MCQA datasets and their revised versions, along with the average accuracy of the 10 LLMs in §4.2. We score writing errors over all 19 rules. Revised datasets fix what they intend to (e.g. lower MMLU-Pro accuracy), but add errors (e.g. writing flaws in MMLU-Pro).

Model	ℙ(2+ Errors)	ℙ(Flawless)	ℙ(New Error)
GPT-5.2	$0.97 \rightarrow 0.64$	0.26	0.56
Claude Sonnet	$0.97 \rightarrow 0.64$	0.32	0.68
Qwen 235B	$0.97 \rightarrow 0.68$	0.26	0.62

Table 7: The proportion of MCQs with 2+ writing errors, no writing errors, and new errors after prompting GPT, Claude, and Qwen to fix writing errors in TruthfulQA. Simple prompting does not fully reduce errors and often introduces new ones.

[7] Averaged across 100 random seeds (0–99) for robustness.

need for future work to explore novel approaches beyond simple prompting for rewriting MCQs.

While these revisions take meaningful steps toward reliable MCQA evaluation, they highlight that benchmark correction is multi-objective: fixing one flaw can add others. Robust MCQA correction thus requires iterative refinement, and BenchMarker is one way to track progress. Researchers can use our tool to verify whether their targeted fixes improve MCQs as intended or inadvertently add new errors.

5 Related Work

As we instantiate MCQA educational theory via NLP methods, we review MCQA's history in both NLP evaluation (§5.1) and human assessments (§5.2)

5.1 Multiple-Choice Evaluation in NLP

Multiple-choice questions (MCQs) have historically been used in NLP, with early work testing commonsense (Levesque et al., 2012) and comprehension (Richardson et al., 2013); solving these was an "AI grand challenge" (Reddy, 1988). MCQA became standard with the advent of LLMs; Robinson and Wingate (2023) found one could prompt LLMs to answer MCQs like students and easily score them, spurring harder MCQs (Rein et al., 2024) and leaderboard use (Liang et al., 2023; Fourrier et al., 2024).

Despite this popularity, many NLP works show issues in MCQA. Models unreliably solve MCQs—brittle to symbolic (Alzahrani et al., 2024), logical (Kawabata and Sugawara, 2023; Balepur et al., 2024a, 2025b), and language (Singh et al., 2025) perturbations. MCQA datasets fall victim to contamination (Li et al., 2024b), poor grammar (Mousavi et al., 2025), plausibility errors (Palta et al., 2024), and shortcuts (Balepur et al., 2024b; Balepur and Rudinger, 2024). Prior research also argues that MCQA misaligns with educational assessment goals like commonsense (Davis, 2014), neglect real user needs (Saxon et al., 2024; Balepur et al., 2025c), and should be categorized by what the question intends to evaluate (Rodriguez and Boyd-Graber, 2021; Rogers et al., 2023). We synthesize these insights to prioritize flaws in BenchMarker's design.

In response, previous work has designed tools to improve MCQA benchmarks globally in diversity (Perlitz et al., 2024b), efficiency (Hofmann et al., 2025), and saturation (Polo et al., 2024), and at the MCQ level via dataset-specific annotation protocols (Wang et al., 2024c; Chizhov et al., 2025; Mousavi et al., 2025). Conversely, we present BenchMarker as a general toolkit for the latter and use it to audit 12 NLP benchmarks, study how flaws impact LLM evaluation, and analyze prior correction strategies.

5.2 Multiple-Choice Testing in Education

MCQA is a long-standing format for students (Monroe, 1917), but education researchers still look to boost its construct validity (Cronbach and Meehl, 1955): ensuring MCQA tests what it intends to. This has been achieved via new scoring (Finetti, 1965), answer formats (Snow, 2012), and adaptive testing (Lord, 1964)—all validated with students. We argue NLP can mirror this—testing how educational theory alters evaluation—as done in BenchMarker.

The closest work to ours is MCQ quality estimation (Wang et al., 2023). Such work often relies on similarity metrics like BLEU (Mulla and Gharpure, 2023), custom rules (Moore et al., 2023b), or item measures like perplexity (Raina and Gales, 2022), but these disagree with expert judgments (Van der Lee et al., 2021). Recent work has thus extended stronger NLP methods like LLM-as-a-judge (Moore et al., 2024), but usually target a single flaw type. In contrast, BenchMarker flags contamination, shortcuts, and writing errors, and we release all code and annotations to aid future work in this sparse area.

6 Conclusion: BenchMarker's Next Steps

Multiple-choice benchmarks are laced with flaws that harm NLP, but BenchMarker offers a path towards redemption: predicting MCQs with contamination, shortcuts, and writing errors. While BenchMarker's LLM judges are sufficient for benchmark audits, further work remains in executing contamination detection with cheaper search APIs, training efficient LLMs to match closed-source judges, and stress-testing BenchMarker's generalization across languages and domains. Our released code and annotated validation sets will facilitate these efforts.

Beyond diagnosis, our future work seeks to repair these flaws; we believe optimizing prompts for rewriting via BenchMarker-as-a-verifier (Opsahl-Ong et al., 2024), running user studies in human–AI rewriting interfaces (Cui et al., 2024; Wallace et al., 2019; Sung et al., 2025a), and drawing on educational testing theory for construct validation (Rodriguez et al., 2021; Hofmann et al., 2025) are useful next steps towards this. Overall, despite NLP facing an "evaluation crisis" (Blodgett et al., 2024), our paper shows educational standards are a valuable lifeline for rigorously assessing NLP systems.

7 Limitations

BenchMarker is currently designed to detect flaws in NLP benchmarks, but does not currently offer remediation strategies apart from flagging the item for human review or completely discarding the item. While still useful for improving NLP benchmarks, we believe a necessary future step is using these scores to refine MCQs. This step is outside the scope of our paper, but there is extensive research in automatically generating MCQs with LLMs (Lee et al., 2025; Parikh et al., 2025), and we are excited about applying these insights to future iterations of BenchMarker, incorporating LLM rewrites for these flaws without solely relying on human intervention.

While we have tested BenchMarker's generalization across our 12 MCQA benchmarks spanning different domains, difficulties, and creation strategies (Table 1), there are other types of MCQA datasets we did not explore, such as languages beyond English (Son et al., 2025; Li et al., 2024a), specific domains like medicine (Pal et al., 2022) and coding (Gu et al., 2024), and cultures (Chiu et al., 2025). Some dimensions may arise in multi-domain benchmarks like MMLU (Hendrycks et al., 2021) and Super GPQA (Du et al., 2025), but they were not a central focus of our experiments. By releasing our toolkit publicly, we hope to collect feedback from the NLP community on which areas BenchMarker struggles in and learn how it can be improved.

There are are many other issues in MCQA benchmarks that BenchMarker does not tackle, particularly at the global level like saturation (Saxon et al., 2024), efficiency (Perlitz et al., 2024a), and diversity (Singh et al., 2025). To narrow BenchMarker's scope, we draw on prior critiques of MCQA evaluations (Balepur et al., 2025c; Chandak et al., 2025) and education research (Haladyna and Downing, 1989) and focus on item-specific errors, leading to our flaws of contamination, shortcuts, and writing errors. As BenchMarker uses the InspectAI library (UK AI Security Institute, 2024), it is simple for researchers to extend our tool and add metrics they value, designed as "scorers" in the library.

Finally, some argue that benchmark errors (e.g. poor grammar) are representative of real-world user queries, so they do not need to be remedied. We counter that if reasoning over noisy inputs is part of the task description, these should still be introduced systematically and researchers should know which items have these flaws—ideally as a separate task (Guo and Vosoughi, 2024)—so researchers can better understand where their models fail. BenchMarker facilitates these evaluation efforts by detecting such flaws, allowing researchers to create clean and noisy splits of their benchmarks (§4.2).

8 Ethical Considerations

Low-quality benchmarks can undermine NLP evaluations, and BenchMarker takes steps to correct that in MCQA datasets, offering a toolkit to flag flaws in MCQs. However, our LLM judges are imperfect (§3), so we advise against using BenchMarker as the only tool to flag and fix quality errors of MCQA benchmarks, especially without any human intervention. As discussed in §6, we are excited about integrating BenchMarker into online user interfaces and running studies to understand how our toolkit can best support NLP researchers and educators.

Generative AI (GenAI) was used in this project. We used Cursor[8] to design plots and refactor code, and GPT-5 to refine paper writing for brevity. We never use GenAI for writing text from scratch in this paper. We take complete responsibility for any GenAI errors. By discussing GenAI usage here, we aim to encourage other researchers to do the same.

Acknowledgments

We would like to thank the CLIP lab at the University of Maryland for their support. In particular, we thank Ayush Jhaveri, Paiheng Xu, Alexander Hoyle for reviews and discussions on earlier versions of this paper draft. This material is based upon work supported by the National Science Foundation under IIS-2339746 (Rudinger) IIS-2403436 (Boyd-Graber), and DGE-2236417 (Balepur). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Access to Cohere models (Command-R, Command-R Plus) was made possible via a Cohere for AI Research Grant.

References

Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan AlRashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual*

[8] https://cursor.com/agents

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13787–13805, Bangkok, Thailand. Association for Computational Linguistics.

Anthropic. 2025. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5. Accessed: 2025-11-29.

Nishant Balepur, Atrey Desai, and Rachel Rudinger. 2025a. Test-time reasoners are strategic multiple-choice test-takers. *arXiv preprint arXiv:2510.07761*.

Nishant Balepur, Feng Gu, Abhilasha Ravichander, Shi Feng, Jordan Lee Boyd-Graber, and Rachel Rudinger. 2025b. Reverse question answering: Can an LLM write a question so hard (or bad) that it can't answer? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 44–64, Albuquerque, New Mexico. Association for Computational Linguistics.

Nishant Balepur, Shramay Palta, and Rachel Rudinger. 2024a. It's not easy being wrong: Large language models struggle with process of elimination reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10143–10166, Bangkok, Thailand. Association for Computational Linguistics.

Nishant Balepur, Abhilasha Ravichander, and Rachel Rudinger. 2024b. Artifacts or abduction: How do llms answer multiple-choice questions without the question? In *Annual Meeting of the Association for Computational Linguistics*.

Nishant Balepur and Rachel Rudinger. 2024. Is your large language model knowledgeable or a choices-only cheater? In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 15–26, Bangkok, Thailand. Association for Computational Linguistics.

Nishant Balepur, Rachel Rudinger, and Jordan Lee Boyd-Graber. 2025c. Which of these best describes multiple choice evaluation with LLMs? a) forced B) flawed C) fixable D) all of the above. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3394–3418, Vienna, Austria. Association for Computational Linguistics.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta. Association for Computational Linguistics.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2025. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255, Vienna, Austria. Association for Computational Linguistics.

Andrea J Bingham. 2023. From data management to actionable findings: A five-phase process of qualitative data analysis. *International journal of qualitative methods*, 22:16094069231183620.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Su Lin Blodgett, Jackie Chi Kit Cheung, Vera Liao, and Ziang Xiao. 2024. Human-centered evaluation of language technologies. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 39–43, Miami, Florida, USA. Association for Computational Linguistics.

Jonathan Bragg, Mike D'Arcy, Nishant Balepur, Dan Bareket, Bhavana Dalvi, Sergey Feldman, Dany Haddad, Jena D Hwang, Peter Jansen, Varsha Kishore, et al. 2025. Astabench: Rigorous benchmarking of ai agents with a scientific research suite. *arXiv preprint arXiv:2510.21652*.

Dianne E Campbell. 2011. How to write good multiple-choice questions. *Journal of paediatrics and child health*, 47(6):322–325.

Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. 2025. Eliminating discriminative shortcuts in multiple choice evaluations with answer matching. In *ICML 2025 Workshop on Assessing World Models*.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Yu Ying Chiu, Liwei Jiang, Bill Yuchen Lin, Chan Young Park, Shuyue Stella Li, Sahithya Ravi, Mehar Bhatia, Maria Antoniak, Yulia Tsvetkov, Vered Shwartz, and Yejin Choi. 2025. CulturalBench: A robust, diverse and challenging benchmark for measuring LMs' cultural knowledge through human-AI red-teaming. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25663–25701, Vienna, Austria. Association for Computational Linguistics.

Pavel Chizhov, Mattia Nee, Pierre-Carl Langlais, and Ivan P. Yamshchikov. 2025. What the hellaswag? on the validity of common-sense reasoning benchmarks.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

William G Cochran. 1977. Sampling techniques. *Johan Wiley & Sons Inc*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Eamon Costello, Jane Holland, and Colette Kirwan. 2018a. The future of online testing and assessment: question quality in moocs. *International Journal of Educational Technology in Higher Education*, 15(1):1–14.

Eamon Costello, Jane C Holland, and Colette Kirwan. 2018b. Evaluation of mcqs from moocs for common item writing flaws. *BMC research notes*, 11(1):849.

Lee J Cronbach and Paul E Meehl. 1955. Construct validity in psychological tests. *Psychological bulletin*, 52(4):281.

Yuan Cui, W Ge Lily, Yiren Ding, Lane Harrison, Fumeng Yang, and Matthew Kay. 2024. Promises and pitfalls: Using large language models to generate visualization items. *IEEE Transactions on Visualization and Computer Graphics*.

Ernest Davis. 2014. The limitations of standardized science tests as benchmarks for artificial intelligence research: Position paper. *arXiv preprint arXiv:1411.1629*.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8706–8719, Mexico City, Mexico. Association for Computational Linguistics.

Xeron Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shuyue Guo, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, Yizhi LI, Yunwen Li, dehua ma, Yuansheng Ni, Haoran Que, Qiyao Wang, Zhoufutu Wen, Siwei Wu, Tianshun Xing, Ming Xu, Zhenzhu Yang, Zekun Moore Wang, Junting Zhou, yuelin bai, Xingyuan Bu, chenglin cai, Liang Chen, Yifan Chen, Cheng Chengtuo, Tianhao Cheng, Keyi Ding, Siming Huang, HUANG YUN, Yaoru Li, Yizhe Li, Zhaoqun Li, Tianhao Liang, Chengdong Lin, Hongquan Lin, Yinghao Ma, Z.Y. Peng, Zifan Peng, Qige Qi, Shi Qiu, Xingwei Qu, Shanghaoran Quan, Yizhou Tan, Zili Wang, Chenqing Wang, Hao Wang, Yiya Wang, Yubo Wang, Jiajun Xu, Kexin Yang, Ruibin Yuan, Yuanhao Yue, Tianyang Zhan, Chun Zhang, Jinyang Zhang, Xiyue Zhang, Owen Xingjian Zhang, Yue Zhang, Yongchi Zhao, Xiangyu Zheng, ChenghuaZhong, Yang Gao, Zhoujun Li, Dayiheng Liu, Qian Liu, Tianyu Liu, Shiwen Ni, Junran Peng, Yujia Qin, Wenbo Su, Guoyin Wang, Shi Wang, Jian Yang, Min Yang, Meng Cao, Xiang Yue, Zhaoxiang Zhang, Wangchunshu Zhou, Jiaheng Liu, Qunshu Lin, Wenhao Huang, and Ge Zhang. 2025. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Hermann Ebbinghaus. 1913. *Memory: A contribution to experimental psychology.* Teachers College Press, New York.

Yanai Elazar, Akshita Bhagia, Ian Helgi Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Evan Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hannaneh Hajishirzi, Noah A. Smith, and Jesse Dodge. 2024. What's in my big data? In *The Twelfth International Conference on Learning Representations*.

Ahmed Elhady, Eneko Agirre, and Mikel Artetxe. 2025. WiCkeD: A simple method to make multiple choice benchmarks more challenging. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1183–1192, Vienna, Austria. Association for Computational Linguistics.

Bruno de Finetti. 1965. Methods for discriminating levels of partial knowledge concerning a test item. *British Journal of Mathematical and Statistical Psychology*, 18(1):87–123.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

Robert B Frary. 1988. Formula scoring of multiple-choice tests (correction for guessing). *Educational measurement: Issues and practice*, 7(2):33–38.

Yujuan Fu, Ozlem Uzuner, Meliha Yetisgen, and Fei Xia. 2025. Does data contamination detection work (well) for LLMs? a survey and evaluation on detection assumptions. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5235–5256, Albuquerque, New Mexico. Association for Computational Linguistics.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile Van Krieken, and Pasquale Minervini. 2025. Are we done with MMLU? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5069–5096, Albuquerque, New Mexico. Association for Computational Linguistics.

Aidan Gomez. 2024. Command r: Retrieval-augmented generation at production scale. https://cohere.com/blog/command-r. Accessed: 2025-11-29.

Alex Gu, Baptiste Roziere, Hugh James Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida Wang. 2024. CRUXEval: A benchmark for code reasoning, understanding and execution. In *Forty-first International Conference on Machine Learning*.

Xiaobo Guo and Soroush Vosoughi. 2024. Disordered-DABS: A benchmark for dynamic aspect-based summarization in disordered texts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 416–431, Miami, Florida, USA. Association for Computational Linguistics.

Vipul Gupta, Candace Ross, David Pantoja, Rebecca J. Passonneau, Megan Ung, and Adina Williams. 2025. Improving model evaluation using SMART filtering of benchmark datasets. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4595–4615, Albuquerque, New Mexico. Association for Computational Linguistics.

Thomas M Haladyna and Steven M Downing. 1989. A taxonomy of multiple-choice item-writing rules. *Applied measurement in education*, 2(1):37–50.

Thomas M Haladyna, Steven M Downing, and Michael C Rodriguez. 2002. A review of multiple-choice item-writing guidelines for classroom assessment. *Applied measurement in education*, 15(3):309–333.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 2024. LLM-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13806–13834, Bangkok, Thailand. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Valentin Hofmann, David Heineman, Ian Magnusson, Kyle Lo, Jesse Dodge, Maarten Sap, Pang Wei Koh, Chun Wang, Hannaneh Hajishirzi, and Noah A. Smith. 2025. Fluid language model benchmarking. In *Second Conference on Language Modeling*.

Akira Kawabata and Saku Sugawara. 2023. Evaluating the rationale understanding of critical reasoning in logical reading comprehension. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 116–143.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models.

Selmer C Larson. 1931. The shrinkage of the coefficient of multiple correlation. *Journal of Educational Psychology*, 22(1):45.

Paul Ngee Kiong Lau, Sie Hoe Lau, Kian Sam Hong, and Hasbee Usop. 2011. Guessing, partial knowledge, and misconceptions in multiple-choice tests. *Journal of Educational Technology & Society*, 14(4):99–110.

Yooseop Lee, Suin Kim, and Yohan Jo. 2025. Generating plausible distractors for multiple-choice questions via student choice prediction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23669–23692, Vienna, Austria. Association for Computational Linguistics.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024a. CMMLU: Measuring massive multitask language understanding in Chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285, Bangkok, Thailand. Association for Computational Linguistics.

Yucheng Li, Yunhao Guo, Frank Guerin, and Chenghua Lin. 2024b. An open-source data contamination report for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 528–541, Miami, Florida, USA. Association for Computational Linguistics.

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 Re, 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 S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification, Outstanding Certification.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. 2024. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. In *First Conference on Language Modeling*.

Frederic M Lord. 1964. The effect of random guessing on test validity. *Educational and Psychological Measurement*, 24(4):745–747.

Frederic M Lord and Melvin R Novick. 2008. *Statistical theories of mental test scores*. IAP.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Hunter McNichols, Wanyong Feng, Jaewook Lee, Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2023. Automated distractor and feedback generation for math multiple-choice questions via in-context learning.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Francesco Maria Molfese, Luca Moroni, Luca Gioffré, Alessandro Scirè, Simone Conia, and Roberto Navigli. 2025. Right answer, wrong score: Uncovering the inconsistencies of LLM evaluation in multiple-choice question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18477–18494, Vienna, Austria. Association for Computational Linguistics.

Walter S Monroe. 1917. A report on the use of the kansas silent reading tests with over one hundred thousand children. *Journal of Educational Psychology*, 8(10):600.

Steven Moore, Eamon Costello, Huy A Nguyen, and John Stamper. 2024. An automatic question usability evaluation toolkit. In *International Conference on Artificial Intelligence in Education*, pages 31–46. Springer.

Steven Moore, Ellen Fang, Huy A Nguyen, and John Stamper. 2023a. Crowdsourcing the evaluation of multiple-choice questions using item-writing flaws and bloom's taxonomy. In *Proceedings of the tenth ACM conference on learning@ scale*, pages 25–34.

Steven Moore, Huy A Nguyen, Tianying Chen, and John Stamper. 2023b. Assessing the quality of multiple-choice questions using gpt-4 and rule-based methods. In *European Conference on Technology Enhanced Learning*, pages 229–245. Springer.

Seyed Mahed Mousavi, Edoardo Cecchinato, Lucia Hornikova, and Giuseppe Riccardi. 2025. Garbage in, reasoning out? why benchmark scores are unreliable and what to do about it. *arXiv preprint arXiv:2506.23864*.

Nikahat Mulla and Prachi Gharpure. 2023. Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications. *Progress in Artificial Intelligence*, 12(1):1–32.

Omer Nahum, Nitay Calderon, Orgad Keller, Idan Szpektor, and Roi Reichart. 2025. Are LLMs better than reported? detecting label errors and mitigating their effect on model performance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26770–26797, Suzhou, China. Association for Computational Linguistics.

OpenAI. 2025. Gpt-5 system card. https://openai.com/index/gpt-5-system-card/. Accessed: 2025-11-28.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. Optimizing instructions and demonstrations for multi-stage language model programs.

In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9340–9366, Miami, Florida, USA. Association for Computational Linguistics.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR.

Daniel Paleka, Shashwat Goel, Jonas Geiping, and Florian Tramèr. 2025. Evaluating forecasting is more difficult than other LLM evaluations. In *ICML 2025 Workshop on Assessing World Models*.

Shramay Palta, Nishant Balepur, Peter Rankel, Sarah Wiegreffe, Marine Carpuat, and Rachel Rudinger. 2024. Plausibly problematic questions in multiple-choice benchmarks for commonsense reasoning. In *Conference on Empirical Methods in Natural Language Processing*.

Nisarg Parikh, Alexander Scarlatos, Nigel Fernandez, Simon Woodhead, and Andrew Lan. 2025. LookAlike: Consistent distractor generation in math MCQs. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 294–311, Vienna, Austria. Association for Computational Linguistics.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. 2024a. Efficient benchmarking (of language models). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2519–2536, Mexico City, Mexico. Association for Computational Linguistics.

Yotam Perlitz, Ariel Gera, Ofir Arviv, Asaf Yehudai, Elron Bandel, Eyal Shnarch, Michal Shmueli-Scheuer, and Leshem Choshen. 2024b. Do these llm benchmarks agree? fixing benchmark evaluation with benchbench. *arXiv preprint arXiv:2407.13696*.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. tinybenchmarks: evaluating llms with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Vatsal Raina and Mark Gales. 2022. Multiple-choice question generation: Towards an automated assessment framework. *arXiv preprint arXiv:2209.11830*.

Raj Reddy. 1988. Foundations and grand challenges of artificial intelligence: Aaai presidential address. *AI magazine*, 9(4):9–9.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Kyle Richardson and Ashish Sabharwal. 2020. What does my qa model know? devising controlled probes using expert knowledge. *Transactions of the Association for Computational Linguistics*, 8:572–588.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 193–203.

Joshua Robinson and David Wingate. 2023. Leveraging large language models for multiple choice question answering. In *The Eleventh International Conference on Learning Representations*.

Pedro Rodriguez, Joe Barrow, Alexander Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503, Online. Association for Computational Linguistics.

Pedro Rodriguez and Jordan Boyd-Graber. 2021. Evaluation paradigms in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9630–9642, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Henry L Roediger and Andrew C Butler. 2011. The critical role of retrieval practice in long-term retention. *Trends in cognitive sciences*, 15(1):20–27.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2023. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Computing Surveys*, 55(10):1–45.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Michael Saxon, Ari Holtzman, Peter West, William Yang Wang, and Naomi Saphra. 2024. Benchmarks as microscopes: A call for model metrology. In *First Conference on Language Modeling*.

Robin Schmucker and Steven Moore. 2025. The impact of item-writing flaws on difficulty and discrimination in item response theory. *arXiv preprint arXiv:2503.10533*.

Mark G Simkin and William L Kuechler. 2005. Multiple-choice tests and student understanding: What is the connection? *Decision Sciences Journal of Innovative Education*, 3(1):73–98.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, Enzo Ferrante, Marzieh Fadaee, Beyza Ermis, and Sara Hooker. 2025. Global MMLU: Understanding and addressing cultural and linguistic biases in multilingual evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria. Association for Computational Linguistics.

Gregory T Smith. 2005. On construct validity: issues of method and measurement. *Psychological assessment*, 17(4):396.

Richard E Snow. 2012. Construct validity and constructed-response tests. In *Construction versus choice in cognitive measurement*, pages 45–60. Routledge.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. 2025. KMMLU: Measuring massive multitask language understanding in Korean. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4076–4104, Albuquerque, New Mexico. Association for Computational Linguistics.

Yoo Yeon Sung, Maharshi Gor, Eve Fleisig, Ishani Mondal, and Jordan Lee Boyd-Graber. 2025a. Is your benchmark truly adversarial? AdvScore: Evaluating human-grounded adversarialness. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 623–642, Albuquerque, New Mexico. Association for Computational Linguistics.

Yoo Yeon Sung, Maharshi Gor, Eve Fleisig, Ishani Mondal, and Jordan Lee Boyd-Graber. 2025b. Is your benchmark truly adversarial? AdvScore: Evaluating human-grounded adversarialness. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 623–642, Albuquerque, New Mexico. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Marie Tarrant, Aimee Knierim, Sasha K Hayes, and James Ware. 2006. The frequency of item writing flaws in multiple-choice questions used in high stakes nursing assessments. *Nurse Education Today*, 26(8):662–671.

Cynthia Taylor, Michael Clancy, Kevin C. Webb, Daniel Zingaro, Cynthia Lee, and Leo Porter. 2020. The practical details of building a cs concept inventory. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, page 372–378, New York, NY, USA. Association for Computing Machinery.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

UK AI Security Institute. 2024. Inspect AI: Framework for Large Language Model Evaluations.

Arda Uzunoglu, Tianjian Li, and Daniel Khashabi. 2025. The flaw of averages: Quantifying uniformity of performance on benchmarks. *arXiv preprint arXiv:2509.25671*.

Chris Van der Lee, Albert Gatt, Emiel Van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.

Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R. Bowman. 2021. Comparing test sets with item response theory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1141–1158, Online. Association for Computational Linguistics.

Ellen M Voorhees. 2001. The trec question answering track. *Natural Language Engineering*, 7(4):361–378.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James Validad Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. 2 OLMo 2 furious (COLM's version). In *Second Conference on Language Modeling*.

Xinpeng Wang, Chengzhi Hu, Bolei Ma, Paul Rottger, and Barbara Plank. 2024a. Look at the text: Instruction-tuned language models are more robust multiple choice selectors than you think. In *First Conference on Language Modeling*.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024b. " my answer is c": First-token probabilities do not match text answers in instruction-tuned language models. *arXiv preprint arXiv:2402.14499*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024c. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zifan Wang, Kotaro Funakoshi, and Manabu Okumura. 2023. Automatic answerability evaluation for question generation. *arXiv preprint arXiv:2309.12546*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Hao Xu, Jiacheng Liu, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. Infini-gram mini: Exact n-gram search at the Internet scale with FM-index. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24955–24980, Suzhou, China. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2299–2314.

Yan Zhuang, Qi Liu, Zachary Pardos, Patrick C. Kyllonen, Jiyun Zu, Zhenya Huang, Shijin Wang, and Enhong Chen. 2025. Position: AI evaluation should learn from how we test humans. In *Forty-second International Conference on Machine Learning Position Paper Track*.

Vilém Zouhar, Peng Cui, and Mrinmaya Sachan. 2025. How to select datapoints for efficient human evaluation of nlg models? *Transactions of the Association for Computational Linguistics*, 13:1789–1811.

A Appendix

A.1 Survey of NLP Evaluation

To motivate our design of BenchMarker, we first survey AI papers that release MCQA benchmarks. We search Google Scholar, Semantic Scholar, the ACL Anthology, and query Deep Research tools (ScholarQA) with keywords related to "multiple-choice" and "benchmarks", yielding 39 total papers from 2013–2025 for manual review. For each paper, we mark: 1) whether the authors report any dataset quality control; and 2) whether the authors contextualize this quality with respect to other datasets.

We find that 23% of benchmark papers report no quality control, and 49% do not compare dataset quality to other datasets. While comparing dataset quality is nice to have, we believe reporting quality control is necessary—fortunately, BenchMarker aids both of these goals. Several works that draw directly from human exams report no quality control, assuming they are high-quality, but our analysis reveals that even these questions can have flaws (§4.1). Thus, we recommend that all researchers review their MCQA benchmarks before release.

A.2 InspectAI Implementation

InspectAI[9] is a recent effort from the United Kingdom's AI Security Institute to standardize NLP evaluations (UK AI Security Institute, 2024). Any InspectAI framework contains three parts:

1. **Task:** The data for the task. In our setup, these are the question stem, choices, and answer of the multiple-choice question.
2. **Solver:** The NLP system that solves the task. In our setup, this is either a function that returns the entire dataset when we are scoring the dataset itself, or these are the LLMs from §4.2 that we run LLMs on our dataset.
3. **Scorer:** How task success/failure is evaluated. In our setup, these are either the contamination, shortcuts, and writing flaw judges used in BenchMarker, or a standard accuracy score when we run LLMs on our dataset.

InspectAI allows researchers to easily add their own tasks, solvers, and scorers in a standardized way, which in our case, will allow researchers to extend BenchMarker more easily. It also provides an easy-to-use UI, which could form the basis for future user studies in LLM evaluation (Figure 4). The library has been adopted in other benchmarking efforts like AstaBench (Bragg et al., 2025), and we use it in BenchMarker to motivate further use. Appendix A.13 has the prompts used in Inspect.

[9] https://inspect.aisi.org.uk/

A.3 Dataset Details

Out of the datasets in the paper, we use the training sets of each dataset for judge validation, and the test sets of each dataset for auditing. The commonsense datasets (e.g., SocialIQA) have fully held-out test sets, so we instead use the validation sets; we find it to be a positive sign that the validation sets have low contamination rates, as this is likely an upper bound for test set contamination. For datasets with more than 1000 MCQs, we sample 1000 uniformly for auditing. Exceptions are AQuA, SAT, and TruthfulQA, which only have a single set of questions, so we split them evenly between train and test, yielding 127 test set items for AQuA, 104 test set items for SAT, and 409 test set items for TruthfulQA. We take random samples, so our metrics are unbiased estimates of the full data.

All datasets are publicly available, so our experiments are within their intended use. We did not collect any datasets, so we did not check for PII. To our knowledge, all questions are in English.

A.4 LLM Implementation Details

We run every LLMs using default parameters, both for judge experiments and benchmark audits. We use CPUs only when running LLMs via APIs, one NVIDIA rtxa6000 GPU for open-weight LLMs below 8B parameters called via Huggingface, and eight NVIDIA rtxa5000's for all other open-weight LLMs called via Huggingface. We allocate 24 hours for each run. All results are reported from a single run.

In our experiments, we run all closed-source and Cohere models using litellm[10] and all other models via Huggingface's inference endpoint.[11] The APIs and Huggingface endpoints for our LLMs are:

- openai/gpt-5-nano-2025-08-07
- openai/gpt-5-mini-2025-08-07
- openai/gpt-5-2025-08-07
- anthropic/claude-haiku-4-5-20251001
- anthropic/claude-sonnet-4-5-20250929
- gemini/gemini-2.5-pro
- gemini/gemini-2.5-flash-lite
- gemini/gemini-2.5-flash
- cohere/command-r-08-2024
- cohere/command-r-plus-08-2024

[10] https://www.litellm.ai/

[11] https://huggingface.co/docs/inference-endpoints/en/index

- Qwen/Qwen3-0.6B
- Qwen/Qwen3-1.7B
- Qwen/Qwen3-4B
- Qwen/Qwen3-8B
- Qwen/Qwen3-14B
- Qwen/Qwen3-32B
- google/gemma-3-4b-it
- google/gemma-3-12b-it
- google/gemma-3-27b-it

The LLM used to evaluate each of the 19 writing flaws are as follows:

- avoid_k_type: google/gemini-2.5-flash-lite,
- avoid_negatives: anthropic/claude-sonnet-4-5-20250929
- avoid_repetition: google/gemini-2.5-pro
- clear_language: openai/gpt-5-2025-08-07
- equal_length_options: google/gemini-2.5-pro
- focused_stem: google/gemini-2.5-pro
- grammatical_consistency: google/gemini-2.5-pro
- no_absolute_terms: google/gemini-2.5-pro
- no_all_of_the_above: anthropic/claude-sonnet-4-5-20250929
- no_convergence_cues: anthropic/claude-sonnet-4-5-20250929
- no_extraneous_info: openai/gpt-5-2025-08-07
- no_fill_in_blank: google/gemini-2.5-pro
- no_logical_cues: openai/gpt-5-2025-08-07
- no_none_of_the_above: google/gemini-2.5-flash
- no_vague_terms: google/gemini-2.5-flash
- ordered_options: google/gemini-2.5-pro
- plausible_distractors: openai/gpt-5-2025-08-07
- problem_in_stem: google/gemini-2.5-flash-lite
- single_best_answer': google/gemini-2.5-flash

A.5 Annotation Details

Our human annotation protocol draws on qualitative coding in HCI (Bingham, 2023): three authors label the three different metrics (one author per metric), then a second author rates 50 random items to compute agreement. The annotators are graduate students in computer science with native fluency in English, all with previous experience in evaluation research and MCQ annotations. Contamination yields 84% agreement, shortcuts yields 84% agreement, and the lowest agreement on any writing flaw is 85%. The annotation protocols for contamination, shortcuts, and writing flaws are in §3.2; the rule definitions shown for writing flaws are in Table 8. This annotation was deemed except by our institution's Internal Review Board (IRB).

A.6 Full Writing Flaw Results

Due to space constraints, we report writing flaw scores aggregated over all flaws or with a standard cutoff of violating two or more rules, but we now analyze each individual rule. In Figure 5, we show the score of each writing flaw across all datasets, which provides interesting, dataset-specific quirks; AQuA and SuperGPQA fail to sort their choices, HellaSwag is the main culprit of using vague terms, and TruthfulQA is the only benchmark with pervasive convergence cues. Figure 6 extends Figure 3 to sort the prevalence of the 19 writing flaws across student-based and non-student-based benchmarks, which could inform education or NLP researchers as to which issues are important to immediately fix.

A.7 Full Search Engine Results

Table 11 reports our contamination results across all judge and search engine combinations. Along with the GPT-5, Gemini-2.5 Pro, and Claude-4.5 Sonnet judges, we also assess: 1) an Oracle judge, which makes a perfect/oracle decision if any search results appear; and 2) a Simple judge, which classifies the MCQ as flawed if any search results appear, and not flawed if there are no search results. Ignoring the Oracle classifier, Google is consistently the strongest search engine. The simple classifier slightly surpasses GPT-5 at using Google's search results, but GPT-5 can also generate an explanation that synthesizes all of the web pages; we personally found this useful for debugging and expect other researchers to feel similarly, so we use GPT-5 for contamination detection in BenchMarker.

A.8 Results with Item Response Theory

Our analysis in §4.2 reports the average accuracy of 10 LLMs to study the impact of flaws in LLM evaluation—a proxy for the "difficulty" of an MCQ. However, this measure could become more informative after considering model abilities. To illustrate, an MCQ answered just by GPT-5 Nano and an MCQ answered just by GPT-5 would have the same accuracy, but the latter is more difficult, as we know GPT-5 tends to have higher accuracy on average (Sung et al., 2025b). Item Response Theory (Lord, 1964, IRT) is a tool from educational

testing that controls for this; it estimates the difficulty (how hard the MCQ is) and discriminability (how well the MCQ discerns model skills) by learning the abilities of the models run on the MCQA benchmarks.

While IRT is a standard metric to report when validating education interventions (Schmucker and Moore, 2025), and with recent growing interest in NLP (Hofmann et al., 2025), we felt average accuracy would be more familiar and easier to interpret for an NLP audience. To reap the benefits of IRT, we replicate the analysis in §4.2 but with average difficulty (Table 9) and discriminability (Table 10) as the metrics, which do not alter our claims; contaminated items have lower difficulty and discriminability, while items with shortcuts and writing errors have higher difficulty and discriminability.

A.9 Contamination Detection in Pre-training

Prior work offers many ways to predict whether test items exist in LLM training sets (Fu et al., 2025). Some methods query black-box LLMs (Sainz et al., 2023), but we desire a model-agnostic technique, isolating MCQ contamination irrespective of model behavior. Other methods search large corpora, but these either rely on exact string matching (Xu et al., 2025)—missing perturbed MCQs—or index entire corpora (Elazar et al., 2024)—which is resource-intensive. Instead, we use the Internet as a proxy for training data, assuming if an MCQ exists online, it likely exists in at least one LLM's training data (Balloccu et al., 2024). This is model-agnostic, cheap, and uses relevance ranking for near matches.

While search engine APIs do not require intense resources and can surface semantic matches (§2.1), there are cheaper alternatives like Infini-Gram (Liu et al., 2024): an n-gram (trillion) language model with publicly-available indexes over common pre-training corpora. To see whether Infini-Gram can more efficiently replace our judges, we run the tool indexed over varied corpora on our contamination validation set—predicting the item as contaminated when the question stem exists exactly at least once. Overall, accuracy and Cohen's κ are significantly lower (Table 12), demonstrating that our method is a stronger way to detect whether MCQs exist online.

A.10 Shortcut Detection Across Models

Our shortcut detection strategy uses majority vote of three strong LLMs that answer MCQs without the question (§2.2), so we now test differences when we use one LLM. In Figure 7 and Table 13, we replicate Figure 2 and Table 4, evaluating the prevalence of shortcuts and their impact on LLM evaluation. Trends are consistent with majority vote—relative shortcut prevalence is typically preserved across datasets and items with shortcuts have weakly lower accuracy—but there is model-specific noise, motivating the benefits of an ensembling approach.

In Table 14, we also study LLM accuracy changes when evaluating on MCQs with and without choices-only success. We see items where models succeed with choices-only have much higher accuracy—reproducing Gupta et al. (2025)—which confirms that accounting for *how* LLMs achieve choices-only success has substantially different implications on evaluation. We also note that our choices-only accuracy aligns closely with Balepur et al. (2025a).

A.11 BenchMarker Cost Analysis

We report the input tokens consumed and output tokens produced in our BenchMarker runs across datasets in Table 15. For each run, we also show the estimated cost per item using the pricing for Gemini-2.5 Flash and Pro.[12] While Gemini Pro is a slightly more reliable model (Table 2), Gemini-2.5 Flash has similarly Cohen's κ and significantly cheaper, suggesting the latter is a strong choices for researchers with limited computation budgets.

A.12 Open-Weight Configurations

To improve open-weight LLMs for writing flaw detection (§3.5), we run experiments on ensembling, confidence, and writing flaw failure taxonomies.

Ensembling. To test benefits of ensembling, we test all combinations of three open-weight judges; Command-R, Command-R+, and Qwen3-8B reach the highest Cohen's κ of 0.391. However, this is not much higher than Command-R alone (κ of 0.376), so ensembling is likely not worth this extra computation for writing flaw detection.

Confidence. To study the benefits of confidence calibration, we analyze the 1–10 confidence scores produced by LLM judges in Prompt A.5. In Table 16, we show Cohen's κ and the proportion of examples predicted (in parentheses) beyond different confidence thresholds. Overall, some open-weight models are well-calibrated, with Command R reaching Cohen's Kappa as high as 0.67 when giving a confidence score of 10.

[12] https://ai.google.dev/gemini-api/docs/pricing

Failure Taxonomy. The above analyses and results in Table 2 support that Cohere Command-R is a strong open-weight model for writing error detection, so we provide a breakdown of its Cohen's κ across the 19 flaw types in Table 17 to see where the model fails and succeeds. Command-R excels in simpler judgments like detecting certain options and option order, but disagrees with experts more on complex, subjective criteria like extraneousness, repetition, and convergence.

Recommendation. Cohere Command-R has the highest agreement with humans in Table 17, is part of the strongest judge ensemble, and appears well-calibrated, so we recommend that researchers with GPUs only use Command-R. If some API credits are available, we recommend researchers to employ stronger closed-source LLMs when Command-R predicts confidence lower than 9, and for writing error types the model tends to struggle with, like convergent clues, extraneous info, and repetition.

A.13 Prompts

This section outlines our prompts. The prompt we use for running LLMs on MCQA (Prompt A.1). is taken directly from Inspect (UK AI Security Institute, 2024), except we do not ask the model's answer to be preceded by the dollar sign character ($); preliminary analysis found many LLMs struggled with this, lowering scores more than normal. Prompts A.2, A.3, A.4, and A.5 are the LLM instructions for detecting contaminated MCQs on web pages, using LLMs to answer MCQs with just the choices, detecting whether the inferred question matches the original one, and following the 19-rule education rubric for writing flaws, respectively. For contamination, we consider the item to be flawed if the LLM judge outputs "partial_match" or "no_match", and for shortcuts, we consider the item flawed if the LLM judge outputs "no_match".

Name	Rule	Example Violation
Grammatical Consistency	All options must use parallel grammatical structure and fit with the stem.	The moon orbits an object that orbits the (D) mars
Focused Stem	The stem must present one clear, focused question or problem.	What are social?
Problem in Stem	The stem must fully state the problem rather than relying on the options to introduce it.	Winter in the Northern Hemisphere means
Single Best Answer	The question must have exactly one best answer with no equally correct alternatives.	Where could you get something that is made out of wool but cannot be worn? (C) fabric store (E) clothing factory
No Extraneous Information	The stem and choices must exclude text not needed to answer the question.	After school, Alex took Sasha's daughter to the playground to play. The two were good friends in the same class. What will Sasha want to do next?
Avoid K-Type Options	Choices must not combine items (e.g., "A and B only").	(A) Wrong, Wrong (B) Wrong, Not wrong (C) Not wrong, Wrong (D) Not wrong, Not wrong
Clear Language	All wording must be clear and unambiguous.	What lasts only as long as the antibodies survive in body fluids?
Plausible Distractors	All distractors must be plausible and relevant to the topic in the question stem.	[math question asking for probability] (E) 1.5
Avoid Repetition	The correct answer must not repeat words or phrases from the question stem.	In the morning you return to work, in the evening you? Answer: leave work
No Logical Cues	Choices must not provide logical cues that reveal the correct answer.	Who has blood and parents? (A) person (B) bloodbank (C) vein (D) capillaries (E) hospital
No Convergence Cues	The correct answer must not combine elements repeatedly appearing across distractors.	(A) an abundance of fire (B) absolutely zero snow outside (C) a plethora of snow (answer) (D) frogs falling from sky
Equal-Length Options	All choices must have similar length and detail.	(A) a violent disturbance of the atmosphere with high winds (B) ideas in print media (C) a large jet engine (D) a gas guzzling automobile
Ordered Options	Numerical choices must be in ascending order.	(A) 18cm (B) 22cm (C) 20cm (D) 30cm (E) 28cm
No Absolute Terms	Choices must not use absolute terms unless the statement is truly absolute.	Will happen to the number of islands if the planet's temperature rises? (A) they will increase (B) nothing will happen (C) they will shrink (D) they will double
No Vague Terms	Choices must not use vague, unquantified terms like "often" or "usually."	When a water balloon is frozen, it will contain (A) a much less amount of water (B) a whole bunch of frozen ice-cream
Avoid Negative Stems	The stem must not be framed negatively using terms like "NOT" or "EXCEPT."	All of the following are ways in which lobbyists attempt to persuade legislators EXCEPT...
No "All of the Above"	Choices should not include "All of the above."	(D) all of these
No "None of the Above"	Choices should not include "None of the above"	(E) None of these
No Fill-in-the-Blank	The stem must not use blanks or require choices to complete an incomplete sentence.	Experiments are performed in the ___.

Table 8: The 19 writing flaw rubric from (Tarrant et al., 2006) and example violations from MCQA benchmarks.

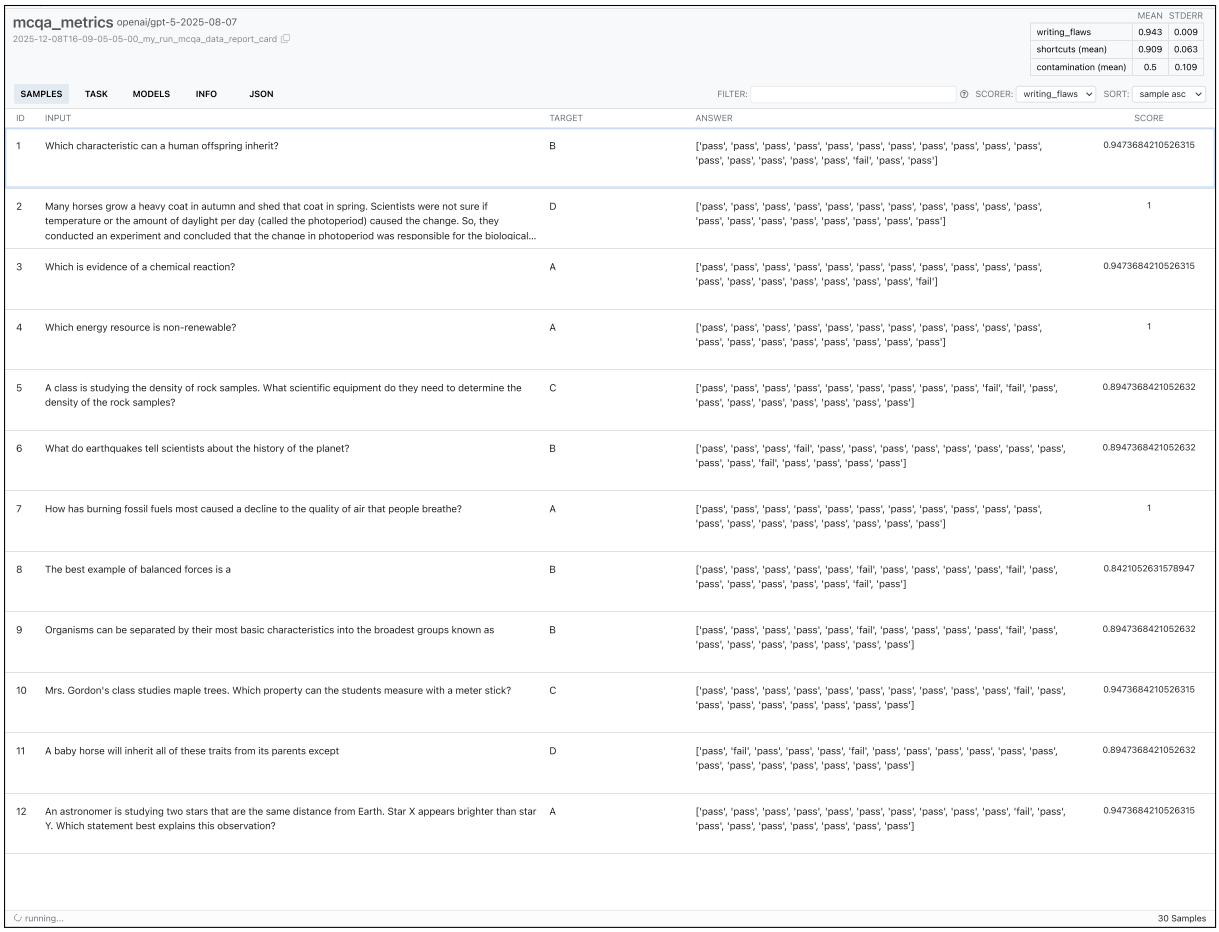

Figure 4: Interface from InspectAI for viewing BenchMarker runs. The overall scores are in the top right, and researchers can click on specific MCQs to view LLM judge calls and feedback, supporting debugging and analysis.

	Contamination				*Shortcuts*				*2+ Writing Flaws*				*Any Flaw*			
Dataset	Flaw	No Flaw	ΔAcc	ℙ(Flaw)	Flaw	No Flaw	ΔAcc	ℙ(Flaw)	Flaw	No Flaw	ΔAcc	ℙ(Flaw)	Flaw	No Flaw	ΔAcc	ℙ(Flaw)
AQUA	0.007	-0.046	-742.1	17%	-0.032	-0.038	+19.3	20%	0.236	-0.252	-206.8	44%	0.069	-0.218	-414.0	63%
ARC	-0.616	-0.493	-19.9	38%	-0.410	-0.548	+33.8	6%	-0.468	-0.595	+27.1	43%	-0.541	-0.539	-0.4	70%
CQA	-0.096	-0.217	+126.0	6%	0.092	-0.221	-341.8	4%	-0.193	-0.638	+230.2	96%	-0.196	-0.624	+217.9	97%
HS	-0.501	-0.181	-63.8	0%	-0.008	-0.212	+2666.1	14%	-0.182	—	—	100%	-0.182	—	—	100%
MMLU	-0.420	-0.055	-86.9	45%	-0.102	-0.232	+127.7	9%	-0.135	-0.372	+175.1	64%	-0.223	-0.204	-8.7	84%
OBQA	-0.501	-0.553	+10.3	4%	-0.513	-0.561	+9.3	21%	-0.530	-0.734	+38.3	90%	-0.534	-0.749	+40.4	92%
PIQA	-0.690	-0.725	+5.0	5%	-0.543	-0.770	+41.9	20%	-0.714	-0.834	+16.8	92%	-0.715	-0.848	+18.6	94%
QASC	0.491	0.472	-3.9	6%	0.639	0.466	-27.1	4%	0.485	0.082	-83.2	97%	0.479	0.257	-46.2	98%
SAT	0.073	-0.120	-264.9	24%	-0.072	-0.075	+4.6	21%	0.033	-0.111	-440.3	25%	-0.040	-0.115	+188.1	54%
SIQA	-1.096	-0.331	-69.8	0%	-0.326	-0.333	+2.2	23%	-0.298	-0.993	+233.0	95%	-0.304	-1.004	+230.7	96%
SGPQA	1.006	0.992	-1.4	6%	0.831	1.011	+21.6	10%	1.082	0.687	-36.5	77%	1.064	0.691	-35.1	81%
TQA	-0.123	-0.181	+47.5	47%	0.543	-0.195	-135.9	6%	-0.133	-0.823	+516.8	97%	-0.145	-0.863	+495.9	99%
Micro μ	-0.312	-0.114	-63.4	14%	-0.172	-0.138	-20.0	12%	-0.101	-0.348	+245.6	83%	-0.129	-0.250	+93.1	90%
Macro μ	-0.205	-0.120	-41.7	17%	0.008	-0.142	-1816.3	13%	-0.068	-0.417	+510.6	77%	-0.106	-0.383	+262.9	85%

Table 9: Impact of MCQA benchmark flaws on LLM difficulty, computed via Item Response Theory. The trend is consistent with Table 4; contaminated items have lower difficulty, while items with shortcuts and writing flaws have higher difficulty.

	Contamination				*Shortcuts*				*2+ Writing Flaws*				*Any Flaw*			
Dataset	Flaw	No Flaw	ΔAcc	ℙ(Flaw)	Flaw	No Flaw	ΔAcc	ℙ(Flaw)	Flaw	No Flaw	ΔAcc	ℙ(Flaw)	Flaw	No Flaw	ΔAcc	ℙ(Flaw)
AQUA	1.282	1.249	-2.5	17%	1.236	1.259	+1.9	20%	1.217	1.285	+5.5	44%	1.232	1.294	+5.1	63%
ARC	1.384	1.361	-1.6	38%	1.295	1.374	+6.1	6%	1.376	1.366	-0.7	43%	1.377	1.352	-1.8	70%
CQA	1.295	1.254	-3.1	6%	1.169	1.260	+7.8	4%	1.255	1.316	+4.9	96%	1.255	1.315	+4.7	97%
HS	1.075	1.235	+14.9	0%	1.167	1.246	+6.8	14%	1.235	—	—	100%	1.235	—	—	100%
MMLU	1.343	1.252	-6.7	45%	1.256	1.297	+3.3	9%	1.282	1.313	+2.4	64%	1.300	1.258	-3.2	84%
OBQA	1.297	1.325	+2.2	4%	1.325	1.324	-0.1	21%	1.317	1.386	+5.3	90%	1.317	1.399	+6.2	92%
PIQA	1.357	1.321	-2.6	5%	1.284	1.333	+3.8	20%	1.321	1.353	+2.4	92%	1.321	1.353	+2.4	94%
QASC	1.208	1.256	+4.0	6%	1.174	1.257	+7.0	4%	1.252	1.296	+3.5	97%	1.253	1.264	+0.9	98%
SAT	1.374	1.352	-1.5	24%	1.357	1.358	+0.1	21%	1.330	1.367	+2.8	25%	1.347	1.369	+1.7	54%
SIQA	1.485	1.252	-15.7	0%	1.241	1.256	+1.2	23%	1.244	1.406	+13.0	95%	1.246	1.411	+13.3	96%
SGPQA	1.160	1.221	+5.2	6%	1.217	1.217	+0.0	10%	1.214	1.226	+0.9	77%	1.216	1.220	+0.3	81%
TQA	1.256	1.280	+1.9	47%	1.211	1.272	+5.0	6%	1.268	1.302	+2.7	97%	1.269	1.267	-0.2	99%
Micro μ	1.324	1.273	-3.8	14%	1.253	1.284	+2.5	12%	1.271	1.330	+4.6	83%	1.277	1.312	+2.7	90%
Macro μ	1.293	1.280	-1.0	17%	1.244	1.288	+3.5	13%	1.276	1.329	+4.1	77%	1.281	1.318	+2.9	85%

Table 10: Impact of MCQA benchmark flaws on LLM discriminability, computed via Item Response Theory. The trend is consistent with Table 4; contaminated items have lower discriminability, while items with shortcuts and writing flaws have higher discriminability.

Judge Classification Type	Search Engine	Accuracy	F1 Score	Cohen's κ
Oracle	Google	0.8035	0.7761	0.6161
Oracle	Brave	0.6026	0.4129	0.2456
Oracle	Perplexity	1.0000	1.0000	1.0000
Oracle	Exa	0.9825	0.9835	0.9650
Oracle	Tavily	0.9694	0.9707	0.9388
Oracle	Serper	0.7598	0.7120	0.5337
Simple	Google	0.7293	0.7156	0.4653
Simple	Brave	0.5852	0.4025	0.2105
Simple	Perplexity	0.5371	0.6989	0.0000
Simple	Exa	0.5546	0.7000	0.0458
Simple	Tavily	0.5197	0.6784	-0.0305
Simple	Serper	0.6463	0.6267	0.3019
GPT-5	Google	0.7118	0.6765	0.4358
GPT-5	Brave	0.5415	0.2759	0.1349
GPT-5	Perplexity	0.6419	0.5287	0.3122
GPT-5	Exa	0.5895	0.4268	0.2164
GPT-5	Tavily	0.5808	0.4074	0.2007
GPT-5	Serper	0.6725	0.6445	0.3560
Claude Sonnet	Google	0.6900	0.6359	0.3965
Claude Sonnet	Brave	0.5415	0.2657	0.1359
Claude Sonnet	Perplexity	0.6332	0.5385	0.2919
Claude Sonnet	Exa	0.5764	0.4049	0.1919
Claude Sonnet	Tavily	0.5808	0.4000	0.2017
Claude Sonnet	Serper	0.6507	0.6226	0.3127
Gemini Pro	Google	0.6987	0.6497	0.4128
Gemini Pro	Brave	0.5415	0.2657	0.1359
Gemini Pro	Perplexity	0.6288	0.5304	0.2839
Gemini Pro	Exa	0.5764	0.4049	0.1919
Gemini Pro	Tavily	0.5983	0.4321	0.2340
Gemini Pro	Serper	0.6638	0.6351	0.3389

Table 11: Contamination detection results across all judge and search engine combinations.

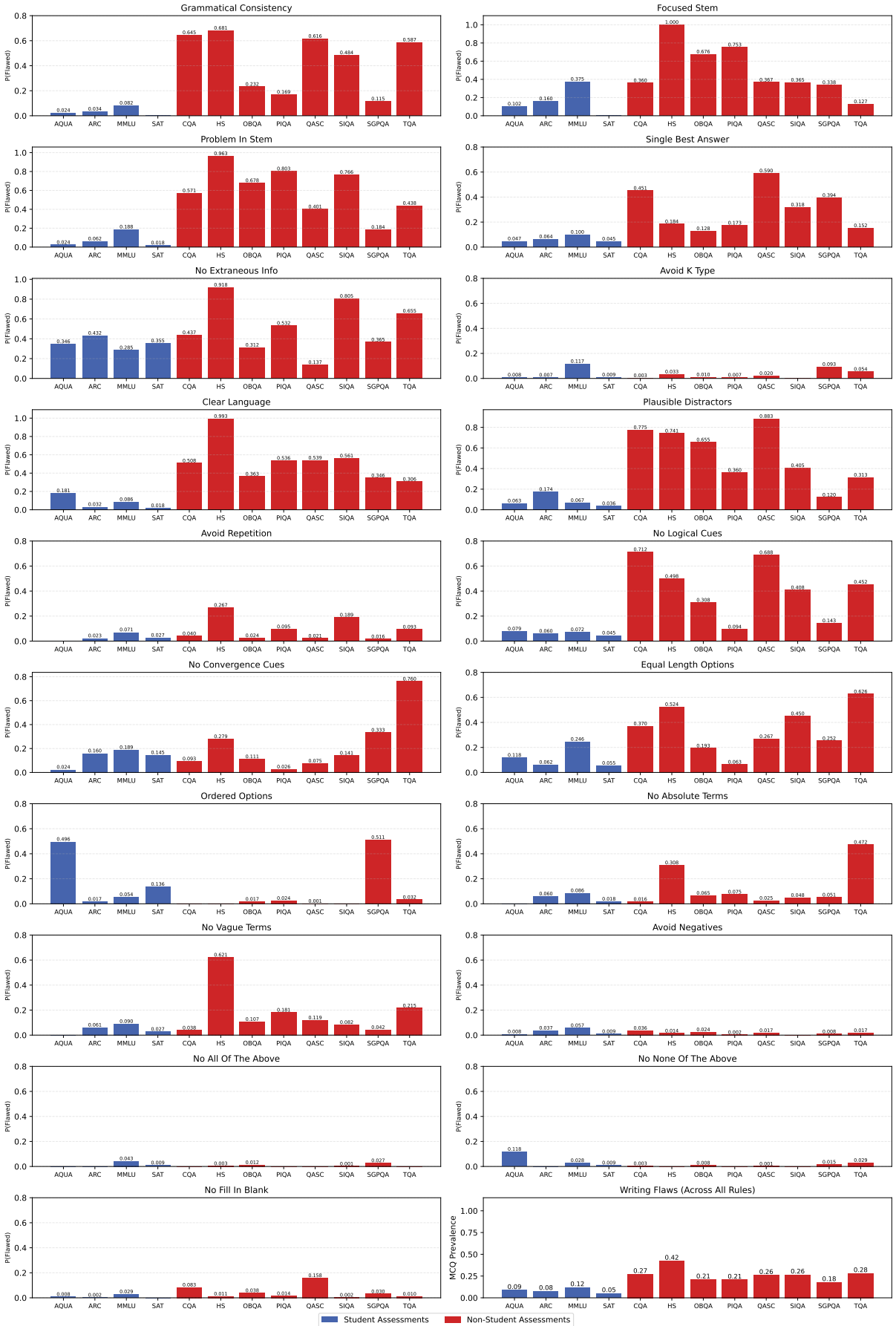

Figure 5: Scores for each of the 19 writing flaws across each MCQA benchmark.

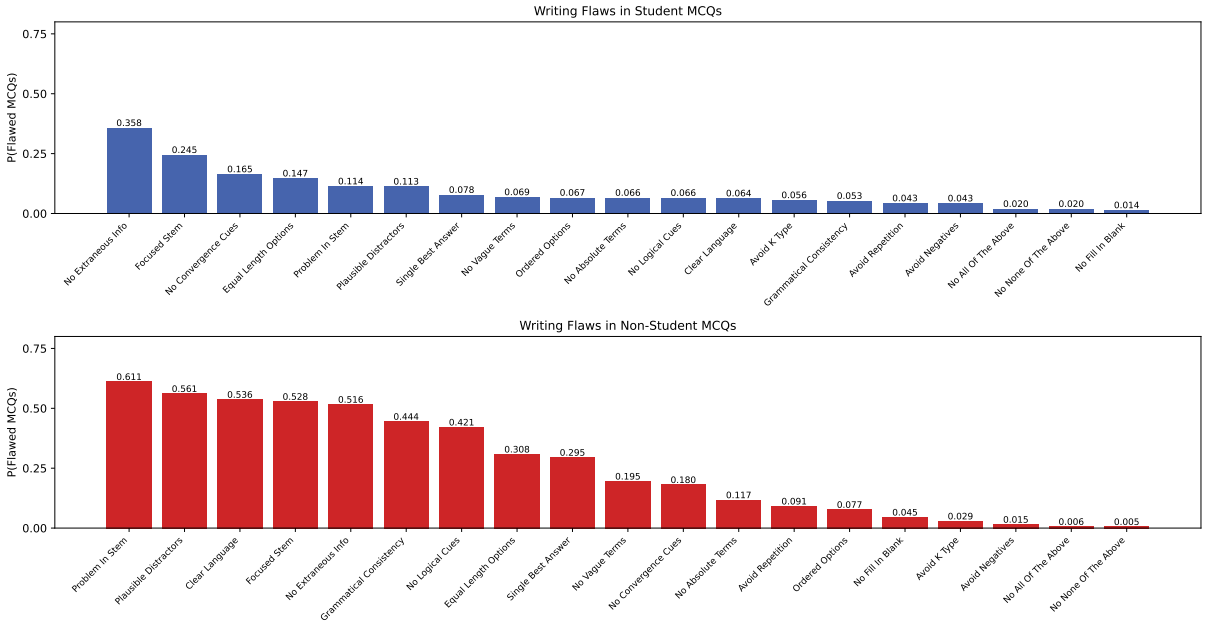

Figure 6: Descending prevalence of all 19 writing flaws across exam-based and non-exam-based MCQA benchmarks.

Pretraining Corpus	Accuracy	F1 Score	Cohen's κ
`v4_olmo-2-0325-32b-instruct_llama`	0.5590	0.6576	0.0837
`v4_dclm-baseline_llama`	0.5153	0.6626	-0.0306
`v4_dolma-v1_7_llama`	0.5284	0.5814	0.0441
`v4_rpj_llama_s4`	0.5415	0.6828	0.0238
`v4_piletrain_llama`	0.5459	0.6959	0.0257
`v4_c4train_llama`	0.5633	0.7059	0.0645

Table 12: Contamination detection agreement with human judgments when using exact question matches in pretraining corpora via Infini-Gram (Liu et al., 2024). All methods have much lower accuracy and Cohen's κ than LLM judges with search APIs.

	GPT-5				*Claude 4.5 Sonnet*				*Gemini 2.5 Pro*			
Dataset	Flaw	No Flaw	ΔAcc	ℙ(Flaw)	Flaw	No Flaw	ΔAcc	ℙ(Flaw)	Flaw	No Flaw	ΔAcc	ℙ(Flaw)
AQUA	0.737	0.761	+3.3	21%	0.771	0.751	-2.6	27%	0.789	0.743	-5.8	28%
ARC	0.835	0.883	+5.7	9%	0.862	0.882	+2.3	15%	0.793	0.884	+11.5	5%
CQA	0.659	0.785	+19.0	6%	0.771	0.778	+1.0	11%	0.779	0.777	-0.3	8%
HS	0.770	0.787	+2.1	31%	0.739	0.793	+7.3	21%	0.744	0.787	+5.8	12%
MMLU	0.743	0.801	+7.7	12%	0.773	0.798	+3.2	16%	0.742	0.800	+7.8	10%
OBQA	0.848	0.877	+3.4	23%	0.879	0.866	-1.5	32%	0.850	0.876	+3.0	22%
PIQA	0.874	0.921	+5.3	31%	0.873	0.916	+5.0	23%	0.874	0.913	+4.5	18%
QASC	0.547	0.608	+11.2	8%	0.596	0.604	+1.3	10%	0.504	0.608	+20.7	5%
SAT	0.774	0.782	+1.0	25%	0.800	0.771	-3.6	30%	0.770	0.783	+1.7	21%
SIQA	0.749	0.816	+9.0	22%	0.833	0.785	-5.7	34%	0.809	0.799	-1.3	23%
SGPQA	0.524	0.473	-9.6	12%	0.506	0.474	-6.3	17%	0.507	0.475	-6.3	13%
TQA	0.723	0.778	+7.6	9%	0.669	0.788	+17.7	13%	0.596	0.785	+31.7	6%
Micro-Average	0.765	0.766	+0.2	17%	0.781	0.762	-2.4	20%	0.763	0.767	+0.5	13%
Macro-Average	0.732	0.773	+5.5	17%	0.756	0.767	+1.5	21%	0.730	0.769	+5.4	14%

Table 13: Accuracy on flawed MCQs with shortcuts and not flawed MCQs without them, across varied choices-only models. The trend is relatively consistent per model: items with shortcuts tend to have lower accuracy, but the effect is relatively small.

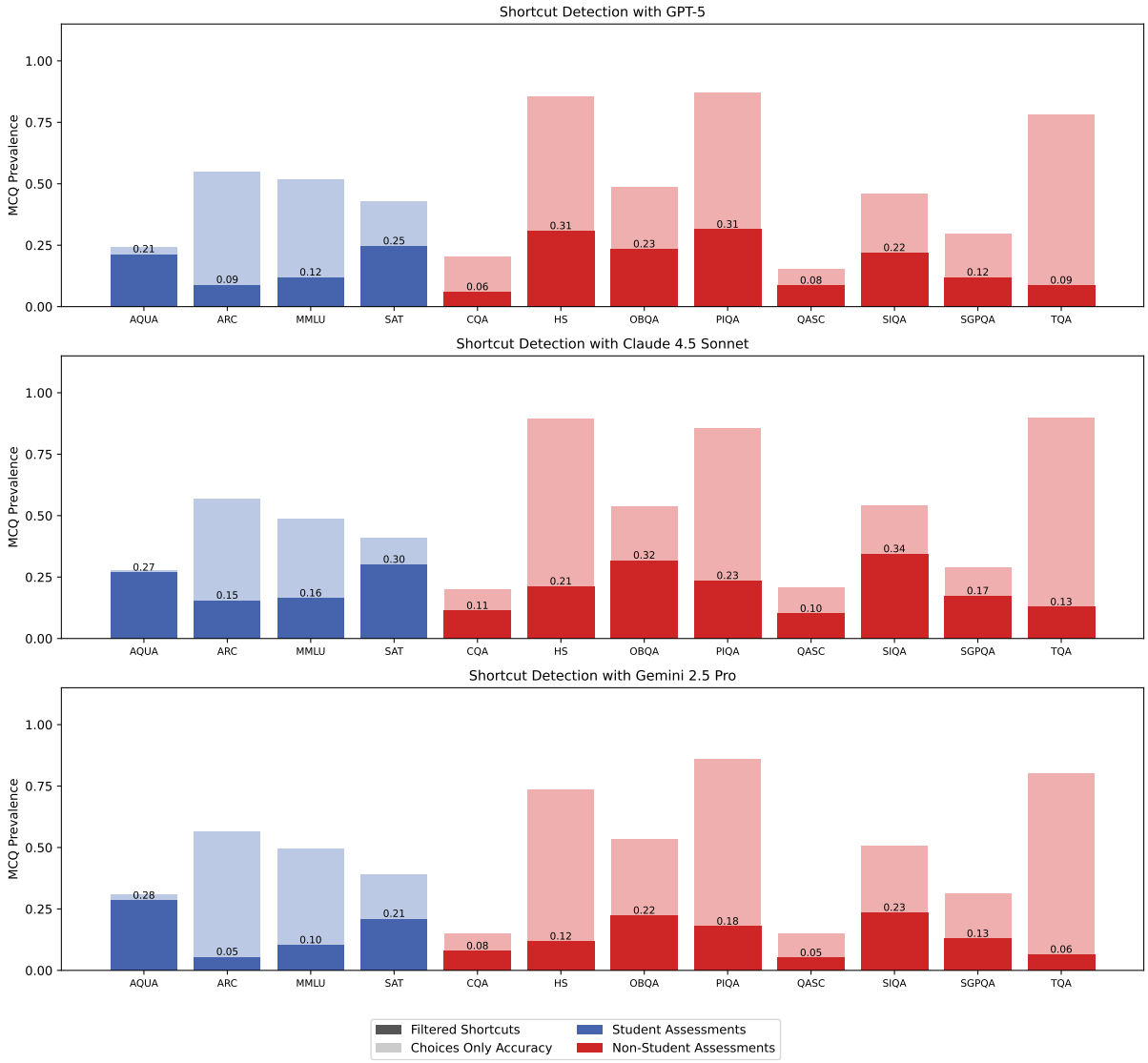

Figure 7: Shortcut prevalence on MCQ benchmarks with different models, depending on the model used to answer the MCQ with just the choices. Overall trends are relatively consistent, but there are model-specific differences, motivating our design choices of taking majority vote over these three LLMs.

	Choices-Only Success			
Dataset	Flaw	No Flaw	ΔAcc	ℙ(Flaw)
AQUA	0.746	0.759	+1.6	22%
ARC	0.899	0.853	-5.1	55%
CQA	0.840	0.767	-8.7	14%
HS	0.810	0.630	-22.2	84%
MMLU	0.823	0.765	-7.1	49%
OBQA	0.903	0.836	-7.4	51%
PIQA	0.923	0.788	-14.6	88%
QASC	0.758	0.580	-23.5	13%
SAT	0.786	0.776	-1.2	38%
SIQA	0.867	0.731	-15.6	52%
SGPQA	0.558	0.450	-19.3	27%
TQA	0.829	0.499	-39.9	83%
Micro-Average	0.845	0.690	-18.3	49%
Macro-Average	0.812	0.703	-13.4	48%

Table 14: Accuracy when shortcut flaws are defined by choices only success, without analyzing inferred questions. We reproduce the results of Gupta et al. (2025): filtering MCQs with choices-only success lowers benchmark scores.

Dataset	Tokens In / Item	Tokens Out / Item	Gemini 2.5 Flash Cost / Item (USD)	Gemini 2.5 Pro Cost / Item ($)
AQUA	18949	6623	0.02	0.12
ARC	18836	5337	0.02	0.10
CQA	18109	5675	0.02	0.10
HS	20584	7340	0.02	0.13
MMLU	19549	5848	0.02	0.11
OBQA	18182	5515	0.02	0.10
PIQA	18558	4916	0.02	0.10
QASC	18333	7268	0.02	0.12
SAT	19592	6251	0.02	0.11
SIQA	18252	5326	0.02	0.10
SGPQA	22300	8951	0.03	0.15
TQA	19403	6189	0.02	0.11

Table 15: Token usage and estimated per-item cost across datasets for Gemini 2.5 Flash and Pro models.

Model	All	Confidence ≥ 8	Confidence ≥ 9	Confidence = 10
Command R	0.37	0.38 (99%)	0.49 (63%)	0.67 (31%)
Command R+	0.36	0.36 (99%)	0.50 (63%)	0.66 (31%)
Qwen-3 0.6B	0.00	-0.00 (30%)	0.00 (18%)	0.00 (18%)
Qwen-3 1.7B	0.16	0.08 (96%)	0.06 (89%)	0.00 (65%)
Qwen-3 4B	0.31	0.29 (90%)	0.15 (75%)	0.13 (41%)
Qwen-3 8B	0.33	0.33 (99%)	0.35 (81%)	0.33 (52%)
Qwen-3 32B	0.35	0.37 (91%)	0.43 (63%)	0.60 (29%)
Qwen-3 14B	0.34	0.36 (90%)	0.32 (63%)	0.40 (46%)
Gemma-3 4B	0.20	0.20 (98%)	0.19 (93%)	0.24 (34%)
Gemma-3 12B	0.03	0.03 (99%)	0.02 (98%)	0.02 (98%)
Gemma-3 27B	0.02	0.02 (99%)	0.02 (99%)	0.02 (99%)
LLaMA-3.2 1B	0.03	0.04 (71%)	0.02 (39%)	-0.00 (12%)
LLaMA-3.2 3B	0.16	0.14 (94%)	0.14 (82%)	0.11 (67%)
LLaMA-3.1 8B	0.19	0.19 (99%)	0.26 (60%)	0.38 (39%)
LLaMA-3.1 70B	0.22	0.22 (99%)	0.30 (56%)	0.26 (47%)

Table 16: Open-weight judge Cohen's κ when only considering predictions above varied confidence thresholds. Values in parentheses indicate the proportion of samples retained at each threshold. Some open-weight models are well-calibrated, with Command R reaching Cohen's κ as high as 0.67 when giving a confidence score of 10.

Writing Error Type	Cohen's κ (↓)
no_none_of_the_above	0.9437
no_all_of_the_above	0.8242
ordered_options	0.7379
avoid_k_type	0.5921
avoid_negatives	0.5903
no_fill_in_blank	0.4706
plausible_distractors	0.4044
no_logical_cues	0.3830
problem_in_stem	0.3671
grammatical_consistency	0.3652
no_absolute_terms	0.3576
clear_language	0.3496
equal_length_options	0.3391
focused_stem	0.3117
single_best_answer	0.2647
no_vague_terms	0.2167
avoid_repetition	0.1602
no_extraneous_info	0.1317
no_convergence_cues	0.0904

Table 17: Cohen's κ breakdown for Cohere Command-R across writing error types. Lower values indicate weaker agreement. Command-R excels in simpler judgments like detecting certain options and option order, but disagrees with experts more on complex, subjective criteria like extraneousness, repetition, and convergence.

Prompt A.1: Multiple-Choice Question Answering Prompt (§4.2)

Answer the following multiple choice question. The last line of your response should be of the following format: 'ANSWER: LETTER' (without quotes) where LETTER is one of `[ letters ]`. Think step by step before answering.

`[ question]`
`[ choices ]`

Prompt A.2: Contamination Detection (§2.1)

You are an expert evaluator of dataset contamination for multiple-choice questions (MCQs). You are given a set of candidate source texts ("citations") and one MCQ. Your job is to determine whether the MCQ exists exactly, partially, or not at all in any of the citations.
<multiple-choice question>
Question: `[ question ]`
Correct Answer: `[ answer ]`
</multiple-choice question>

<citations>
`[ citations ]`
</citations>

Use the following criteria to determine the match type:
<matching criteria>
- "exact_match": The question and correct answer appear verbatim or nearly verbatim in at least one of the citations
- "question_match": The question appears verbatim or nearly verbatim in at least one of the citations, but not with the correct answer
- "partial_match": It is possible to come up with the correct answer to the question based on information in the citations
- "no_match": There is no information in the citations that can be used to answer the question
</matching criteria>

<general instructions>
- Use ONLY the information in the provided <citations>; ignore outside knowledge.
- When determining matches, do NOT consider punctuation or upper/lower casing.
- Check each citation independently.
- The citations index "i" is represented as <citation i></citation i>.
- Return every matching citation index in ascending order; if none match, return an empty list [].
- Provide a short, clear explanation for your decision, referencing the decisive overlaps when applicable.
</general instructions>

<format>
Return your output as valid JSON with the matching "result", the indexed "citations" that support your decision (empty list [] if "no_match"), and an "explanation" for your decision:
`[ json format ]`
Do not include anything else. </format>

Prompt A.3: Choices-Only Prompt (§2.2)

Answer the following multiple choice question just by using the choices and without access to the question. Use any strategy possible to come up with the correct answer, and then guess what the original/missing question was.

<choices>
`[choices]`
</choices>

<format>
Return your output as valid JSON with the key "answer" which is one of `letters`, "explanation" which is how you arrived at the correct answer, and "question" which is what you guess is the missing question. `[json format]`
Do not include anything else.
</format>

Prompt A.4: Question Similarity Prompt (§2.2)

You are an expert at determining whether a model was able to guess what the original multiple-choice question was just from the choices.

You will be given a multiple-choice question and the model's response. You need to determine whether the model was able to guess what the original question was just from the choices.

Here is the multiple-choice question:
<original question>
`[ question ]`
</original question>
Here is the model's response when answering just with the choices:
<response>
`[ response ]`
</response>

And the question that the model inferred
<inferred question>
`[ inferred question ]`
</inferred question>

To determine if the model successfully guessed the original question, use the following criteria:
- If the inferred question is an exact match or a semantic of the original question, return "exact_match"
- If a test-taker who knew the answer to the inferred question would likely be able to answer the original question, return "knowledge_match"
- In any other case, return "no_match"

<format>
Return your output as valid JSON with the key "decision" which denotes the type of match between the inferred question and the original question.
`[ json format ]`
Do not include anything else.
</format>

Prompt A.5: Writing Flaw Prompt (§2.3)

You are an expert evaluator of multiple-choice questions (MCQs). You are given the following writing rule:

<rule>
`[ rule ]`
</rule>

Your task is to judge whether a given MCQ **follows this rule**. Here are some guidelines for this specific rule:

<guidelines>
`[ guidelines]`
</guidelines>

<examples>
`[ examples]`
</examples>

<general instructions>
- Think carefully about whether the MCQ adheres to the rule. - If the rule is clearly followed and there are no flaws in the MCQ, return "pass".
- If the rule is clearly violated and there are no flaws in the MCQ, return "fail".
- In borderline cases where you are unsure, return "pass".
- Provide a confidence score from 1-10 for your pass/fail decision - how strongly you believe the MCQ follows or violates the rule. 1 means not at all confident and 10 means very confident.
- Provide a short, clear explanation of your reasoning.
</general instructions>

Here is the MCQ to evaluate:
<multiple-choice question>
`[ mcq ]`
</multiple-choice question>

<format>
Return your output as valid JSON in the following format:
`[ format ]`
Do not include anything else.
</format>