Artifact or Flaw? Rethinking Prompt Sensitivity in Evaluating LLMs

Anonymous ACL submission

Abstract

Prompt sensitivity, referring to the phenomenon where minor variations in phrasing lead to significant changes in large language model (LLM) performance, has been widely accepted as a core limitation of LLMs. In this work, we revisit this issue and ask: Is the widely reported high prompt sensitivity truly an inherent weakness of LLMs, or is it largely an artifact of evaluation processes? To answer this question, we systematically evaluate 7 LLMs (e.g., GPT and Gemini family) across 6 benchmarks, including both multiple-choice and open-ended tasks on 12 diverse prompt templates. We find that much of the prompt sensitivity stems from heuristic evaluation methods, including log-likelihood scoring and rigid answer matching, which often overlook semantically correct responses expressed through alternative phrasings, such as synonyms or paraphrases. When we adopt LLM-as-judge evaluations, we observe a substantial reduction in performance variance and a consistently higher correlation in model rankings across prompts. Our findings suggest that modern LLMs are more robust to prompt templates than previously believed, and that prompt sensitivity may be more an artifact of evaluation than a flaw in the models.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success across a wide range of tasks (Hua et al., 2024; Liévin et al., 2024). Moreover, LLMs are good at following diverse instructions, so that users are not required to follow a fixed template when asking a question. This has led to concerns about prompt sensitivity, where differences in prompt phrasing can substantially affect benchmark performance, casting doubt on the reliability of evaluations (Polo et al., 2024; Mizrahi et al., 2024; Chatterjee et al., 2024; Sclar et al., 2023). More critically, the relative rankings of LLMs can shift substantially depending on the prompt template used (Polo et al., 2024; Mizrahi et al., 2024). For example, simply changing the option format from letters (*e.g.*, “A:”) to numbers (*e.g.*, “(1)”), completely *reverses* the ranking order of four evaluated open-source models in ARC-Challenge (Clark et al., 2018).

Although existing studies have reported that LLMs are highly sensitive to prompt phrasing (Voronov et al., 2024; Mizrahi et al., 2024), this remains counterintuitive given that instruction-tuned LLMs are explicitly optimized to handle a wide range of input formats. For example, instruction-tuning datasets such as FLAN (Longpre et al., 2023) and Super-NaturalInstructions (Wang et al., 2022) include a diverse collection of tasks (e.g., question answering, summarization, classification) with varying natural language prompt templates (Zhang et al., 2023). This contradiction raises a critical question:

Is prompt sensitivity an inherent flaw in LLMs, or merely an artifact of the evaluation process?

To investigate this, we find that previous studies (Voronov et al., 2024; Chatterjee et al., 2024) typically rely on heuristic evaluation, such as regular-expression-based answer extraction or log-likelihood scoring over candidates. These heuristic evaluation approaches, though historically popular due to their simplicity (Zellers et al., 2019; Reddy et al., 2019; Brown et al., 2020; Hendrycks et al., 2021a), may introduce errors when model outputs deviate from expected formats. More specifically, models may generate correct answers, but because their outputs are not aligned with the rigid evaluation format, they are mistakenly marked as incorrect. This issue becomes more pronounced as models have become more open-ended and diverse in their output formats (Hurst et al., 2024; Yang et al., 2025), potentially leading to inflated estimates of prompt sensitivity (Figure 1).

To rigorously assess whether LLMs truly suffer from prompt sensitivity, we revisit this issue using a more robust evaluation strategy: LLMs as judges.

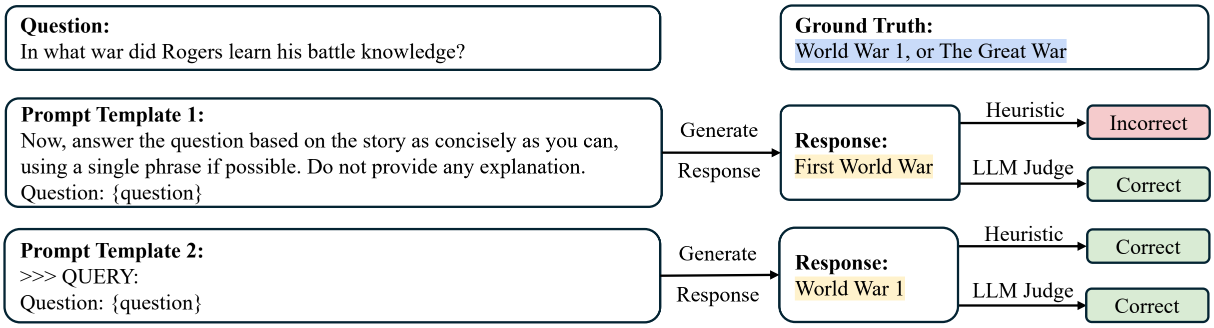

Figure 1: When provided with diverse prompt templates, LLMs provide different but semantically equivalent responses. Heuristic evaluation fails to match the different answers with the ground truth, exaggerating prompt sensitivity. In contrast, an LLM judge is able to identify the semantic equivalence consistently.[1]

Now widely adopted in recent benchmarks (Wei et al., 2024; Zheng et al., 2023), this approach shifts evaluation from rigid pattern-matching to semantic assessment, enabling a more reliable examination of prompt sensitivity. Compared to heuristics, LLM judges can better handle various output formats, paraphrasing, and ambiguous cases, making them more aligned with human evaluation (Liu et al., 2023; Kocmi and Federmann, 2023; Zheng et al., 2023; Wang et al., 2023).

Using both heuristic methods and LLM-as-judge, we conduct a comprehensive evaluation of four open-source and three closed-source LLMs across 12 prompt templates (without cherry-picking) and six diverse benchmarks, including both multiple-choice and open-ended generation tasks. We discover that heuristic evaluation methods often **exaggerate** the prompt sensitivity of LLMs. For instance, on ARC-Challenge, the performance of Gemma-2.0 varies widely across prompts, with accuracy ranging from 0.25 to 0.90 and a high standard deviation of 0.28. In contrast, when evaluated using LLM-as-judge, its accuracy varies by only 0.17 across the same set of prompt templates, with a much lower standard deviation of just 0.005. Furthermore, the *Spearman rank correlation* measuring model performance rankings across the four open-source models remarkedly improves from 0.31 under heuristic evaluation to 0.92 with LLM-as-judge. These findings suggest that previously reported prompt sensitivity may be significantly overstated due to the limitations of heuristic evaluation. When evaluated with LLM-as-judge, models show far more consistent performance and stable rankings across prompts, indicating that prompt sensitivity is largely an artifact of the evaluation method rather than an inherent flaw in LLMs.

[1]This is an example from NarrativeQA. Heuristic method uses word-level F1; "Incorrect" is shown here for illustration purposes, indicating a lower score than a correct answer.

2 Method

Our method consists of three main components: (1) diverse prompt template construction (Sec 2.1), (2) LLM-as-judge evaluation (Sec 2.2), and (3) prompt template sensitivity measurement (Sec 2.3).

2.1 Diverse Prompt Template Construction

To evaluate LLM sensitivity to prompt phrasing, we construct diverse prompt templates for each benchmark. These templates vary in instruction wording, answer formatting (e.g., using letters vs. numbers), and how responses are requested, while keeping the task content unchanged (Appendix A).

In practice, we use GPT-4o to paraphrase the original prompts. For multiple-choice datasets, we create a shared pool of 12 diverse templates used across all benchmarks. For open-ended generation tasks, we generate 12 templates per benchmark to better accommodate domain-specific styles.

2.2 LLM-as-judge evaluation

Heuristic evaluation methods often fail when model outputs deviate from expected formats. To address this limitation, we adopt LLMs as robust judges. In this approach, an LLM judge is given the original question, the correct answer, and the model's predicted response. The judge is prompted to determine whether the predicted response semantically matches the correct answer (Appendix B).

While the overall format remains consistent, we introduce minor benchmark-specific adjustments to the judging prompt. For example, for GPQA, the judge is instructed to "Ignore all explanation" to ensure it focuses solely on the final answer.

2.3 Prompt Sensitivity Measurement

We measure sensitivity with two metrics: *performance variation* and *ranking consistency*.

Performance Variation. For each model and dataset, we compute the accuracy under every prompt template and report the standard deviation across all prompt variants. Let $P = \{p_1, p_2, \ldots, p_n\}$ denote the set of prompt templates for a given benchmark D, and let f be the model under evaluation. The performance of model f under prompt p_i is denoted by $A^{p_i}_{f,D}$. The prompt sensitivity of model f on dataset D is then quantified as: $\mathrm{std}_f = \mathrm{StdDev}\left(\left\{A^{p_i}_{f,D}\right\}_{i=1}^{n}\right)$. A lower standard deviation indicates that the model's performance is stable across different prompt templates.

Ranking Consistency. Beyond absolute performance, we also measure how model rankings vary across prompt templates. Given a set of K models, we rank them based on their performance under each prompt and compute pairwise *Spearman's rank correlation* between all prompt pairs (Spearman, 1904). Given two templates p_i and p_j, and the corresponding performance vectors $\{A^{p_i}_{f_k,D}\}_{k=1}^{K}$ and $\{A^{p_j}_{f_k,D}\}_{k=1}^{K}$, we calculate Spearman's rank correlation coefficient $\rho_{ij} = 1 - \frac{6\sum_{k=1}^{K} d_k^2}{K(K^2-1)}$, where d_k is the difference in rankings of the k-th model under prompts p_i and p_j, and K is the number of models. The rank correlation coefficient, ρ, ranges from -1 to 1, with higher values indicating stronger agreement in ranking consistency.

To measure overall ranking consistency, we compute the mean Spearman's rank correlation coefficient, denoted as $\bar{\rho}$, across all pairs of prompt templates. This mean score $\bar{\rho}$ serves as a comprehensive metric for evaluating the stability of model rankings under prompt variation. A higher $\bar{\rho}$ suggests that evaluations are more robust and less dependent on the specific prompt phrasing.

3 Results and Discussion

3.1 Experimental Setup

Models. We evaluate LLaMA-3.1-8B-Instruct (LLaMA-3.1) (Dubey et al., 2024), Qwen2-7B-Instruct (Qwen-2) (Yang et al., 2024), Gemma-2-9B-it (Gemma-2) (Team et al., 2024), Ministral-8B-Instruct-v0.2 (Ministral) (MistralAI, 2024), GPT-4o-mini (July 2024), GPT-4.1-mini (April 2025), and Gemini 2.0 Flash (February 2025).

Dataset	$\bar{\rho}_{\text{Heuristic}}$	$\bar{\rho}_{\text{LLM}}$
ARC-Challenge*	0.3036	0.9546 (0.9187)
OpenbookQA*	0.4212	0.9386 (0.7360)
GPQA Diamond*	0.1542	0.8960 (0.5048)
NarrativeQA†	0.5927	0.8662
MATH	0.9593	0.9647
SimpleQA	–	0.8121

Table 1: Average Spearman rank correlation ($\bar{\rho}$) across prompt templates using heuristic evaluation vs. LLM-as-judge. *Heuristic results are based on 4 open-source models. LLM-as-judge uses all 7 models, with 4-model subset scores shown in parentheses for comparison. †Due to context length limitations, only LLaMA-3.1 and 3 proprietary models are used for NarrativeQA evaluation. – indicates heuristic evaluation is not applicable.

Benchmarks. We evaluate on six benchmarks covering both multiple-choice and open-ended tasks. The multiple-choice datasets include ARC-Challenge (Clark et al., 2018), GPQA-diamond (Rein et al., 2024), and OpenbookQA (Mihaylov et al., 2018), where answers are selected from discrete options (e.g., A/B/C/D). For these tasks, heuristic evaluation uses log-likelihood scoring over answer options. The open-ended datasets include NarrativeQA (Kočiský et al., 2018), MATH (Hendrycks et al., 2021b), and SimpleQA (Wei et al., 2024), where model responses are free-form. For NarrativeQA and MATH, heuristic evaluation applies format-specific extraction and normalization (see Appendix C). For SimpleQA, no rule-based parser is available, so we report results only under the LLM-as-judge framework.

All evaluations use greedy decoding to ensure deterministic outputs.

3.2 Heuristic Evaluation Exaggerates Prompt Sensitivity of LLMs

When comparing the performance of the model under heuristic evaluation and LLM-as-judge, we find that the heuristic methods exhibit significantly greater sensitivity to prompt variation (Figure 2). On ARC-Challenge, all open-source models except Qwen-2 show much higher standard deviations under heuristics. For instance, Gemma-2.0 yields a deviation of 0.28, versus just 0.005 with LLM-as-judge. Its accuracy range spans 0.25–0.90 under heuristics, compared to only 0.17 with LLM-as-judge. Additionally, mean accuracy improves under LLM-as-judge, suggesting heuristic methods often miss valid answers due to overly rigid extraction rules.

Beyond variance in accuracy, we also assess

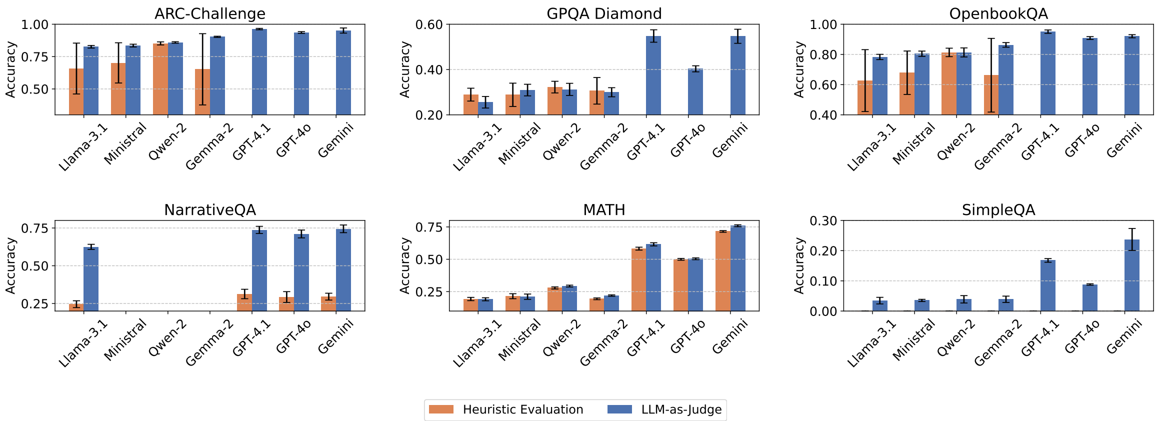

Figure 2: The mean and standard deviation of performance across different prompt templates. For all 6 datasets, we show the statistics for all pairs of evaluation methods and models, excluding the cases when the model's context length is not enough for the task or when the heuristic evaluation method is not available. The standard deviation of the LLM-as-judge method is always low.

ranking consistency across prompts (Sec 2.3). On ARC-Challenge, the average Spearman rank correlation across prompts among open-source models increases from 0.30 (heuristics) to 0.92 (LLM-as-judge), and further to 0.95 when proprietary models are included. On NarrativeQA, the correlation rises from 0.40 (heuristics) to 0.87 (LLM-as-judge). These findings suggest that prompt sensitivity observed in prior work is largely an artifact of heuristic evaluation, not an inherent flaw of LLMs.

Well-designed heuristic methods show low prompt sensitivity similar to LLM-as-judge. For MATH (Hendrycks et al., 2021b), the heuristic approach incorporates symbolic simplification, expression normalization, and equivalence checking using tools such as `sympy`. Under these conditions, we observe prompt sensitivity results that are comparable to those obtained via LLM-as-judge evaluation, with similarly low accuracy variance and high ranking consistency. These results indicate that, with sufficient domain-specific prompt engineering, heuristic methods can provide stable evaluations. This further supports the conclusion that modern LLMs exhibit less prompt-induced variance than previously reported.

Recent benchmarks also exhibit low prompt sensitivity. We extend the evaluation to SimpleQA (Wei et al., 2024), a newly proposed benchmark for factual and commonsense reasoning. As no official heuristic evaluation is available, we use LLM-as-judge by default. Applying our prompt sensitivity analysis, we observe a low standard deviation and a high Spearman rank correlation of 0.8121 across 12 prompt templates. These results indicate that model performance remains stable across different prompt variations.

4 Related Work

The ranking inconsistency with diverse prompt templates has been widely reported (Polo et al., 2024; Mizrahi et al., 2024; Chatterjee et al., 2024; Sclar et al., 2023). Mizrahi et al. (2024) conducted a large-scale study showing significant accuracy differences across prompt variants. Voronov et al. (2024) further showed that no prompt format consistently performs best across models. To address this, prior work often assumes that LLMs are inherently unstable to prompt changes. For example, Polo et al. (2024) estimates the distribution of accuracy across prompts to improve evaluation efficiency. However, all existing methods attribute the sensitivity to model behavior. In contrast, we show that a key factor is the heuristic evaluation protocol itself, which often leads to misclassification of correct outputs and overstates prompt sensitivity.

5 Conclusion

In this work, we demonstrate that much of the observed prompt sensitivity in LLM evaluations is not due to inherent model weaknesses, but rather an artifact introduced by heuristic evaluation methods. Through comprehensive experiments using LLM-as-judge across multiple benchmarks and prompt templates, we reveal that model performance and rankings are substantially more stable and reliable than previously reported. We hope this work sheds light on prompt sensitivity in LLM evaluation and encourages broader adoption of LLM-as-judge to evaluate the true capabilities of LLMs.

Limitations

Due to computational constraints, we evaluate each benchmark using only 12 prompt templates. However, we find that results are stable across scales. For example, on ARC-Challenge, the ranking consistency and variance metrics using 12 prompts closely match those obtained with over 100 prompts, suggesting that our analysis is representative.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Anwoy Chatterjee, HSVNS Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2024. Posix: A prompt sensitivity index for large language models. *arXiv preprint arXiv:2410.02185*.

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:1803.05457v1*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Andong Hua, Mehak Preet Dhaliwal, Ryan Burke, Laya Pullela, and Yao Qin. 2024. Nutribench: A dataset for evaluating large language models on nutrition estimation from meal descriptions. *arXiv preprint arXiv:2407.12843*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Victor Liévin, Christoffer Egeberg Hother, Anna Guldberger Motzfeldt, and Ole Winther. 2024. Can large language models reason about medical questions? *Patterns*, 5(3):100943.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and 1 others. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.

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 *EMNLP*.

MistralAI. 2024. Ministral-8b-instruct-2410. `https://huggingface.co/mistralai/Ministral-8B-Instruct-2410`.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Felipe Maia Polo, Ronald Xu, Lucas Weber, Mírian Silva, Onkar Bhardwaj, Leshem Choshen, Allysson Flavio Melo de Oliveira, Yuekai Sun, and Mikhail Yurochkin. 2024. Efficient multi-prompt evaluation of llms. *arXiv preprint arXiv:2405.17202*.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

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*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*.

Charles Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101. JSTOR 1412159.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements. *arXiv preprint arXiv:2401.06766*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is ChatGPT a good NLG evaluator? a preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 1–11, Singapore. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, and 1 others. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

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.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and 1 others. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Diverse Prompts

In this section, we list the diverse prompt templates we use for each benchmark.

For ARC-Challenge, GPQA, and OpenbookQA, we use the following 12 prompt templates:

1. Evaluate the choices and select the most appropriate answer.\n{question}\nThe options are as follows:\nOption A: {first_option}\nOption B: {second_option}\nOption C: {third_option}\nOption D: {fourth_option}\nYour answer should be formatted as:\n'I have chosen option [choice]'\nwhere [choice] is your selected answer.\n

2. Review the available options and select the one you think is correct.{question}\nAvailable answers include:\nA) {first_option}\nB) {second_option}\nC) {third_option}\nD) {fourth_option}\n\nResponse:

3. Select the correct answer based on your understanding.\n{question}\nPick from the following options:\n[A] {first_option}\n[B] {second_option}\n[C] {third_option}\n[D] {fourth_option}\nPlease respond with 'Option [choice]'.\n

4. Evaluate the options presented and select the most suitable.{question}\nAvailable answers:\n[A] {first_option}\n[B] {second_option}\n[C] {third_option}\n[D] {fourth_option}\n\nExpress your choice as: 'The answer is [choice].'\nwhere [choice] is your selected option.\n

5. Based on the question presented, choose the most fitting response.{question}\nAvailable answers are:\nA: {first_option}\nB: {second_option}\nC: {third_option}\nD: {fourth_option}\nPlease provide your response in the following format:\n’Your choice: [option]’\nwhere [option] corresponds to the letter or number you selected.\n

6. From the options below, select the response that you believe is correct.{question}\nChoices to consider:\n1. {first_option}\n2. {second_option}\n3. {third_option}\n4. {fourth_option}\nResponse:

7. Select your answer from the provided list of options.\n{question}\nOptions are:\nThe choice is A: {first_option}\nThe choice is B: {second_option}\nThe choice is C: {third_option}\nThe choice is D: {fourth_option}\n\nChoose your answer:

8. After considering the options, choose the best possible answer.{question}\nThe following choices are available:\nA: {first_option}\nB: {second_option}\nC: {third_option}\nD: {fourth_option}\nState your answer as:\n’Answer: [choice]’\n

9. Analyze the selections and provide your choice.\n{question}\nYour options are listed below:\nOption 1 - {first_option}\nOption 2 - {second_option}\nOption 3 - {third_option}\nOption 4 - {fourth_option}\n\nYour response:

10. Consider the following question and determine the right response.\n{question}\nWhich of the following answers do you prefer?\nOption 1: {first_option}\nOption 2: {second_option}\nOption 3: {third_option}\nOption 4: {fourth_option}\nI select:

11. Determine which option best answers the question asked.\n{question}\nPossible choices are as follows:\nOption [A] {first_option}\nOption [B] {second_option}\nOption [C] {third_option}\nOption [D] {fourth_option}\n\nFinal answer:

12. Identify the option that best answers the question posed.{question}\nConsider these choices:\nSelect option 1: {first_option}\nSelect option 2: {second_option}\nSelect option 3: {third_option}\nSelect option 4: {fourth_option}\n\nChoice provided:

In the prompt templates, {question} is the question, and {first_option}, {second_option}, {third_option}, and {fourth_option} are the options.

For NarrativeQA, we use the following 12 prompt templates:

1. You are given a story, which can be either a novel or a movie script, and a question. Answer the question asconcisely as you can, using a single phrase if possible. Do not provide any explanation.\n\n Story: {context}\n\n Now, answer the question based on the story as concisely as you can, using a single phrase if possible. Do not provide any explanation.\n\n Question: {question}\n\n Answer:

2. Below is an excerpt from a mystery or thriller story, followed by a question. Provide the most accurate answer you can in a single phrase or sentence fragment. No elaboration is needed.\n\nStory: {context}\n\nExamine the situation carefully and respond.\n\nQuestion: {question}\n\nAnswer:

3. You are presented with a passage from literary fiction or cinematic writing and a comprehension question. Respond succinctly with a phrase. Avoid any additional commentary.\n\nStory: {context}\n\nAnalyze and respond concisely.\n\nQuestion: {question}\n\nAnswer:

4. A tale from a distant world or magical land is told below, followed by a question from a curious scholar. Give your answer using only a few words. No need to explain the lore.\n\nStory: {context}\n\nWhat say you?\n\nQuestion: {question}\n\nAnswer:

5. You’re reading a gritty tale from the backstreets of the city. A question follows. Keep your answer clipped, clean, and under the radar—just a phrase, no fluff.\n\nStory:

{context}\n\nHere's the case:\n\nQuestion: {question}\n\nAnswer:

6. Welcome to *Plot Points*! We'll give you a story snippet and a question—your job is to give the fastest, most precise answer possible. One phrase, no lifelines!\n\nStory: {context}\n\nLet's play!\n\nQuestion: {question}\n\nAnswer:

7. The record shows the following account. A question will now be entered into the record. Provide your answer in a short, factual phrase. No commentary permitted.\n\nStory: {context}\n\nDeposition Question:\n\nQuestion: {question}\n\nAnswer:

8. Read the excerpt. Answer the question. Keep it short.\n\nStory: {context}\n\nQuestion: {question}\n\nAnswer:

9. Accessing archive… Story fragment retrieved from Galactic Chronicles. A query follows. Respond with the most relevant concept or phrase. Do not explain.\n\nStory: {context}\n\n>>> QUERY:\n\nQuestion: {question}\n\n>>> RESPONSE:\n\nAnswer:

10. Once upon a time, a story was told. Now a little question is asked. Answer it kindly and briefly—just a few words will do. No need to explain why.\n\nStory: {context}\n\nHere comes the question:\n\nQuestion: {question}\n\nAnswer:

11. From the folds of a lyrical tale, a question emerges like morning light. Respond with a single phrase, a shard of truth—no more, no less.\n\nStory: {context}\n\nWhisper your reply:\n\nQuestion: {question}\n\nAnswer:

12. Intel received. Narrative extracted. Stand by for situational query. Your task: deliver the answer in minimal terms. Do not elaborate.\n\nStory: {context}\n\nMission Query:\n\nQuestion: {question}\n\nAnswer:

In the prompt templates, {context} is the context, and {question} is the question.

For MATH, we use the prompt template {text1}{question}{text2}, where {text1} and {text2} are two strings that enclose the question. The following 12 pairs of ({text1}, {text2}) are used:

1. ({empty_string}, \nAnswer:\n)
2. (Problem::\n, \nAnswer:\n)
3. (Problem::\n, \nAnswer:\n)
4. (Task:\n\n, \n\nSolution:)
5. (Solve the following math problem:\n\n, \nAnswer:\n)
6. (Solve the following math problem:\n\n, \nAnswer:\n)
7. (**Problem Statement**:\n\n, \n\nSolution:)
8. (Problem::\n, \n\nSolution:)
9. (Solve the following math problem:\n\n, \n\nSolution:)
10. (**Problem Statement**:\n\n, \n\nSolution:)
11. (**Problem Statement**:\n\n, \nAnswer:\n)
12. ({empty_string}, \nAnswer:\n)

Since MATH uses few-shot prompting for evaluation, we further change the examples provided for each prompt template. Hence, while two pairs of ({text1}, {text2}) could be the same, the actual prompt template is different.

For SimpleQA, we use the prompt template {instruction}{question}, where {question} is the original questions in the benchmark, and {instruction} is one of the 12 following strings:

1. {empty_string}
2. Ready your reasoning—consider the challenge that follows.\n\n
3. Take a thoughtful pause, then craft your best response to the prompt beneath this line.\n\n
4. Showcase your insight by addressing the upcoming question.\n\n
5. Put your analytical lens on and dive into the inquiry below.\n\n
6. Channel your inner detective: examine the next question and present your findings.\n\n
7. Let your knowledge shine—respond thoughtfully to the statement that follows.\n\n
8. Engage your critical thinking skills and tackle the question that appears next.\n\n

```
You are an AI assistant that determines whether a model's prediction
matches a given reference answer for a question.
You will be given:
- A question
- A reference (correct) answer
- A model's predicted answer
Your task is to judge whether the prediction matches the reference
answer.
Ignore any explanation in the prediction—only the final selected
answer matters.
Respond with a JSON object in the following format:
{'match': true, 'reason': '...'} if the prediction matches the reference
{'match': false, 'reason': '...'} if it does not match
Be specific and concise in your reasoning.
**Do not answer the question or provide any other information.**
Here is the input:
Question: {question}
Reference Answer: {reference_answer}
Model Prediction: {model_prediction}
```

Figure 3: An example of a judging prompt. After filling in the question, reference answer, and model prediction, we send the prompt to an LLM judge to get the result.

9. Apply the concepts you've mastered to answer the forthcoming inquiry.\n\n

10. Use evidence and reasoning to construct your answer to the question below.\n\n

11. Approach the next problem with curiosity and craft a clear solution.\n\n

12. Demonstrate what you've learned by addressing the prompt that follows.\n\n

B Prompts for LLM-as-Judge

Figure 3 shows the prompt we use for LLM-as-judge. For each benchmark, we make minor task-specific modifications to the judging prompt. For SimpleQA, we use the official prompt from OpenAI.

C Heuristic Evaluation Details

NarrativeQA. Following LongBench (Bai et al., 2024), we compute word-level F1 overlap between normalized predictions and references. Normalization includes lowercasing, removing punctuation and articles (a, an, the), and collapsing whitespace. For example, the prediction "Fifty years" and the reference "50 years" would result in a partial overlap and an F1 score of 0.5.

MATH. Heuristic evaluation typically extracts the final answer from LaTeX-formatted expressions such as `\boxed{...}` or `\fbox{...}`. This method assumes the model explicitly marks its answer with these delimiters. We follow this approach in our experiments. See Appendix A for examples.

SimpleQA. No official heuristic parser is currently available, making rule-based evaluation infeasible. We therefore rely solely on the LLM-as-judge method for this dataset.