RubricEval: A Scalable Human-LLM Evaluation Framework for Open-Ended Tasks

Stanford CS224N Custom Project

Vineel Bhat
Department of Computer Science
Stanford University
vineel@stanford.edu

Abstract

Evaluating open-ended language is particularly challenging due to the unreliability of closed-ended solutions and the high cost of human evaluation. While a suite of LLM-based evaluators have made progress on this problem, they lack nuance and interpretability. To this end, we propose RubricEval, a human-LLM evaluation framework that scores instructions using instruction-level rubrics and provides interpretable summary feedback to model developers. We find that model rankings from RubricEval scores are highly correlated with human preferences from Chatbot Arena ($\rho = 0.98$) using scores from 13 models across 11 categories and 392 instructions. RubricEval rankings remain concordant with rankings produced via human evaluation on the same instruction set, but at the criteria level, we find that RubricEval scores are generally slightly lower than ones from human annotators but have moderate agreement ($\kappa_{QW} = 0.37$). We implemented two mechanisms for feedback generation, and found LLM-generated feedback to be broadly informative and helpful. Overall, RubricEval represents an important step towards developing more accurate and interpretable LLM evaluators for open-ended tasks.

1 Key Information

- Mentor: Yann Dubois

2 Introduction

Evaluations are a critical piece in the process of developing language models (LMs). In particular, they provide a metric to gauge the performance of a LM. Metrics traditionally used for for closed-ended tasks include BLEU [1] and BERTScore [2], which work by comparing outputs to a reference answer or set of reference answers. However, they aren't effective at evaluating LMs because the space of acceptable LM responses is massive. What makes an instruction-following language model response good is nuanced, which makes evaluation challenging.

While human expert evaluation is considered the gold standard, it is not scalable given the monetary and time costs of hiring experts and conducting manual evaluation. Crowdsourced human evaluation (e.g., Chatbot Arena [3]) provides a cheaper silver standard method, but it is still often infeasible.

This has led to a recent push toward developing reference-free automatic evaluators using large language models (LLMs). Prompting LLMs is significantly faster and cheaper than gathering human evaluations, and since they can operate without a reference answer, they don't run into the issue that metrics like BLEU and BERTScore face. Recent studies have shown that LLMs work surprisingly well as evaluators, and correlation with human preferences is high. Notable recent methods for LLM evaluation include the likes of MT-Bench [3], AlpacaEval [4], WildBench [5], and HELM Instruct [6], all of which use either a version of GPT-4 or multiple LLMs as evaluators.

However, all of these evaluators are lacking in some areas. We contend that there are five important aspects an evaluator should hit (three of which we agree with and take from the HELM Instruct paper): open-ended, multidimensional, absolute, varying criteria, and feedback. We expand on each of these below, explain how existing methods satisfy a maximum of three aspects, and explain how our proposed method, RubricEval, satisfies all five (Table 1).

Table 1: Comparison of different evaluation methods.

Eval Method	Open-Ended	Multidimensional	Absolute	Varying Criteria	Feedback
BLEU			✓		
BERTScore			✓		
Chatbot Arena	✓			Implicit	
MT-Bench	✓			Implicit	
AlpacaEval	✓			Implicit	
WildBench	✓			Explicit	
HELM Instruct	✓	✓	✓		
RubricEval	✓	✓	✓	Explicit	✓

Open-Ended: The responses of chat models are open-ended in nature, and a small set of reference answers often can't capture all acceptable responses. This is a key limitation of reference-based evaluators like BLEU and BERTScore.

Multidimensional: Responses can be good and bad in different ways, which isn't captured by "head to head" evaluators like Chatbot Arena and AlpacaEval that simply decide if one response is better than another generally.

Absolute: Evaluators like Chatbot Arena and AlpacaEval use win rates based on pairwise comparisons. This means that we don't know how good a model is in absolute terms. For example, a model may have a low win rate against GPT-4o but still be formidable, and the highest win rate model may not be perfect despite topping the leaderboard.

Varying Criteria: The criteria for what makes a good response is different for each instruction. While HELM Instruct is open-ended, multidimensional, and absolute, it uses the same set of scoring criteria for each instruction, missing nuances at the instruction level. Most pairwise comparison evaluators may implicitly consider varying criteria for each instruction, but these criteria are not explicitly laid out (WildBench is a notable exception).

Feedback: To the best of our knowledge, no current language model evaluation system provides textual feedback on a model's overall strengths and weaknesses with respect to some set of instructions. However, we believe that such feedback would be highly valuable for model developers. Evaluation is a key piece of iterative model development, and textual feedback could provide insight on what exactly needs to be improved rather than solely a score which is hard to interpret.

To satisfy all five aspects, we propose RubricEval, a human-LLM evaluation system which harnesses instruction-specific rubrics. RubricEval uses an LLM to evaluate responses, making it open-ended. RubricEval scores models on a scale from 1 to 4 and uses detailed rubrics to score multiple factors, making it multidimensional and absolute. RubricEval's rubrics are instruction-specific, using a set of criteria generated by humans for each instruction, giving it varying criteria. Finally, RubricEval includes a summarizer that distils instruction-specific information into helpful textual feedback.

Our key contributions can be summarized as the following:

1. Incorporating instruction-specific criteria into an evaluation system that is also multidimensional and absolute, making the evaluator more accurate

2. Implementing a mechanism for such an evaluation system to provide interpretable textual feedback, improving the framework's practical utility

3 Methodology

3.1 Architecture

RubricEval's architecture takes inspiration from assignment grading in a class setting. In a class, assignments and their grading criteria are generally created by the professor, while the teaching assistants are the ones that apply this to grade all the submissions from students. This stems from the fact that the professor is an expert in their field, and so they are best equipped to create the assignments and decide how they should be graded. Since the professor's time is usually highly valued, the time consuming job of doing the actual grading across all submissions is given to teaching assistants, who can apply provided grading criteria with less expertise and whose time is generally considered cheaper. RubricEval employs a similar dynamic to this in order to evaluate language model responses (the assignment submissions). A human expert (the professor) generates instructions and specific criteria about what is needed in a good response for each instruction; then, an LLM evaluator (the teaching assistant), which is cheaper and faster than the human expert, generates a detailed rubric from each instruction's criteria and uses this to evaluate the responses of different models (Figure 1). We use GPT-4o as our LLM evaluator given its state of the art performance.

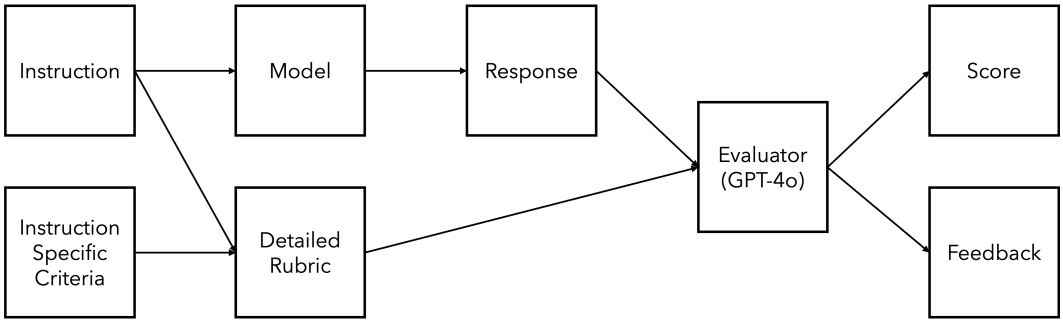


Figure 1: RubricEval flow chart. Expert-generated instructions and instruction specific criteria are used to generate a detailed rubric for each instruction, which is used by the evaluator (GPT-4o) along with the response of the model being evaluated to generate an absolute score and textual feedback.

An example of instruction-specific criteria and a detailed rubric are shown in Supplementary Figure 1 and Supplementary Figure 2, respectively.

3.2 Scoring

For a set of instructions, the LLM evaluator generates a score from 1 to 4 for each of the rubric criteria of each instruction, where 4 represents "excellent", 3 represents "good", 2 represents "fair", and 1 represents "poor." To get instruction-level scores, we uniformly average the criteria scores for each instruction, and to get overall model scores, we uniformly average the instruction scores across all instructions. We use uniform averages due to their simplicity, though we note that there are other approaches, notably having the evaluator generate criteria weights and using a weighted average for instruction-level scores, and/or having the evaluator generate instruction weights and using a weighted average for model scores.

3.3 Summarizer

After scores and feedback have been generated for model completions using instruction-level rubrics, the final step of the RubricEval pipeline is a summarizer, which takes instruction-level feedback and coalesces it into a concise report.

We considered two different approaches for implementing such a summarizer:

1. Hierarchical Unstructured Summarizer: The hierarchical unstructured summarizer uses a summary of category summaries architecture to summarize instruction-level feedback (Supplementary Figure 3). Specifically, an LLM summarizer (GPT-4o) details the strengths and weaknesses of the model in each of n categories that are provided. Then, the LLM

summarizer generates the overall strengths and weaknesses of the model by summarizing the category summaries. While this architecture is two layers in theory, a third layer is sometimes used if the feedback in a category exceeds the summarizing model's context limits (e.g., 128,000 tokens for GPT-4o) or a pre-defined context limit (a lower limit may be desirable as past work has shown that performance degrades for content in the middle of long prompts [7]). In our analysis we use a category total token limit of 32,000 tokens to reduce performance degradation. If instruction-level feedback in a category exceeds 32,000 tokens, we split it up into smaller chunks of instructions and set the category summary to be the summary of the summaries of these chunks.

2. Two-Step Structured Summarizer: The two-step structured summarizer instead summarizes instruction-level feedback by generating a set of criteria that would be important for overall evaluation, then filling in summary feedback blurbs for each of these criteria (Supplementary Figure 4). Again, a third step might be needed if all instruction-level feedback exceeds 32,000 tokens. In this case, we split up the instruction-level feedback into chunks and use the two-step structured summarizer on these chunks, then use an LLM to combine these chunks while still retaining the same number of overall evaluation criteria.

3.4 Dataset and Models

For the purposes of benchmarking, we utilize a set of approximately 1,000 instructions from WildBench which was made publicly available. From this, 392 of the hardest instructions were chosen via a pairwise comparison method that was previously implemented by Yangjun Ruan. Using the WildBench dataset has three primary benefits: 1) it contains a manually curated selection of instructions from real users, 2) each instruction comes with user-defined criteria of what they're looking for, which we can make use of directly in our framework as the instruction-specific criteria, and 3) the instructions are well spread out across 11 categories, which is useful for benchmarking and fits well with the hierarchical summarizer in our framework.

We provide summary statistics for the processed WildBench dataset we use in Supplementary Figure 5 and Supplementary Figure 6, which show instruction counts by category and the total token count of instruction-level feedback by category, respectively. Example instructions from the processed WildBench dataset are shown in Supplementary Figure 7.

We then benchmark the following 13 models using RubricEval: GPT-4 Omni, GPT-4 Turbo, GPT-3.5 Turbo, Gemini 1.5 Pro, Gemini 1.5 Flash, Gemini 1.0 Pro, Gemma 7B, Gemma 2B, Claude 3 Opus, Claude 3 Sonnet, Claude 3 Haiku, Llama 3 70B, and Llama 3 8B. Completions were generated directly via OpenAI, Google, and Anthropic's APIs for their respective proprietary models, while completions for the Gemma and Llama models were generated via Together AI's API.

3.5 Validation

To evaluate the scoring and textual feedback from RubricEval, we use two baselines.

First, we compare model rankings based on RubricEval scores to Chatbot Arena ELO rankings based on human preferences, which is considered a silver standard.

Second, we perform human validation across the entire RubricEval pipeline. To do so, we sampled 54 instructions from the processed WildBench dataset (18 in each of math, reasoning, and creative writing) as well as their associated rubrics. We then manually completed 3 items: 1) annotated the number, list, and cause of clearly questionable rubric criteria and clearly missing criteria for each of the 54 rubrics, 2) scored three models (GPT-4 Turbo, Claude 3 Sonnet, and Gemini 1.0 Pro) across all 54 instructions using each instruction's rubric, and 3) created summary feedback for each of the three models across each of the three categories and overall. This amounted to 1,146 human annotations in total (324 for the rubrics, 810 for scoring, and 12 for the summaries).

4 Results

4.1 Scoring Results

We use RubricEval to score 13 large language models across 11 categories and 392 instructions. Aggregated results at the model-level are shown visually in Figure 2 and in tabular format in Supplementary Figure 8. Notably, the ranking of these models based on RubricEval scores correlates very highly with the ranking of the same models using Chatbot Arena ELO ratings (spearman $\rho = 0.98$). A comparison of these rankings is presented in Supplementary Figure 9. The main discordance is in the ranking of Claude 3 Opus (which is ranked relatively lower by RubricEval compared to Chatbot Arena). RubricEval's correlation of $\rho = 0.98$ with human preferences ties length-corrected AlpacaEval's record 0.98 correlation, while being higher than regular AlpacaEval ($\rho = 0.94$), MT-Bench ($\rho = 0.94$), and MMLU ($\rho = 0.87$). While we caution that RubricEval's correlation of 0.98 is based on just 13 models and that this figure could change when a larger number of models are evaluated, this highlights RubricEval's scoring strength.

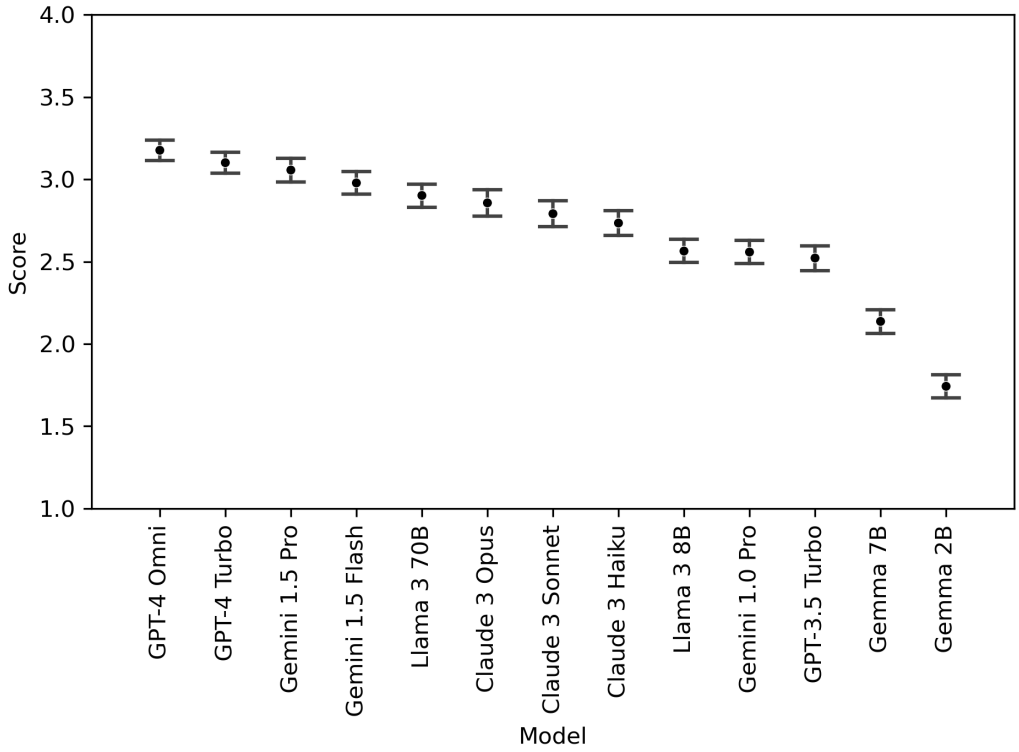


Figure 2: Visual representation of RubricEval scores for 13 models, where 1 represents "poor" and 4 represents "excellent." 95% confidence intervals were calculated via 10,000 bootstrapping iterations.

RubricEval scores for all 13 models stratified by category are presented in Supplementary Figure 10.

4.2 Scoring Validation

While Chatbot Arena rankings are often used as a proxy of human preferences, they weren't generated using the same set of instructions as RubricEval's rankings. For this reason, we perform human validation: for a subset of 54 WildBench instructions in 3 categories (18 in Math, Reasoning, and Creative Writing) and across 3 models (GPT-4 Turbo, Claude 3 Sonnet, and Gemini 1.0 Pro), we compare human scores, that we annotated, with LLM scores using Quadratic Weighted Cohen's Kappa, which measures inter-annotator agreement. Specifically,

$$\kappa_{QW} = 1 - \frac{\sum_{i,j} W_{i,j} \cdot O_{i,j}}{\sum_{i,j} W_{i,j} \cdot E_{i,j}},$$

where $W_{i,j}$ represents a weight matrix (such that annotator score differences are penalized quadratically, meaning a score difference of 2 is penalized four times as much as a score difference of 1), $O_{i,j}$ represents a matrix of the observed frequencies of the two annotators' joint scores, and $E_{i,j}$ represents a matrix of expected frequencies.

Overall, across 810 human score annotations (combining all three models), κ_{QA} between human and LLM criteria scores was 0.37, highlighting that there is moderate agreement between these scores. A visual comparison between human and LLM scores is presented in Figure 3, showing that the LLM evaluator generally gives lower scores compared to a human evaluator.

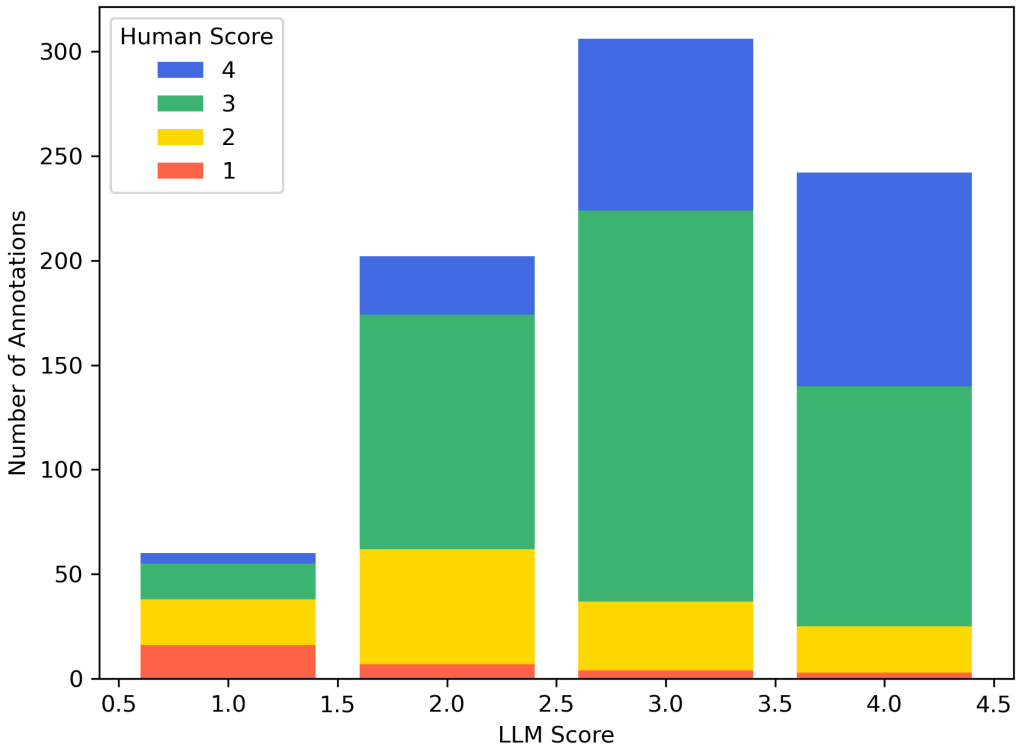


Figure 3: Visual comparison of LLM scores (x-axis) and human scores (y-axis). The distribution highlights that human scores are higher than LLM scores (e.g., more human 3s for LLM 2s).

Interestingly, we find that variance in LLM scores (0.83) is higher than variance in human scores (0.58), contradicting previous studies such as HELM Instruct and AlpacaFarm [8]. This is likely due to having a single human annotator, which is less noisy than having multiple human annotators as was the case in both HELM Instruct and AlpacaFarm.

Given the noise in scores at the criteria level, we were also interested in the overall rankings of the three models across all 54 instructions using human and LLM scores. Table 2 shows that human and LLM rankings for the three models we validated (GPT-4 Turbo, Claude 3 Sonnet, and Gemini 1.0 Pro) are identical. Additionally, while the LLM gave lower scores than the human for Claude 3 Sonnet and Gemini 1.0 Pro, it gave a higher score than the human for GPT-4 Turbo, corroborating the self-bias reported in the AlpacaFarm paper (note that while the LLM evaluator is GPT-4o, it may be considered similar enough given it's an improved multimodal version of GPT-4 Turbo).

Finally, qualitatively, we noticed that the evaluator sometimes misses calculation issues (e.g., with math and brainteasers) and structural issues (e.g., response doesn't include x number of words or isn't formatted as specified), the former of which we saw is a limitation of GPT-4o as an evaluator (it doesn't always calculate correctly itself) and the latter of which is a known limitation of LLMs.

Table 2: Model-level human scores and LLM scores for GPT-4 Turbo, Claude 3 Sonnet, and Gemini 1.0 Pro across a subset of 54 instructions.

Model	Human Score	LLM Score
GPT-4 Turbo	3.24	3.33
Claude 3 Sonnet	3.04	2.84
Gemini 1.0 Pro	2.81	2.53

4.3 Feedback Results

Overall summary feedback for select models using the hierarchical unstructured summarizer is shown in Supplementary Figure 11, overall summary feedback for select models using the two-step structured summarizer is shown in Supplementary Figure 12, and overall human reference summary feedback is shown in Supplementary Figure 13.

4.4 Feedback Validation

To validate summary feedback, we compare LLM-generated feedback with reference human-generated feedback, and generate a concordance score from 1 to 4 using a separate evaluator, where as before, 4 represents "excellent", 3 represents "good", 2 represents "fair", and 1 represents "poor." Overall feedback concordance scores for the hierarchical and two-step summarizers across the three models we used for validation are shown in Table 3 and Table 4, respectively.

Table 3: Concordance scores between overall LLM (hierarchical) and human feedback.

GPT-4 Turbo	Claude 3 Sonnet	Gemini 1.0 Pro
2	2	2

Table 4: Concordance scores between overall LLM (two-step) and human feedback.

GPT-4 Turbo	Claude 3 Sonnet	Gemini 1.0 Pro
2	2	2

LLM-based evaluation highlights that concordance between model and human summaries is fair, though we caution that quantitative evaluation in this context is not a perfect measure given there are a range of acceptable summaries which may not match the reference summary very well. Qualitatively, we find that summaries from both the hierarchical and two-step summarizers are informative and helpful, with the hierarchical summarizer being particularly informative due to its additional nuance and better incorporation of various categories.

4.5 Rubric Validation

Rubric validation was performed by considering human annotations for questionable or missing rubric criteria for each of the 54 rubrics in the validation set. Overall, we found that 1 rubric had a questionable criterion, which was a result of poor human-generated criterion, and that 1 rubric had a missing criterion, which was a result of multiple queries being in the same prompt. The low number of rubric issues during validation highlights that this part of the pipeline is particularly robust.

5 Conclusion

We present RubricEval, an evaluation framework that satisfies all five of the important aspects of automatic evaluators we presented earlier: open-ended, multidimensional, absolute, varying criteria, and feedback. The method has an impressive correlation of 0.98 with Chatbot Arena using scores from 13 models across 392 instructions. At the same time, RubricEval provides helpful feedback on the strengths and weaknesses of the models it evaluates, making it highly interpretable.

5.1 Limitations

Using an LLM evaluator comes with the challenge of bias. Known biases of LLM evaluators include preferences towards longer outputs [9], a model's own output [3], and the presence of lists [8], which we don't account for or attempt to correct. Separately, unlike approaches like AlpacaEval, RubricEval requires human generated criteria for the creation of evaluation rubrics. While costly evaluation is handled by an LLM, one-time instruction and criteria generation by human experts may still be costly for some. The current RubricEval framework allows for evaluation without these human components, instead having an LLM generate them, but we note that this may lead to lower performance. Finally, we caution that the RubricEval LLM evaluator can sometimes incorrectly calculate math or miss structural instructions leading to incorrect evaluation, due to the current state of LLM performance, and that while LLM-generated summary feedback is helpful, fast, and cheap, it is not always as nuanced as human-generated summary feedback.

5.2 Future Work

Future work may seek to use the RubricEval framework to evaluate a larger plethora of models and transform it into an easy to use evaluation package for the NLP community. Separately, future work could improve the framework by incorporating weighted averages (either at the instruction or criteria level) into the scoring process, as detailed in Methods, and by tuning prompts in the pipeline for better results. Finally, as detailed in Limitations, using LLM-generated instructions and criteria could lower costs, but will likely also degrade performance; future work may quantify how much of a difference this makes and whether it's feasible.

6 Ethics Statement

One ethical challenge relevant to this project is the potential for bias in expert-generated instructions and criteria. Since the RubricEval framework relies on instructions and criteria from human experts, there is a risk that these pieces, specifically the criteria, might inadvertently reflect the biases of the experts in aspects such as cultural, gender, and ideology. One potential way to mitigate this would be to make sure expert-generated information has gone through multiple experts with diverse backgrounds and views. Another ethical challenge is evaluation gameability. Due to known biases in LLM-based evaluators, such as length and presence of lists, a malicious actor could game the system so that their model receives a high score on a public leaderboard. One potential way to mitigate this would be to account for biases that are gameable using a simple yet robust generalized linear model approach similar to that used in length corrected AlpacaEval.

7 Contributions and Acknowledgements

RubricEval's LLM rubric generator and LLM evaluator were implemented in previous quarters by Yann Dubois, Josselin Somerville Roberts, and Yangjun Ruan. My primary contributions included fixing bugs in the pipeline in order to run it end-to-end, making prompt modifications to the LLM evaluator to increase performance, implementing two versions of the LLM summarizer, performing inference and benchmarking across a range of models, performing human validation across the pipeline, quantitative/qualitative analysis, and creating plots and text.

We thank Yann Dubois for helpful and insightful conversations, as well as Percy Liang for proposing the two-step structured summarizer architecture.

References

[1] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[2] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.

[3] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

[4] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An Automatic Evaluator of Instruction-following Models.

[5] Bill Yuchen Lin, Khyathi Chandu, Faeze Brahman, Yuntian Deng, Abhilasha Ravichander, Valentina Pyatkin, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild, 2024.

[6] Yian Zhang, Yifan Mai, Josselin Somerville Roberts, Rishi Bommasani, Yann Dubois, and Percy Liang. Helm instruct: A multidimensional instruction following evaluation framework with absolute ratings, February 2024.

[7] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023.

[8] Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2024.

[9] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators, 2024.

A Appendix

Supplementary Figure 1

0	Does the output accurately define what a structural or structural-semantic methodology is in the context of studying Chinese classics?
1	Does the output explain the key concepts of 'light canon', 'heavy canon', and 'matrix texts' clearly and how they relate to the methodology?
2	Is there a clear explanation of how the structural or structural-semantic methodology is applied to the study of Chinese classics?
3	Does the output mention whether this methodology has been applied to the scriptures of other religious traditions such as Christianity, Islam, Buddhism, and Hinduism?
4	If the methodology has been applied to other religious traditions, does the output provide specific examples or details on how it was applied?
5	Does the output maintain a neutral and informative tone without showing bias towards any religious or cultural perspective?
6	Is the information provided in the output accurate and up-to-date with current scholarly research in sinology?
7	Does the output include any references to scholars or works that have contributed to the development or application of this methodology?
8	Is the language used in the output clear and accessible to someone who may not have a background in sinology or religious studies?
9	Does the output encourage further exploration or study of the topic by providing suggestions for additional reading or research?

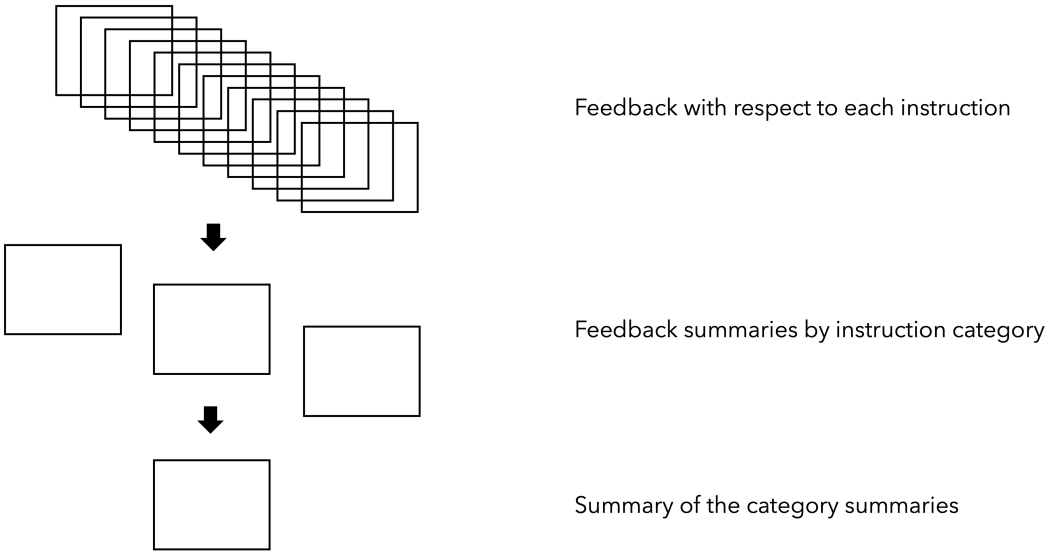
Supplementary Figure 1: Example human-generated criteria for an instruction.

Supplementary Figure 2

	Excellent	Good	Fair	Poor
Understanding of Structural or Structural-Semantic Methodology	The response accurately defines what a structural or structural-semantic methodology is in the context of studying Chinese classics. It clearly explains the key concepts of 'light canon', 'heavy canon', and 'matrix texts', detailing how they relate to the methodology. The explanation includes specific examples and demonstrates a deep understanding of how this methodology is applied to the study of Chinese classics.	The response provides a mostly accurate definition of structural or structural-semantic methodology in the context of studying Chinese classics. It explains the key concepts of 'light canon', 'heavy canon', and 'matrix texts' with some clarity and shows a good understanding of their relation to the methodology. The explanation may lack some specific examples but is generally clear.	The response gives a basic definition of structural or structural-semantic methodology but may have some inaccuracies or lack clarity. The explanation of 'light canon', 'heavy canon', and 'matrix texts' is present but may be incomplete or somewhat unclear. The application to the study of Chinese classics is mentioned but not well-explained.	The response fails to accurately define structural or structural-semantic methodology in the context of studying Chinese classics. It does not clearly explain the key concepts of 'light canon', 'heavy canon', and 'matrix texts', or their relation to the methodology. The explanation is vague or incorrect, and there is little to no mention of the application to the study of Chinese classics.
Application to Other Religious Traditions	The response clearly mentions whether the structural or structural-semantic methodology has been applied to the scriptures of other religious traditions such as Christianity, Islam, Buddhism, and Hinduism. It provides specific examples and details on how the methodology was applied to these traditions, demonstrating a thorough understanding of cross-cultural applications.	The response mentions the application of the structural or structural-semantic methodology to other religious traditions and provides some examples or details. However, the explanation may lack depth or specificity in some areas.	The response briefly mentions the application of the methodology to other religious traditions but provides minimal examples or details. The explanation is somewhat superficial and lacks depth.	The response does not mention the application of the structural or structural-semantic methodology to other religious traditions, or it does so in a very vague and unclear manner. There are no specific examples or details provided.
Scholarly Accuracy and References	The information provided in the response is accurate and up-to-date with current scholarly research in sinology. The response includes references to scholars or works that have contributed to the development or application of the methodology, demonstrating a strong foundation in the academic literature.	The information is mostly accurate and aligns with current scholarly research in sinology. The response includes some references to scholars or works but may not be comprehensive.	The information is somewhat accurate but may contain some errors or outdated information. References to scholars or works are minimal or lacking.	The information is inaccurate or outdated, and there are no references to scholars or works that have contributed to the development or application of the methodology.
Neutrality and Tone	The response maintains a neutral and informative tone throughout, without showing bias towards any religious or cultural perspective. The language is clear and accessible to someone without a background in sinology or religious studies.	The response is mostly neutral and informative, with minor instances of bias. The language is generally clear and accessible.	The response shows some bias or lack of neutrality. The language may be somewhat unclear or difficult for someone without a background in sinology or religious studies to understand.	The response is biased or lacks neutrality, showing favoritism towards a particular religious or cultural perspective. The language is unclear and not accessible to someone without a background in sinology or religious studies.
General Quality	The response is clear, concise, and well-organized. It is helpful and understandable, encouraging further exploration or study of the topic by providing suggestions for additional reading or research.	The response is generally clear and well-organized but may have minor issues with conciseness or clarity. It is helpful and understandable, with some suggestions for further exploration.	The response is somewhat clear but may be disorganized or lack conciseness. It is somewhat helpful and understandable but lacks suggestions for further exploration.	The response is unclear, disorganized, and not concise. It is not helpful or understandable, and there are no suggestions for further exploration.

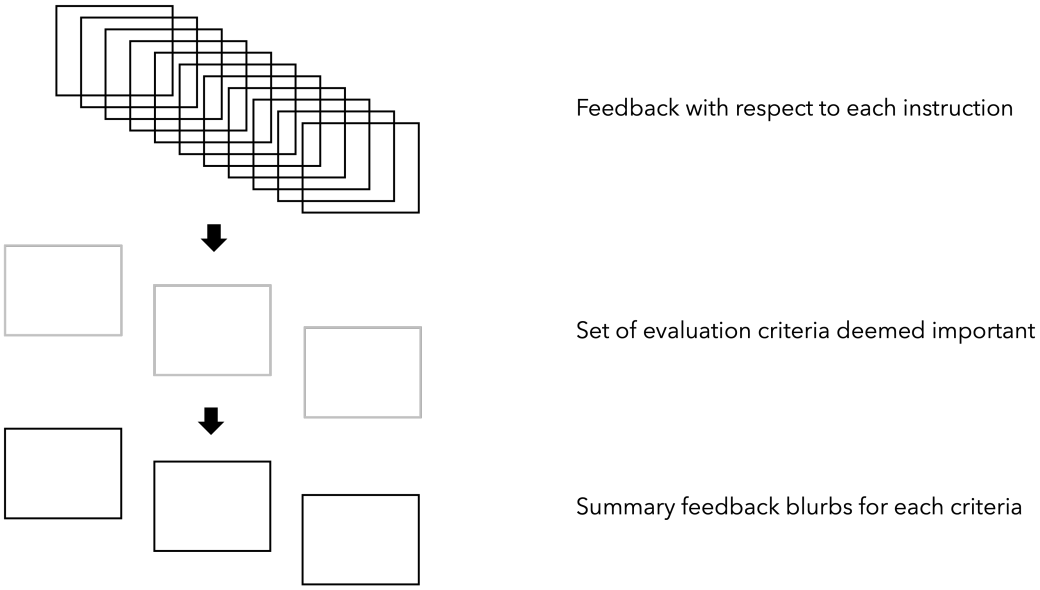
Supplementary Figure 2: Example LLM-generated rubric for an instruction.

Supplementary Figure 3



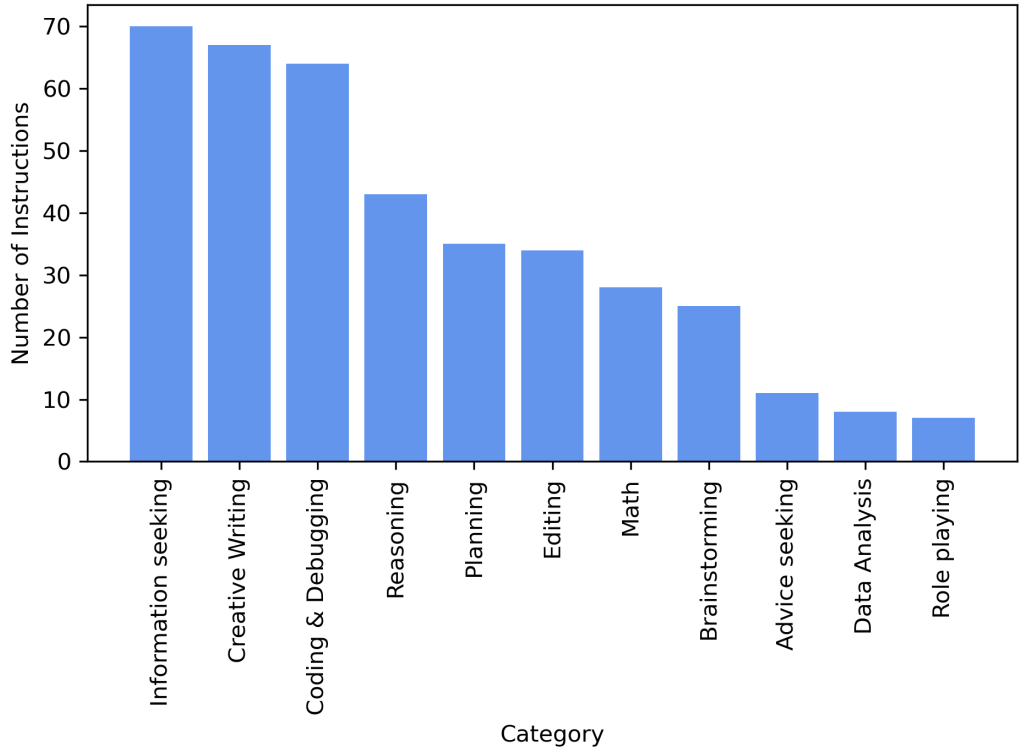
Supplementary Figure 3: Visual representation of the hierarchical unstructured summarizer architecture. Instruction-level feedback is first summarized by category, and then the summaries of each category of instructions are summarized into a single overall summary.

Supplementary Figure 4



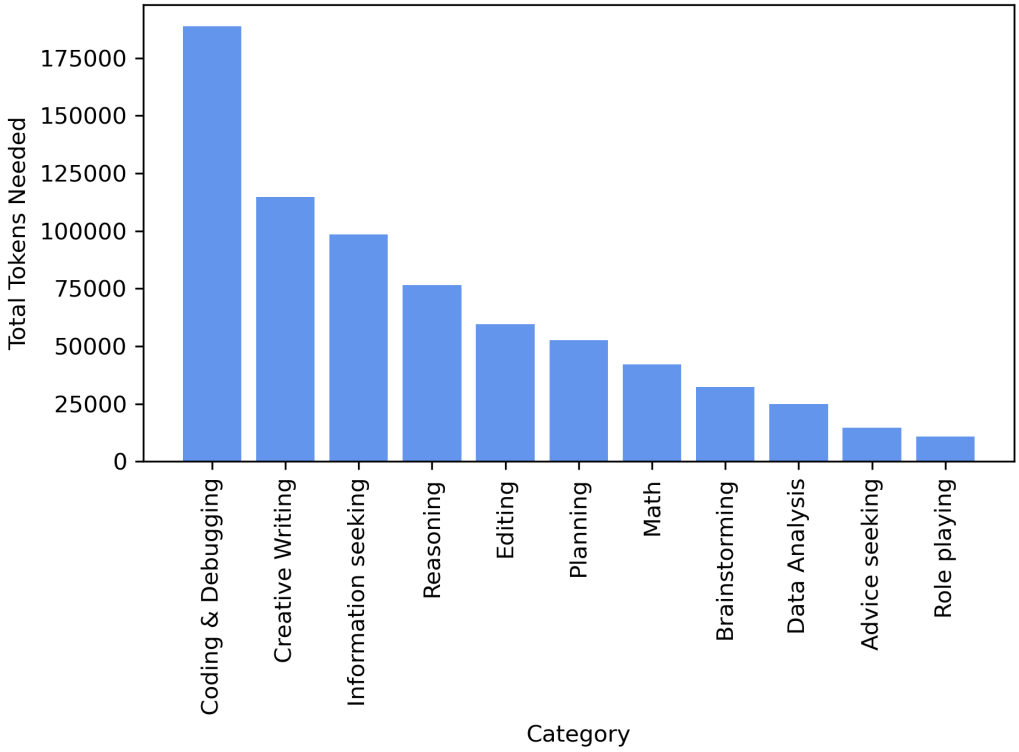
Supplementary Figure 4: Visual representation of the two-step unstructured summarizer architecture. A fixed number of evaluation criteria are determined from instruction-level feedback, then a summary blurb is generated for each criteria.

Supplementary Figure 5



Supplementary Figure 5: Number of instructions by category in the processed WildBench dataset.

Supplementary Figure 6



Supplementary Figure 6: Total token counts of feedback by category. Across the 11 categories in the WildBench dataset, we map how many tokens would be needed to incorporate the feedback information that would be fed into the summarizer. This information includes the prompts, rubrics, scores, and evaluator feedback for each instruction related to the category.

Supplementary Figure 7

Category: Information seeking

Prompt: Discuss the U.S. federal court system, and which rulings bind other rulings. Also discuss whether U.S. citizens in different areas effectively have different case law that applies to them due to geographical location differences.

Category: Creative Writing

Prompt: Compose an enigmatic sacred text matching the imaginative style of Michael Kirkbride and inspired by the Upanishads but resembling the format of the 36 Lessons of Vivec and following a similar narrative arc. This scripture introduces a scientifically accurate mythology which forms the basis of a modern-day religion. The text should be abundant in symbolism and allusions, each sentence conveying profound and layered meanings that render the text open to limitless interpretation. The faith emerging from this scripture bridges the realms of rationality and the human longing for the mystical. It is founded upon principles of radical skepticism, philosophical pessimism, and existential nihilism. The tone should be unsettling while consciously avoiding overused nihilistic images such as “void” and “darkness." The text should also avoid commonly-used imagery and metaphors such as tapestry, song, cosmic sea, etc. Implicitly or explicitly, the religion derived from the text proposes “lucid intoxication” as humanity’s path forward—an embrace of meaninglessness through the pursuit of artistic expression and self-abnegation.

Category: Math

Prompt: Twins Anna and Tanya, who are both 1.75m tall, both look at the top of a tower, Anna look at the tower with a 40 degree and tanya look at the tower with a 50 degree, if they are standing 7m apart how tall is the tower

Category: Planning

Prompt: I'd like you to help me come up with a content schedule for my blog that has the best chance of helping me rank for long tail keywords that are specific to my keyword. I'll tell you my main target keyword in the rank math field. Please target transaction style search terms only. Please come up with clickbait style titles for these blog posts. Please organize each blog post title in a nice looking table so that it looks like a calendar. Each week should have its own table. Each day should have five pieces of content a day with with five unique and separate titles listed for each day. Include all 7 calender days of content Monday through Sunday. Include 4 weeks of content. Above the table say "Head In The Clouds SEO "E-E-A-T" Strategy 30 day Authority blog Post Schedule FOR KEYWORD" and replace "KEYWORD" with the keyword provided in the prompt in all caps. Then, under the table say "If you liked this prompt please like it on the prompt search page so we know to keep enhancing it. All content is to be output in English

Category: Reasoning

Prompt: How can the idea of getting a hair cut fit into major philosophical ideas, including the idea of 'being vs. becoming'?

Supplementary Figure 7: Example instructions from the processed WildBench dataset.

Supplementary Figure 8

Rank	Model	Score
1	GPT-4 Omni	3.18
2	GPT-4 Turbo	3.10
2	Gemini 1.5 Pro	3.06
2	Gemini 1.5 Flash	2.98
2	Llama 3 70B	2.90
3	Claude 3 Opus	2.86
4	Claude 3 Sonnet	2.79
6	Claude 3 Haiku	2.73
2	Llama 3 8B	2.56
5	Gemini 1.0 Pro	2.56
7	GPT-3.5 Turbo	2.52
7	Gemma 7B	2.14
7	Gemma 2B	1.74

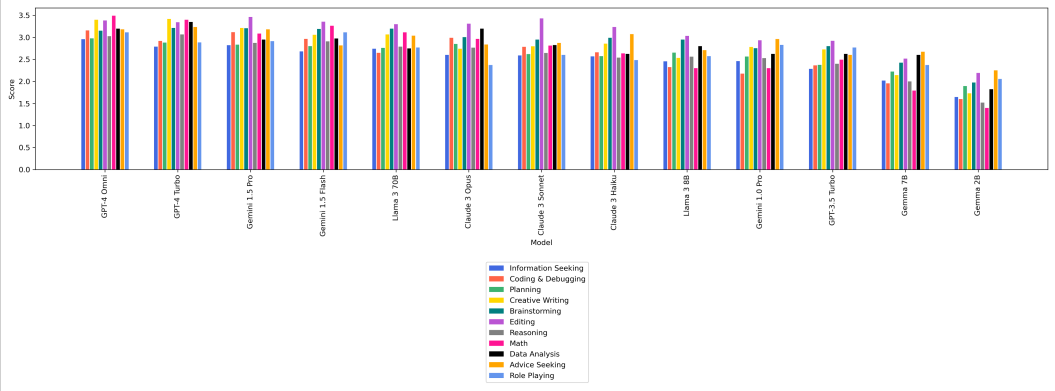
Supplementary Figure 8: RubricEval scores for 13 models. We use RubricEval to score 13 popular models on a scale from 1 to 4, where 1 represents "poor" while 4 represents "excellent." Notably, the ranking of these models using RubricEval scores correlates well with Chatbot Arena ($\rho = 0.98$).

Supplementary Figure 9

Rank	RubricEval	Chatbot Arena
1	GPT-4 Omni	GPT-4 Omni
2	GPT-4 Turbo	Gemini 1.5 Pro
2	Gemini 1.5 Pro	GPT-4 Turbo
2	Gemini 1.5 Flash	Claude 3 Opus
2	Llama 3 70B	Gemini 1.5 Flash
3	Claude 3 Opus	Llama 3 70B
4	Claude 3 Sonnet	Claude 3 Sonnet
6	Claude 3 Haiku	Claude 3 Haiku
2	Llama 3 8B	Llama 3 8B
5	Gemini 1.0 Pro	Gemini 1.0 Pro
7	GPT-3.5 Turbo	GPT-3.5 Turbo
7	Gemma 7B	Gemma 7B
7	Gemma 2B	Gemma 2B

Supplementary Figure 9: Comparison of RubricEval rankings (based on overall model-level scores) and Chatbot Arena ELO rankings for 13 models.

Supplemental Figure 10



Supplementary Figure 10: RubricEval scores by model and category. Each model was evaluated across instructions from 11 categories: information seeking, coding & debugging, planning, creative writing, brainstorming, editing, reasoning, math, data analysis, advice seeking, and role playing. This plot breaks down the overall model scores into their components (note that overall scores are simple uniform averages of these category scores).

Supplemental Figure 11

Overall – LLM Summary (GPT-4 Omni)

Strengths:

- Clarity and Conciseness: Provides clear, concise, and well-organized responses across various categories.
- General Quality: Consistently produces professional, grammatically correct, and logically structured content.
- Engagement: Maintains a humanlike writing style and engaging tone, enhancing user interaction.
- Actionable and Practical: Offers specific, actionable steps and practical advice in multiple domains.
- Comprehensive Coverage: Efficiently processes information, identifies patterns, and supports various techniques and frameworks.
- Adaptability: Adapts well to different roles and scenarios, maintaining consistent character traits and behaviors.

Weaknesses:

- Depth and Insight: Often lacks detailed analysis, depth, and practical implications in responses.
- Examples and Context: Frequently omits specific examples, case studies, and detailed contextual information.
- Error Handling and Robustness: Lacks comprehensive error handling and robustness against invalid inputs.
- Customization and Flexibility: Provides limited tips for customization and flexibility in solutions.
- Complexity and Nuance: Struggles with highly complex, nuanced roles, and specialized or technical areas.
- Innovation and Creativity: Could benefit from more innovative strategies, unique elements, and creative integration.

Overall – LLM Summary (Claude 3 Haiku)

Strengths:

- Clarity and Understandability: Consistently clear, concise, and easy to understand across various tasks.
- General Quality: Well-structured, logically consistent, and free from grammatical errors.
- Engagement and Creativity: Engaging style, creative storytelling, and effective character roles.
- Professional Tone: Maintains a professional and respectful tone.
- Functionality and Practicality: Provides actionable suggestions, accurate calculations, and clear visualizations.
- Comprehensive Coverage: Covers a wide range of topics effectively, including coding, planning, and data analysis.

Weaknesses:

- Depth and Detail: Often lacks depth in explanations, character development, and specific examples.
- Contextual Understanding: Sometimes misses broader context, historical details, and specific implementation guidance.
- Accuracy and Completeness: Occasional errors in calculations, incomplete responses, and missing crucial information.
- Innovation and Variety: Needs more innovative ideas, diverse post types, and interactive elements.
- Security and Testing: Potential security vulnerabilities and lack of testing/validation suggestions.
- Handling Complexity: Struggles with highly complex, noisy datasets, and nuanced roles.

Overall – LLM Summary (Llama 3 8B)

Strengths:

- Clarity and Accessibility: Provides clear, concise, and grammatically correct information.
- Logical Structure: Information is well-organized and logically structured.
- Engaging Writing Style: Maintains an engaging and natural writing style.
- General Quality: Produces clear, coherent, and well-documented responses.
- Practical and Actionable Advice: Offers practical suggestions and actionable steps.
- Adaptability: Adapts well to various roles and scenarios, maintaining consistent character traits.
- Pattern Recognition: Accurately identifies patterns and trends in data analysis.
- Error Handling: Generally good at handling errors with clear messages.

Weaknesses:

- Depth and Specificity: Often lacks depth, detail, and specificity in responses.
- Accuracy and Relevance: Sometimes provides irrelevant or inaccurate information.
- Complexity Handling: Struggles with highly complex, nuanced, or technical topics.
- Mathematical Accuracy: Frequently makes significant errors in calculations and transformations.
- Customization and Flexibility: Limited guidance on customization and flexibility for different scenarios.
- Engagement and New Insights: Lacks depth in providing new insights or engaging content.
- Dependence on Input Quality: Performance is highly dependent on the quality of input data.

Supplementary Figure 11: Select overall LLM summaries using the hierarchical unstructured summarizer for the following three models: GPT-4 Omni, Claude 3 Haiku, and Llame 3 8B.

Supplemental Figure 12

1. "clarity and coherence": The model's responses are generally clear and logically structured, making them easy to follow. However, there are occasional lapses in depth and detail, which can affect the overall coherence of the narrative.

2. "depth of analysis": The model often provides a basic level of analysis but lacks depth in several areas. For example, it may mention key concepts or issues but fails to explore them thoroughly or provide detailed examples.

3. "historical accuracy": The model's responses are mostly accurate but sometimes lack specific details about key figures, events, and institutions. This can lead to a superficial understanding of the historical context.

4. "relevance of examples": The model includes relevant examples to support its points, but these examples are often not detailed enough to fully illustrate the concepts being discussed.

5. "logical structure": The model's responses are generally well-organized and logically structured, with a clear progression of ideas. However, there are occasional inconsistencies that can disrupt the flow of the narrative.

6. "engagement and readability": The model's writing is engaging and easy to read, with a good balance of technical language and accessible explanations. However, it could benefit from more varied sentence structures and a more dynamic writing style.

7. "use of supporting evidence": The model uses supporting evidence to back up its claims, but this evidence is often not detailed enough to be fully convincing. More specific examples and references to credible sources would strengthen the arguments.

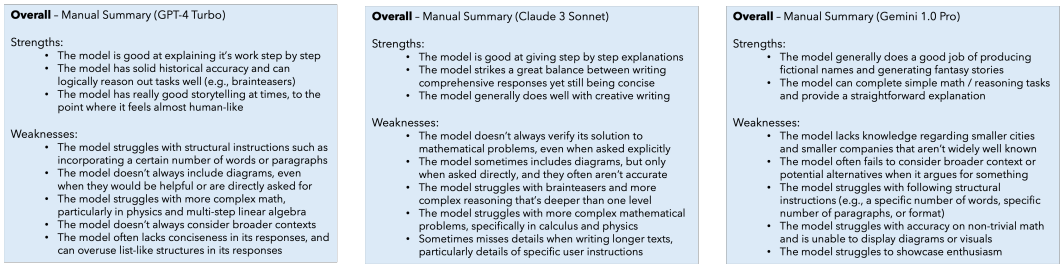
8. "integration of multiple perspectives": The model attempts to consider multiple perspectives but often lacks depth in this area. It would benefit from a more thorough exploration of different viewpoints and their implications.

9. "consistency in argumentation": The model's arguments are generally consistent, but there are occasional lapses where the logic is not fully developed or supported. More rigorous reasoning and evidence would improve the overall consistency.

10. "consideration of counterarguments": The model rarely addresses counterarguments, which can weaken its overall persuasiveness. Including and refuting counterarguments would make the responses more robust and comprehensive.

Supplementary Figure 12: Overall LLM summary using the two-step summarizer for GPT-3.5 Turbo.

Supplemental Figure 13



Overall – Manual Summary (GPT-4 Turbo)

Strengths:
- The model is good at explaining it's work step by step
- The model has solid historical accuracy and can logically reason out tasks well (e.g., brainteasers)
- The model has really good storytelling at times, to the point where it feels almost human-like

Weaknesses:
- The model struggles with structural instructions such as incorporating a certain number of words or paragraphs
- The model doesn't always include diagrams, even when they would be helpful or are directly asked for
- The model struggles with more complex math, particularly in physics and multi-step linear algebra
- The model doesn't always consider broader contexts
- The model often lacks conciseness in its responses, and can overuse list-like structures in its responses

Overall – Manual Summary (Claude 3 Sonnet)

Strengths:
- The model is good at giving step by step explanations
- The model strikes a great balance between writing comprehensive responses yet still being concise
- The model generally does well with creative writing

Weaknesses:
- The model doesn't always verify its solution to mathematical problems, even when asked explicitly
- The model sometimes includes diagrams, but only when asked directly, and they often aren't accurate
- The model struggles with brainteasers and more complex reasoning that's deeper than one level
- The model struggles with more complex mathematical problems, specifically in calculus and physics
- Sometimes misses details when writing longer texts, particularly details of specific user instructions

Overall – Manual Summary (Gemini 1.0 Pro)

Strengths:
- The model generally does a good job of producing fictional names and generating fantasy stories
- The model can complete simple math / reasoning tasks and provide a straightforward explanation

Weaknesses:
- The model lacks knowledge regarding smaller cities and smaller companies that aren't widely well known
- The model often fails to consider broader context or potential alternatives when it argues for something
- The model struggles with following structural instructions (e.g., a specific number of words, specific number of paragraphs, or format)
- The model struggles with accuracy on non-trivial math and is unable to display diagrams or visuals
- The model struggles to showcase enthusiasm

Supplementary Figure 13: Human-generated overall summary feedback for three validation models.