LLMs Show Surface-Form Brittleness Under Paraphrase Stress Tests

Juan Miguel Navarro Carranza
Stanford University
jmnavarr@stanford.edu

Abstract

Benchmark scores for Large Language Models (LLMs) can be inflated by *memorization* of test items or near duplicates. We present a simple, protocol that probes *generalization* by re-evaluating models on *paraphrased* versions of benchmark questions. Using Mistral-7B-Instruct and Qwen2.5-7B-Instruct, we measure the accuracy gap between original and paraphrased items on ARC-Easy and ARC-Challenge. Our pipeline controls decoding, enforces multiple-choice output format, and includes a robust paraphrase-cleaning step to preserve semantics. We find that paraphrasing induces a non-trivial accuracy drop (original vs. paraphrased), consistent with prior concerns about contamination and brittle surface-form shortcuts.

1 Introduction

Recent work shows that LLMs can regurgitate training data [Carlini et al., 2021], and that duplication in pretraining corpora amplifies this effect [Kandpal et al., 2022]. As a result, static benchmarks may overestimate model capability, especially when test items or near duplicates leak into training sets [Dong et al., 2024]. A complementary perspective focuses on *behavioral robustness* to surface perturbations—if a model relies on phrasing, small paraphrases can cause large performance swings [Ribeiro et al., 2020]. Code-evaluation work reinforces this: evolved or mutated prompts can sharply reduce apparent competence [Liu et al., 2023].

Contributions. (1) A protocol to measure generalization via paraphrase stress tests; (2) a paraphrase–cleaning pipeline that preserves semantics while removing formatting artifacts; (3) a fully specified setup (*Benchmark:* ARC-Easy/Challenge [Clark et al., 2018], *Models:* Mistral-7B-Instruct, Qwen2.5-7B-Instruct for answering and paraphrasing) using 4-bit inference on a single A100; (4) empirical findings that paraphrasing induces a measurable accuracy drop, suggesting residual memorization/brittleness; (5) released code for reproducibility.

2 Related Work

Memorization and privacy. Training-data extraction and memorization in LMs are well documented [Carlini et al., 2021]; deduplication mitigates leakage and privacy risk [Kandpal et al., 2022]. **Contamination and trustworthy eval.** Methods to detect contamination and separate memorization from generalization continue to emerge [Dong et al., 2024]. **Behavioral robustness.** CheckList formalizes capability/behavior tests with perturbations [Ribeiro et al., 2020]. **Evolved tests.** For code, augmented or mutated tests (e.g., EvalPlus) expose spurious success [Liu et al., 2023]. We adopt a minimal paraphrase-only variant that applies across QA benchmarks.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling **[selected for contributed talk]**.

3 Method

We assess surface–form robustness with a *paraphrase stress test*. We use *surface–form robustness* to mean that, under meaning–preserving rewordings (paraphrases, light syntactic changes, formatting tweaks), a model's predictions remain unchanged. For each multiple–choice item, we evaluate an instruction–tuned LLM once on the original question and once on a semantically equivalent paraphrase, then report the paired accuracy gap Δ between the two conditions.

3.1 Task and Data

We consider multiple–choice question answering (MCQ). Each item i comprises a question q_i, an option set $C_i = \{c_{i1}, \ldots, c_{ik_i}\}$, and a ground–truth answer letter $a_i \in \{A, \ldots\}$. Our experiments use the ARC benchmark [Clark et al., 2018] (Hugging Face distribution [Allen AI, 2025]), evaluating both ARC–EASY and ARC–CHALLENGE on their validation splits. We evaluate a fixed subset (ARC–Easy: **300**, ARC–Challenge: **299**) selected deterministically with a fixed random seed. Only questions are paraphrased; answer options C_i are kept verbatim to preserve label mappings.

3.2 Models and Inference

Roles and cross–pairing. We use two instruction–tuned decoder–only LLMs and assign disjoint roles: *Answerer* (produces the MCQ choice) and *Paraphraser* (rewrites only the question stem). To discourage style echo and self–consistency artifacts, the paraphraser is always a *different model family* than the answerer.

We evaluate `mistralai/Mistral-7B-Instruct-v0.3` [Mistral AI, 2025, Jiang et al., 2023] and `Qwen2.5-7B-Instruct` [Qwen Team, 2025, Yang et al., 2024]. The paraphraser always uses the model not acting as the answerer in that run (cross–family separation). We report both cross–pairings: (Mistral-7B-Instruct $\rightarrow$ answerer, Qwen2.5-7B-Instruct $\rightarrow$ paraphraser) and (Qwen2.5-7B-Instruct $\rightarrow$ answerer, Mistral-7B-Instruct $\rightarrow$ paraphraser).

Compute and environment. All runs are executed on a single NVIDIA A100–SXM4–80GB (CUDA 12.4, driver 550.54.15). Models are loaded via `transformers` with 4–bit quantization (`bitsandbytes` NF4) using `device_map="auto"`. Randomness is controlled via `random.seed(1337)` and `torch.manual_seed(1337)`. No few–shot examples, chain–of–thought, tools, or retrieval are used.

Quantization. We follow a standard 4–bit inference recipe: `load_in_4bit=True`, `bnb_4bit_use_double_quant=True`, `bnb_4bit_quant_type="nf4"`, and `bnb_4bit_compute_dtype=torch.bfloat16`. This yields stable, memory–efficient inference on the A100 while preserving accuracy in our setting.

Prompts. *Answering.* We format each item with the question followed by lettered options A, B, C, ... and require a *single–letter* decision. When `force_letter=True`, the instruction requires a one–object JSON: `{"answer": "LETTER", "explanation": "1-2 sentences"}`, and forbids any extra text.

Paraphrasing. We use a constrained rewrite prompt that (i) rewrites the *question stem* only, (ii) preserves all meaning and details *verbatim* (numbers/units/entities), (iii) outputs only the rewritten stem without labels or commentary.

Decoding. *Answering (deterministic):* `max_new_tokens`=32, `temperature`=0.0, `top_p`=1.0, `do_sample=False`. Determinism ensures fair comparisons across original vs. paraphrased conditions.

Paraphrasing (light sampling): `max_new_tokens`=128, `temperature`=0.7, `top_p`=0.95, `do_sample=True`. Mild diversity avoids trivial restatements while remaining close to the source; fidelity checks (below) prevent semantic drift.

Output parsing and formatting guardrails. For the answerer, we parse the JSON field `answer` when present and validate it against the option set. If the model emits residual text, we fall back to

a strict letter extractor that (i) prefers the JSON key when available, (ii) otherwise searches for a leading pattern like `"Answer: <LETTER>"` and (iii) rejects spurious matches (e.g., picking the first uppercase letter encountered). This avoids the common failure mode where generic capital letters (e.g., "A" at the start of "Answer: C") are misread as the choice. We score with exact letter agreement against the gold key.

Paraphrase fidelity checks and retries. We paraphrase the stem once and run a sequence of automatic checks; if a check fails, we *retry* up to $K{=}3$ times with the same decoding settings: (i) nonempty output after cleaning (trim quotes/bullets, strip prefixes); (ii) output differs from the original stem case–insensitively; (iii) minimum length threshold: $\mathrm{len}(\tilde{q}) \geq \max(10,\ 0.6\,\mathrm{len}(q))$ to avoid fragmentary rewrites. Only stems that pass all checks are accepted; otherwise we fall back to the original stem to keep the item count fixed. Answer options are *never* paraphrased to preserve label mappings.

Dataset slice and evaluation protocol. Unless stated otherwise, we evaluate ARC validation splits with a fixed, deterministic subset size (e.g., **300** for ARC-EASY, **299** for ARC-CHALLENGE). For each item, let r_i^{orig} and r_i^{para} be predictions on original and paraphrased questions, respectively. Define correctness $o_i = \mathbb{I}[r_i^{\text{orig}}{=}a_i]$ and $p_i = \mathbb{I}[r_i^{\text{para}}{=}a_i]$. We report

$$\mathrm{Acc}_{\text{orig}} = \frac{1}{n}\sum_{i=1}^{n} o_i, \quad \mathrm{Acc}_{\text{para}} = \frac{1}{n}\sum_{i=1}^{n} p_i, \quad \Delta = \mathrm{Acc}_{\text{orig}} - \mathrm{Acc}_{\text{para}}.$$

4 Results

Overall trends. Across both ARC splits, paraphrasing consistently reduces accuracy ($\Delta > 0$). Table 1 shows drops ranging from 0.06 to 0.10, confirming that surface–form changes measurably degrade performance even when semantic content is preserved. The effect is robust across cross–pairings: regardless of whether Mistral-7B-Instruct or Qwen2.5-7B-Instruct is the answerer, accuracy on paraphrased items is lower than on the original items.

Dataset difficulty. Absolute accuracy is higher on ARC-EASY than on ARC-CHALLENGE for both models, consistent with the benchmark design. However, relative brittleness is not uniform: the largest drop occurs for Mistral-7B-Instruct answering ARC-EASY ($\Delta = 0.10$), while both models show more moderate drops on ARC-CHALLENGE ($\Delta = 0.06$–0.07). This suggests that even "easier" items are fragile under rewording.

Cross–model comparison. When acting as the answerer, Qwen2.5-7B-Instruct achieves higher baseline accuracy (0.90 on Easy, 0.89 on Challenge) than Mistral-7B-Instruct (0.84 and 0.75 respectively). Yet both exhibit similar paraphrase sensitivity (Δ in the same 0.06–0.10 band), indicating that brittleness is not confined to a single model family but is a shared vulnerability.

Qualitative flips. We observe two categories of changes: (i) *original→incorrect*, where a paraphrase induces an error despite a correct original response; (ii) *incorrect→correct*, where paraphrasing helps the model recover the right answer. The former dominates, but the latter occurs in a nontrivial minority of cases, highlighting that paraphrasing does not simply act as uniform noise but can also reshape decision boundaries in helpful ways.

Table 1: Accuracy on original vs. paraphrased items using Mistral-7B-Instruct and Qwen2.5-7B-Instruct. Δ quantifies brittleness to paraphrase.

Answerer	Paraphraser	Dataset	n	Acc (orig)	Acc (para)	Δ (drop)
Qwen2.5-7B-Instruct	Mistral-7B-Instruct	ARC-Easy	**300**	**0.90**	**0.84**	**0.06**
Qwen2.5-7B-Instruct	Mistral-7B-Instruct	ARC-Challenge	**299**	**0.89**	**0.83**	**0.07**
Mistral-7B-Instruct	Qwen2.5-7B-Instruct	ARC-Easy	**300**	**0.84**	**0.74**	**0.10**
Mistral-7B-Instruct	Qwen2.5-7B-Instruct	ARC-Challenge	**299**	**0.75**	**0.69**	**0.06**

5 Discussion

Contamination vs. robustness. The observed accuracy drop under paraphrase indicates that models may be relying on brittle surface–form patterns rather than robust semantic generalization. This raises the question of contamination: if benchmark items or near duplicates appeared in pretraining corpora, high performance on the original phrasing may reflect memorization rather than reasoning. Paraphrasing disrupts such surface matches, exposing whether the model has internalized underlying concepts or simply memorized familiar strings. While our study does not perform explicit contamination auditing, methods such as n-gram overlap checks or retrieval-based similarity [Dong et al., 2024] could strengthen causal attribution in future work.

Instruction tuning sensitivity. Instruction tuning strongly conditions both the answerer and the paraphraser. On the answering side, small prompt variations can flip predictions, suggesting that instruction templates may inadvertently favor one phrasing style over another. On the paraphrasing side, instruction tuning interacts with the model's generative priors, sometimes yielding paraphrases that are formally valid but less faithful semantically. Thus, prompt design and instruction alignment are not neutral components of the pipeline, but active factors shaping robustness outcomes.

Paraphrase fidelity. Despite targeted prompts and automated cleaning, occasional semantic drift occurs in paraphrases. Manual inspection revealed cases where models introduced or omitted information, which neither cleaning heuristics nor retries could fully correct. Because the framework relies on an LLM to generate paraphrases, paraphrase quality becomes a limiting factor for evaluation fidelity. Improving this component—for example by using human-in-the-loop filtering, specialized paraphrasing models, or multi-pass verification—would increase confidence that measured drops stem from answerer brittleness rather than paraphrase artifacts.

Joint influence of answerer and paraphraser. By design, the paraphraser is always from a different model family than the answerer, avoiding leakage through stylistic self-imitation. However, this also means that results reflect the *interaction* of two models, not the answerer in isolation. If the paraphraser generates awkward or biased rewrites, measured brittleness may partially reflect its limitations. Future protocols could disentangle these roles more cleanly, for example by evaluating answerers against a fixed, high-fidelity paraphrase set.

Format dependence. Forcing a letter-only output format simplifies scoring but also interacts with instruction tuning. Some errors appear to arise from rigid formatting constraints, rather than from the model's underlying knowledge. Although our strict parser reduces spurious matches, output-format sensitivity highlights that evaluation design decisions can influence reported robustness.

Scope and generalization. Our study is limited to the ARC benchmark, which targets grade-school science. While useful for controlled analysis, these results may not directly transfer to broader tasks such as open-domain QA, multi-turn dialogue, or reasoning-intensive benchmarks. Extending paraphrase stress tests across domains and task formats is an important direction for future work.

6 Conclusion

We evaluated two 7B instruction models (Mistral-7B-Instruct, Qwen2.5-7B-Instruct) under a paraphrase stress test and found consistent accuracy drops of 6–10 points on ARC–Easy and ARC–Challenge. This indicates that strong benchmark scores may partly reflect memorization or reliance on brittle surface patterns rather than robust reasoning. The results underscore the importance of paraphrase-aware evaluation and point to future work on higher-fidelity paraphrasing, contamination auditing, and extending stress tests beyond ARC to broader tasks and domains.

Acknowledgments and Disclosure of Funding

Thanks to the open-source community for releasing models, datasets, and tools that make public-only evaluations feasible.

References

Allen AI. allenai/ai2_arc (Hugging Face Datasets), 2025. URL https://huggingface.co/datasets/allenai/ai2_arc.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting Training Data from Large Language Models, June 2021. URL http://arxiv.org/abs/2012.07805. arXiv:2012.07805 [cs].

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge, March 2018. URL http://arxiv.org/abs/1803.05457. arXiv:1803.05457 [cs].

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. Generalization or Memorization: Data Contamination and Trustworthy Evaluation for Large Language Models, May 2024. URL http://arxiv.org/abs/2402.15938. arXiv:2402.15938 [cs].

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, October 2023. URL http://arxiv.org/abs/2310.06825. arXiv:2310.06825 [cs].

Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating Training Data Mitigates Privacy Risks in Language Models, December 2022. URL http://arxiv.org/abs/2202.06539. arXiv:2202.06539 [cs].

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation, October 2023. URL http://arxiv.org/abs/2305.01210. arXiv:2305.01210 [cs].

Mistral AI. mistralai/Mistral-7B-Instruct-v0.3 (Model Card), 2025. URL https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3.

Qwen Team. Qwen/Qwen2.5-7B-Instruct (Model Card), 2025. URL https://huggingface.co/Qwen/Qwen2.5-7B-Instruct.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond Accuracy: Behavioral Testing of NLP models with CheckList, May 2020. URL http://arxiv.org/abs/2005.04118. arXiv:2005.04118 [cs].

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 Technical Report, September 2024. URL http://arxiv.org/abs/2407.10671. arXiv:2407.10671 [cs].