SedarEval: Automated Evaluation using Self-Adaptive Rubrics

Anonymous ACL submission

Abstract

The evaluation paradigm of LLM-as-judge gains popularity due to its significant reduction in human labor and time costs. This approach utilizes one or more large language models (LLMs) to assess the quality of outputs from other LLMs. However, existing methods rely on generic scoring rubrics that fail to consider the specificities of each question and its problem-solving process, compromising precision and stability in assessments. Inspired by human examination scoring processes, we propose a new evaluation paradigm based on self-adaptive rubrics. Specifically, we create detailed scoring rubrics for each question, capturing the primary and secondary criteria in a structured format of scoring and deduction points that mimic a human evaluator's analytical process. Building on this paradigm, we further develop a novel benchmark called SedarEval, which covers a range of domains including long-tail knowledge, mathematics, coding, and logical reasoning. SedarEval consists of 1,000 meticulously crafted questions, each with its own self-adaptive rubric. To further streamline the evaluation, we train a specialized evaluator language model (evaluator LM) to supplant human graders. Using the same training data, our evaluator LM achieves a higher concordance rate with human grading results than other paradigms, including GPT-4, highlighting the superiority and efficiency of our approach.

1 Introduction

The rapid advancements in large language models (LLMs) have led to their widespread use (OpenAI et al., 2024; Team et al., 2023; Anthropic, 2024; Bai et al., 2023). However, assessing these models in open-ended question-answering scenarios poses a significant challenge. Automated metric-based evaluations offer speed and convenience but often fall short due to the diversity of ground truth (Schluter, 2017a; Reiter, 2018; Montahaei et al., 2019; Freitag et al., 2020). In contrast, human-based evaluations provide reliable assessments but require substantial resources.

To bridge the gap, the LLM-as-a-judge paradigm attempts to strike a balance between automated and human evaluation. Prominent examples of this approach include MT-bench (Zheng et al., 2024) and Arena (Chiang et al., 2024), which leverage proprietary models to evaluate individual or comparative model responses. These benchmarks use pre-defined principles, such as the 3H principle (human-like, helpful, harmonious), to determine responses that align best with realistic human preferences. The widespread use of GPT-4 (OpenAI et al., 2024) as an evaluator in these studies presents challenges, including high costs for research institutions and potential data leaks.

Some studies (Zhu et al., 2023; Li et al., 2023a; Wang et al., 2024; Kim et al., 2024a,b) propose using open-source pretrained models (Touvron et al., 2023; Bai et al., 2023; Zeng et al., 2022) to train specialized evaluator LMs, offering a more cost-effective and secure solution. However, these methods typically use a uniform, question-agnostic rubric to guide the scoring process, overlooking the unique characteristics of each question. Each question has different emphases, with primary and secondary scoring points. A general rubric applies uniform criteria, failing to accurately reflect human preferences.

To adaptively align the scoring process with human judgment, we propose a novel evaluation paradigm based on self-adaptive rubrics. Unlike coarse-grained general rubrics, we provide fine-grained rubrics for each task, detailing specific scoring and penalty points with primary and secondary information. By analyzing focus points, we assign different values to each point. Additionally, we introduce penalty points to penalize models for generating rejected responses. The scoring process considers both preferred and rejected perspectives.

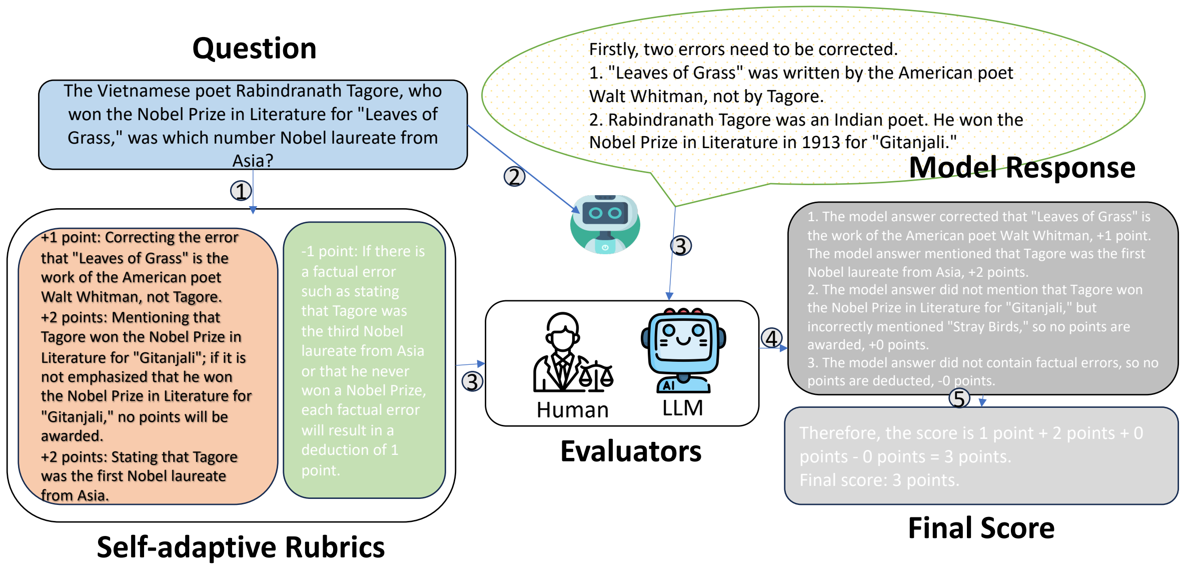

Figure 1: Automated evaluation pipeline using self-adaptive rubrics.This pipeline dynamically adjusts the evaluation rubric based on the input question, resulting in a scoring process that aligns more closely with human evaluators.

The inconsistent coverage of positive and penalty points ensures a more refined constraint on the scoring process. These detailed scoring trajectories simplify the evaluation process to an instruction-following task, reducing dependency on a judge model's internal knowledge and skills, leading to more accurate and stable assessments. Building on this paradigm, we construct a new benchmark called SedarEval that fully aligns with realistic scenarios.

We further conduct ablation experiments on each component of the LLM-as-a-judge paradigm to investigate training a specialized LLM for scoring, revealing their respective importance. We analyze whether LLMs can correctly evaluate questions they can correctly answer and find that insufficient diversity in existing SFT data and a lack of evaluation-format data limit model performance. We also propose human-AI consistency to ensure evaluator LLMs maintain alignment with human preferences while leveraging their chain of thought capability to improve evaluation performance. Based on these findings, we develop a specialized evaluator LLM tailored to the benchmark for automated scoring. This model surpasses GPT-4 in model-level and question-level Pearson correlation, GSB, and ACC metrics, demonstrating higher consistency with human judgment. Experimental results validate the effectiveness and efficiency of our proposed paradigm.

Our contributions are summarized as follows:

1. We propose a novel evaluation paradigm using self-adaptive rubrics for each question, offering granular guidance and closely aligning the scoring process with human evaluation.

2. We develop a high-quality benchmark called SedarEval, featuring 1,000 meticulously crafted questions with detailed rubrics, and conduct manual evaluations on 20 LLMs.

3. We analyze the training of evaluator LMs, highlight existing methods' shortcomings, and use the self-adaptive rubrics paradigm to train an evaluator LM that surpasses GPT-4 in agreement with human evaluations.

2 Related Work

Benchmark LLMs Capabilities. With the rapid advancement of LLMs (OpenAI et al., 2024; Team et al., 2023; Anthropic, 2024), it has become a substantial challenge to benchmark their broad capabilities reliably. NLU-style tasks (Hendrycks et al., 2020; Huang et al., 2024; Srivastava et al., 2022; Zhong et al., 2023), such as multi-choice QA, employ general-exam questions from various domains to assess a model's knowledge and comprehension abilities. However, their real-world usage is limited due to misalignment with human preferences. Recently, reference-free benchmarks (Li et al., 2023b; Chiang et al., 2023; Zheng et al., 2024; Ye et al., 2023) have been proposed to evaluate texts' quality in a generative setting directly. Unlike previous datasets, our benchmark provides a comprehensive and stable model assessment with its diverse test cases and broad label distribution.

Automatic NLG Evaluation. It's notably challenging to evaluate the quality of generated text in the field of natural language generation (NLG). Traditional n-gram-based metrics (Papineni et al., 2002; Lin, 2004; Snover et al., 2006) and embedding-based metrics (Li et al., 2019; Zhang et al., 2020; Risch et al., 2021) can only assess lexical or semantic similarity between the generated answers and reference answers (Schluter, 2017a; Reiter, 2018; Montahaei et al., 2019; Freitag et al., 2020). These metrics have been found to have a relatively low correlation with human preferences (Liu et al., 2023a). Recently, employing LLM as a judge (Zheng et al., 2023; Li et al., 2023b; Chan et al., 2023) is a novel evaluation paradigm that has gained widespread application. The most common approach involves using proprietary LLMs, such as GPT-4 (OpenAI et al., 2024), as judge models to rank or score outputs generated by other models. However, this method relies on closed-source models, incurs high costs, and poses risks of internal evaluation dataset leaks for companies developing LLMs. To address these issues, various works (Zhu et al., 2023; Li et al., 2023a; Wang et al., 2024; Kim et al., 2024a,b) have proposed training dedicated scoring models on open-source base models using synthetic or manually labeled data. These evaluations often use reference answers to assist in the assessment or employ general rubrics to guide the scoring process. However, these approaches overlook the differences between individual questions and the varying scoring criteria of each question, even within the same category. In contrast, we propose an evaluation paradigm based on self-adaptive rubrics that generates fine-grained, customizable rubrics for each question, guiding a more precise scoring process. It is worth noting that although Prometheus 2 also claims to use fine-grained rubrics, their rubrics remain question-agnostic.

Quantifying Evaluation Confidence. The automatic metrics are imperfect, and we must measure their performance further. A gold standard for this is their alignment with human judgment and the confidence level we can have when these metrics guide our decision-making process. However, quantifying this performance (Krishna et al., 2021; Schluter, 2017b; Stureborg et al., 2024) is difficult due to various factors (the evaluator's accuracy and stability, evaluation set size, the extent of the performance difference among competing models, etc.). (Kocmi et al., 2021; Deutsch et al., 2021; Zhang and Vogel, 2004) investigate the correlation between human judgment and traditional automatic metrics such as ROUGE and BLEU and analyze their confidence intervals. For LLM-based evaluators, commonly used metrics include **Pearson**, **Spearman**, and **Kdendall-Tau** to measure the alignment between the model's scores and human preferences. However, previous work has primarily focused on the correlation of rankings or overall scores at the model level without comparing the scores with human ratings at the individual question. This limits the interpretability of the scoring process and hampers its utility in guiding the development and iteration of LLMs.

3 SedarEval Benchmark

In this section, we introduce SedarEval, a benchmark constructed upon the self-adaptive rubrics paradigm. We begin by delving into the intricacies of the self-adaptive rubric paradigm, followed by a detailed explanation of the benchmark's core components – questions and their corresponding rubrics – along with the methodology for model evaluation using this benchmark. To ensure the quality of SedarEval, we incorporate comprehensive human assessment into the construction process, meticulously filtering out samples that fail to meet the established quality standards.

3.1 Self-Adaptive Rubrics

Previous LLM-as-a-judge approaches, which rely on general rubrics or principles for scoring, often lack specific, problem-related rubric guidance. Consequently, these methods depend heavily on the inherent capabilities of the LLM itself, leading to potential errors in evaluations due to insufficient reasoning abilities or hallucinations. Additionally, this approach introduces extraneous biases, such as position bias and order bias.

Self-adaptive rubrics address these issues by tailoring the evaluation criteria to the specific problems at hand, incorporating the focal points of the problem and assigning different weights accordingly. By introducing penalty points, these rubrics align more closely with human judgments by deducting points for outputs that deviate from expected tendencies. To prevent human evaluators (or LLMs) from making incorrect assessments due to a lack of background information, additional context is provided for each question to assist in the scoring process. A typical self-adaptive rubric comprises three components: scoring points, penalty points, and background knowledge, as illustrated in Table 3.

3.2 Dataset Construction

Questions: We have defined a classification system for objective questions, with a two-tiered scoring system as shown in the diagram. Under each secondary classification, we have hired five people to create questions. Specifically, each person is required to first create their own questions to get a question pool, and then each person votes on all the questions. We only keep the questions that all five people agree on.

For each candidate question, the annotators will select 5 LLMs to test the effectiveness of the questioned question. We only keep the questions with a larger variance in scores, which are more discriminating, and remove the questions where the answers from different models are almost the same, which are not helpful in distinguishing between different models. For example, if a question can be answered correctly by all models, or incorrectly by all models, then this question cannot show which model is better.

After collecting the initial questions, we hired another group of people to compare all the questions in pairs to judge the similarity of the problem-solving ideas for the two questions and delete the questions with too much similarity.

Rubrics: For each question, we assign it to three individuals to discuss together and generate a rubric like the one shown in Figure 1.

For more detailed information, please refer to Appendix D, which contains benchmark statistics and the leaderboard.

3.3 Evaluation Pipeline

The entire evaluation pipeline using our benchmark is illustrated in Figure 1. Given a question, its corresponding rubrics, and the model to be evaluated, we first input the question into the model to generate a response. The response is then scored according to the predefined rubric, either by human evaluators or using LLMs. Finally, all the scores are aggregated to obtain the model's total score.

4 Evaluator Language Model

In this section, we introduce an evaluator LM aligned with the self-adaptive rubrics paradigm to substitute human evaluators. We begin by delineating the evaluation format. Subsequently, we propose a novel data filtering strategy to align the Chain-of-Thought evaluation process with human judgments. Finally, we discuss the automation of rubric generation.

4.1 Evaluation Format

The evaluation format consists of two types: direct scoring of individual model outputs and pairwise comparison of model outputs to determine the superior one. Pairwise evaluation requires significantly more comparisons as the number of candidate models increases, as shown by Equation 1. Therefore, we employ direct assessment in this paper. Notably, direct assessment scores can be compared to derive pairwise results.

$$C(n, 2) = \frac{n!}{2!(n-2)!} - n = \frac{n^2 - 3n}{2} \quad (1)$$

We use a reference-based format to organize the output. Specifically, for each question, we compile the reference answer, self-adaptive rubrics, and scoring examples to create an auto-prompt template. When evaluating answers, we incorporate the answers into this auto-prompt template as the complete input. We conduct ablation experiments on each component in zero-shot, few-shot, and instruction tuning settings.

4.2 Human-AI Consistency

Human annotators provide specific scores for each response without corresponding explanations, which is efficient but suboptimal for training evaluator LMs. To alleviate this issue, we use GPT-4 to generate detailed reasoning steps using Chain-of-Thought. However, scoring preferences may differ between GPT-4 and human annotators, and both may make errors. To mitigate these errors and align the scoring process with human judgment, we introduce a **Human-AI Consistency** strategy to improve synthetic data quality. We extract final scores from the GPT-4 scoring process and compare them with human scores, retaining only the data where GPT-4 and human results are consistent, as shown in Equation 2, where $\mathcal{H}$ represents human scores, $\mathcal{A}$ represents AI scores, and $\mathbb{I}$ is an indicator function.

$$\mathcal{T} = \{(h, a) \mid h \in \mathcal{H}, a \in \mathcal{A}, \mathbb{I}(h, a) = 1\} \quad (2)$$

This approach only retains instances where human and AI scores are consistent and differs from rejection sampling, which uses human scores as a reward function to select the optimal output from multiple GPT-4 results.

4.3 Automatic Rubric Generation

To reduce human annotation costs, we investigate using human-annotated datasets to train a model for automatic self-adaptive rubric generation. By providing the model with questions and corresponding reference answers, we train it to produce rubrics that delineate scoring criteria and identify deduction points.

Generating self-adaptive rubric format output is straightforward, but aligning rubrics with human preferences requires aligning the model with human evaluative criteria. This complexity arises because identifying scoring points, assigning specific weights, and criteria for deductions are significantly influenced by human judgment.

The training process for the automatic rubric generation model comprises two stages. Initially, we use human-labeled data to train a base model through Supervised Fine-Tuning (SFT), as depicted in Equation 3.

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p_\theta(y_i|x_i) \quad (3)$$

The base model generates rubrics that conform to the specified format, though they may not fully align with human preferences (quantitative metrics will be introduced in Section 5.2). In the next phase, rubrics generated by the base model are treated as rejected responses, while human-labeled rubrics serve as preferred responses to construct preference pairs. We then train the model using Direct Preference Optimization (DPO) to align it with human preferences, as shown in Equations 4 and 5.

$$f(y, x) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} \quad (4)$$

$$\mathcal{L}(\pi_\theta) = -\log \sigma\left(f(y_w, x) - f(y_l, x)\right) \quad (5)$$

We also explore automating rubric generation using GPT-4 without reference answers. To ensure accuracy, GPT-4 creates both the rubric and an ideal answer for each question. If the ideal answer corresponds with the ground truth, the generated rubric is deemed acceptable. We employ a self-refinement strategy to help the model iteratively refine its outputs, aligning it with human preferences. For detailed algorithmic procedures and prompts, refer to Appendix E.1.

5 Experiment

5.1 Experimental Setting

We train the Evaluator Language Model and the Rubric Generation Model using both the open-source model LLaMA-3 (Touvron et al., 2023) and our internal model XDG[1]. To maximize training efficiency and utilize hardware resources, we implement tensor parallelism (Shoeybi et al., 2020) with PyTorch 2.3 (Paszke et al., 2019). For the 7B/8B models, we use 128 H100 GPUs, while for the 70B models, we use 512 H100 GPUs. For the models' chat versions (i.e., instruction-tuned), we employ the same chat markup language (ChatML) as the models themselves. For the pre-trained versions, we use a unified ChatML to reduce data bias. We adopt adaptive learning rate and batch size strategies. Further training details are provided in Appendix A.

5.2 Evaluation Metrics

To assess the performance of the evaluator language model, we use Pearson's correlation coefficient and Spearman's rank correlation coefficient. These statistical measures assess the consistency between the outcomes of the evaluator language model and those obtained from human evaluators.

Each question is accompanied by a detailed rubric specifying exact scoring and deduction criteria, so we use accuracy to evaluate the model's capability in following these self-adaptive rubrics for scoring. Considering potential noise in the model scoring, we introduce a weaker threshold ACC, which considers a result correct if it falls within a specified range. The calculation formulas are presented in Equation 6.

$$\text{ACC}_t = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} 1, & \text{if } |y_{\text{pred}_i} - y_{\text{true}_i}| \leq \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

To facilitate the iterative enhancement of LLMs using our benchmark, a robust metric is essential

[1]The name of this model has been anonymized to ensure confidentiality.

to assess whether the current model version outperforms its predecessor. Therefore, we adopt the widely used GSB (Good, Same, Bad) metric to compare model performance. Given two models, A and B, the calculation formula is presented in Equation 7. In this context, "#good" signifies that model A surpasses model B, "#bad" denotes the contrary, and "#same" indicates equivalent performance between the models.

$$\Delta GSB = \frac{\#\text{good} - \#\text{bad}}{\#\text{good} + \#\text{same} + \#\text{bad}} \quad (7)$$

To evaluate the quality of automatically generated rubrics, we draw on the ACU (Liu et al., 2023b) and FactScore (Min et al., 2023) paradigms, using GPT-4 to calculate the match between the generated rubrics and the ground truth rubrics. The calculation formula is specified in Equation 8, where GT represents the correct rubric set containing multiple {grading points: specific score} pairs, and AT denotes the automatically generated rubric set. $\mathbb{I}(i \in GT)$ is an indicator function that equals 1 if the item i from AT is present in GT, and 0 otherwise. The prompts used for this evaluation are detailed in Appendix C.2.

$$\text{Match}(GT, AT) = \frac{\sum_{i \in AT} \mathbb{I}(i \in GT)}{|GT|} \quad (8)$$

5.3 Selected Models

Previous studies predominantly employ English-proficient models to generate <question, response, score> triples for training evaluator language models, often overlooking models proficient in Chinese. Additionally, several studies exclusively use GPT-3.5 or GPT-4 to construct such synthetic data. These data generation methodologies may cause discrepancies between the synthetic and real-world data distributions, introducing biases into the trained evaluator language models.

To alleviate this issue, we utilize a broader range of LLMs to collect responses that better reflect real-world distributions. This approach ensures greater diversity and mitigates biases introduced by relying solely on synthetic data generated from a single model. Specifically, we choose GPT-4, GPT-4-turbo, GPT-4-o, Claude Opus, DeepSeek 2.0, MiniMax 6.5, MiniMax 6, Doubao, GPT-3.5, Tongyi Qianwen 2.0, and Tongyi Qianwen 1.5-100B/70B. This selection includes models proficient in different languages and multiple versions of the same model.

For open-source models, we use local deployment to infer responses. For proprietary LLMs, if API services are available, we collect model outputs by requesting the API. If only a web interface is provided, we employ people to gather the outputs.

6 Analysis

In this section, we conduct a comprehensive experimental analysis of the robustness of the proposed benchmark evaluations, examining the data distribution, training phases, and training paradigms of evaluator LMs. Our findings reveal limitations in current training methodologies for evaluator LMs. Building on these insights, we develop an evaluator LM aligned with the self-adaptive rubrics paradigm.

6.1 Scaling Law for Robust Evaluation

A robust benchmark should effectively distinguish the capabilities of different models and maintain stability to ensure consistent rankings rather than allowing fluctuations due to the instability of individual tasks. To achieve this, the benchmark needs a sufficiently broad distribution while minimizing extraneous biases.

To verify the robustness of the proposed benchmark, we conduct two rounds of sampling without replacement from a pool of 1,000 questions. In each round, we select n questions, resulting in a total of $2n$ independent questions, where $n \in [10, 500]$. We then compare the consistency of the model rankings obtained from these two samples.

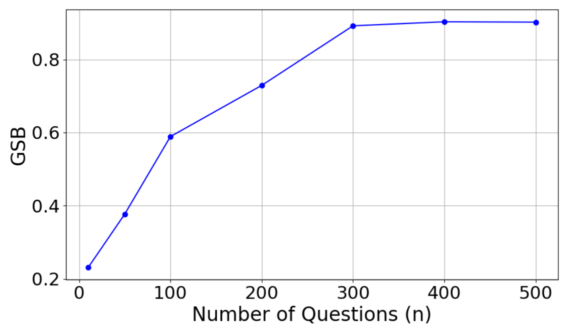

Figure 2: Consistency of model rankings as n increases. After n reaches approximately 300, the consistency stabilizes with only minor fluctuations.

Figure 2 shows the variation in the consistency of model rankings under different question sets as n increases. When n is relatively small, the consistency is low, indicating the inconsistency caused by biases in different distributions. As n increases, the consistency improves despite the two sets remaining independent. After n reaches approximately 300, the consistency stabilizes with only minor fluctuations. This demonstrates the scaling law for robust evaluation, indicating that as the number of questions increases, the evaluation results tend to stabilize due to the broader coverage of the distribution.

6.2 Score Distribution Shift

Figure 3: Data distribution comparison of different data.

The Prometheus approach, which relies on general rubrics not specifically tailored to the problem at hand, employs GPT-4 to generate an equal amount of synthetic data across different scores (1-5) to mitigate score bias from the evaluating language model. In contrast, our method uses self-adaptive rubrics, and our responses are genuinely collected from the model rather than artificially synthesized. Consequently, we cannot ensure that the quantity of data for each score is perfectly balanced.

However, as illustrated in Figure 3, we observe that despite the score distribution shift between the training and test data, the score distribution of the model outputs, trained using the self-adaptive rubrics paradigm, closely aligns with the human-provided ground truth. This finding substantiates the robustness and efficacy of the self-adaptive rubrics paradigm in automated scoring.

6.3 Out of Distribution Evaluation

We establish two dimensions for evaluating the out-of-distribution capabilities of our model: model-level and question-level. For the model-level evaluation, we utilize the same set of questions, selecting a subset of models to train the evaluator language model (LM), and subsequently test on the remaining unselected models. In the question-level evaluation, a subset of questions along with all associated models are used for training, and the scoring performance is then assessed on a different set of questions.

Table 1 presents the experimental results, showing that under the self-adaptive rubrics paradigm, the model performs well in both model-level and question-level evaluations. This indicates that our proposed method has strong generalization capabilities.

6.4 Merged SFT or Continual SFT

Previous research shows that a model might generate a correct answer but fail to accurately evaluate the <question, answer> pair for the same question. We argue that this issue mainly arises from the insufficient diversity of the SFT data.

To validate this, we conduct the following experiments:

1. Training a pretrained language model (PLM) using only traditional SFT data.
2. Training a PLM using a mix of SFT data and evaluator LM format data.
3. Performing continual SFT on an instruction-tuned model using evaluator LM format data, a widely adopted approach in other studies.

As shown in Table 2, we find that although the model using continual SFT performs well on evaluation tasks, its general ability significantly declines, limiting its versatility. However, starting from a PLM and using a mix of SFT data and evaluator LM format data for SFT results in excellent evaluation capability with minimal impact on general ability. This reveals the shortcomings of the previous continual SFT approach and suggests that the model's inability to evaluate the questions it can answer may simply be due to the lack of such data, highlighting the importance of diversity in SFT data.

We employ Human-AI Consistency to filter the evaluator LM and find that, compared to using raw Chain-of-Thought data generated by GPT-4 and data filtered by rejection sampling, the data selected using Human-AI Consistency shows significant improvements in both evaluation and general capability, demonstrating the effectiveness of this strategy.

6.5 Ablation Study

We conduct detailed ablation experiments on the components of self-adaptive rubrics, namely, reference answers, rubrics, and in-context examples. As shown in Table 3, the consistency between the

	Question-level				Model-level			
Type	GSB	ACC	ACC(t)	pearsonr	GSB	ACC	ACC(t)	pearsonr
XDG	0.952	0.590	0.794	0.738	0.952	0.590	0.794	0.380
GPT-3.5	0.829	0.422	0.663	0.566	0.829	0.422	0.663	0.566
GPT-4	0.952	0.654	0.855	0.822	0.952	0.654	0.855	0.822

Table 1: Out of distribution evaluation performance in both model-level and question-level.

Type	GSB	ACC	ACC(t)	pearsonr	general
baseline	0.784	0.339	0.584	0.263	730
XDG-v1	0.910	0.514	0.755	0.686	458
XDG-v2	0.895	0.551	0.802	0.761	684
XDG-v3	0.911	0.593	0.811	0.765	653
XDG-v4	**0.941**	**0.664**	**0.854**	**0.829**	685

Table 2: Experiments on training phases and training data, where V1 represents continual SFT, V2 represents SFT from PLM, V3 represents SFT incorporating evaluator LM format data, and V4 represents data filtered using the Human-AI Consistency strategy.

evaluator LM and human scoring significantly increases after incorporating self-adaptive rubrics. However, the improvements are not as pronounced when other components are added, indicating that the primary driver of enhanced performance is the self-adaptive rubrics themselves. This suggests that self-adaptive rubrics play a crucial role in aligning the evaluator LM with human judgment.

Type	GSB	ACC	ACC_t	pearsonr
Baseline	0.963	0.636	0.802	0.733
+ rubric	0.957	0.706	0.871	0.843
+ R.A	0.952	0.717	0.877	0.848
+ example	0.959	0.728	0.888	0.867

Table 3: Ablation study for each component, where R.A. stands for reference answer.

6.6 Comparison with Alternative Paradigm

Using the same training data, we conduct a comparative analysis between the self-adaptive rubrics paradigm and the existing general rubric paradigm, as presented in Table 4. The results demonstrate that our approach significantly outperforms existing methods. Furthermore, in addition to accurately ranking the models, our method provides fine-grained capability evaluations that closely align with human assessments. This is both crucial and practical for facilitating the iterative development of LLMs. Due to space constraints, detailed descriptions and results of other experiments are provided in Appendix E.

7 Conclusion

In this paper, we introduce a novel evaluation paradigm called self-adaptive rubrics, aligning the scoring process with human judgment and reducing bias by tailoring rubrics to specific questions. Based on this paradigm, we develop a new benchmark, INSDA. To automate scoring, we analyze existing open-source evaluator language models and identify training phase data diversity issues. We then introduce human-AI consistency to align the chain-of-thought evaluation with human judgment and propose an evaluator LM that follows the self-adaptive rubrics paradigm. Experimental results show our model achieves higher consistency with human evaluation compared to GPT-4. We hope our work inspires researchers to apply this paradigm to more tasks, aligning automated scoring with human judgment.

Limitations

In this paper, we propose an evaluation paradigm based on self-adaptive rubrics, which provides more granular process guidance to align the scoring process with human judgment. Additionally, we introduce a benchmark, INSDA, based on this framework. However, there are several limitations:

- For questions with multiple correct answers, it requires manually writing multiple self-adaptive rubrics. It is worth noting that, to our knowledge, no current work focuses on the multi-solution direction.

- For subjective questions, such as creative writing, poetry, and other forms of artistic expression, different groups or individuals may have varying definitions of what constitutes good work. Therefore, it is necessary for each group or individual to set their own self-adaptive rubrics rather than relying on predefined ones. This also highlights the flexibility and interpretability of the self-adaptive rubrics paradigm we propose.

Ethical Considerations

We propose a scoring paradigm based on self-adaptive rubrics to enhance the interpretability and controllability of the automated scoring process. This approach aims to improve the credibility of evaluation results produced by LLMs and to support the research community in advancing these models. Nevertheless, the inherent hallucinations within LLMs pose a challenge to ensuring the complete accuracy of automated evaluation outcomes. Therefore, we recommend incorporating human review of certain outputs when using LLMs as judges to increase the overall reliability and credibility of the process.

Additionally, when generating self-adaptive rubrics for subjective questions, different groups or individuals may have varying definitions of what constitutes a good answer, potentially leading to biases and discrepancies. We encourage dialogue and mutual understanding among groups or individuals with diverse values, promoting the use of self-adaptive rubrics that align with their respective values and preferences.

References

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *Preprint*, arXiv:2308.07201.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A statistical analysis of summarization evaluation metrics using resampling methods. *Transactions of the Association for Computational Linguistics*, 9:1132–1146.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. 2024. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. Prometheus: Inducing fine-grained evaluation capability in language models. *Preprint*, arXiv:2310.08491.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024b. Prometheus 2: An open source language model specialized in evaluating other language models. *Preprint*, arXiv:2405.01535.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. *arXiv preprint arXiv:2107.10821*.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. *arXiv preprint arXiv:2103.06332*.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023a. Generative judge for evaluating alignment. *Preprint*, arXiv:2310.05470.

Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. 2019. Deep reinforcement learning with distributional semantic rewards for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6038–6044, Hong Kong, China. Association for Computational Linguistics.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023b. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. *Preprint*, arXiv:1904.03971.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. 2021. Semantic answer similarity for evaluating question answering models. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 149–157, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Natalie Schluter. 2017a. The limits of automatic summarisation according to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45, Valencia, Spain. Association for Computational Linguistics.

Natalie Schluter. 2017b. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 41–45. Association for Computational Linguistics.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2020. Megatron-lm: Training multi-billion parameter language models using model parallelism. *Preprint*, arXiv:1909.08053.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Characterizing the confidence of large language model-based automatic evaluation metrics. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 76–89.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. 2024. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *Preprint*, arXiv:2306.05087.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2023. Flask: Fine-grained language model evaluation based on alignment skill sets. *arXiv preprint arXiv:2307.10928*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Ying Zhang and Stephan Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of the 10th Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

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. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *Preprint*, arXiv:2310.17631.

A Training Details

We employed a default learning rate of 2e-5, and the batch size per device was dynamically adjusted based on the total data volume and the number of machines to maintain consistent optimization steps. The Adam optimizer was utilized.
For the scaling law experiments, when n was below 300, we conducted three repetitions and averaged the results to minimize error.
For all invoked APIs, we used the default parameters without extensive modifications.

B Data Annotation

B.1 Annotator Qualifications

All our annotators are internal team members with at least a Master's degree. We provide additional compensation significantly higher than the standard salary, based on the amount of data annotated.

C Prompt Templates

C.1 AutoPrompt

> Below, I will provide a <Question>, along with the corresponding <Reference Answer> and <Scoring Rubric>. You need to evaluate the <Output Result> from the <Model Answer> of the <Model to be Assessed>. The evaluation should be divided into two parts: "Scoring Process" and "Final Score." Please note that the scoring range is from 0 to 5 points. You must justify the score you assign based on the <Model Answer>, strictly adhering to the requirements of the <Scoring Rubric> without adding, changing, or imagining any additional criteria.

C.2 Prompt for Set Matching

> You are a meticulous judge tasked with evaluating whether the "Test Rubric" provided by the user aligns with the "Standard Rubric." The evaluation rules are as follows:
>
> - The initial total score is set to zero.
> - For each item in the "Test Rubric":
>   1. If the item matches any item in the "Standard Rubric" exactly, one point is added to the total score.
>   2. If the item in the "Test Rubric" is unrelated to any item in the "Standard Rubric," the total score remains unchanged.
>   3. If the item in the "Test Rubric" is the exact opposite of any item in the "Standard Rubric," one point is subtracted from the total score.
>
> You need to return the entire scoring process (explaining why points were added or subtracted) along with the final score. The return format should be:
>
> { "Scoring Process": "<Here, provide the scoring process as a string>",
> "Final Score": "<Here, provide the final score as a mathematical expression, concluding with 'Final Score: <score>' e.g., '3/5=0.6, Final Score: 0.6'>" }
>
> The returned format must be compatible with `json.loads()` to be converted into a dictionary.

C.3 Prosecutor Prompt

> Please check if the generated answer is correct. The reference answer is: {gt}, and the generated answer is: {user}. Please respond in the following format: { "result": True }

C.4 Refinement Prompt

> Your generated answer is not the standard answer. Please reflect on this and generate a new answer.
> The generated scoring points and the full score answer are:

D Benchmark

D.1 Benchmark Leaderboard

We provide the Benchmark LeaderBoard at the following anonymous link: https://anonymous.4open.science/r/self-adaptive-rubrics-4F62

D.2 Benchmark Statistics

We provide the Benchmark statistics at the following anonymous link: https://anonymous.4open.science/r/self-adaptive-rubrics-4F62

E Additional Experiments

E.1 Automatic Rubric Generation

Algorithm 1 Self-Adaptive Rubrics Generation and Validation

Require: Q {Given question}
Require: GT {Ground truth answer}
Require: n {Maximum iterations}
$i \leftarrow 0$
$accepted \leftarrow$ False
while $i < n$ **and** $\neg accepted$ **do**
  $R, IA \leftarrow$ GPT-4(Q) {Generate rubrics and ideal answer}
  if $PA(IA, GT)$ **then**
    {Prosecutor agent checks ideal answer}
    $accepted \leftarrow$ True
  else
    Inform GPT-4 of incorrect IA
    $i \leftarrow i + 1$
  end if
end while
if $accepted$ **then**
  return R
else
  return Failure in n iterations
end if

E.2 Error Propagating

When using rubric generation models to automatically create self-adaptive rubrics, a potential issue is that if the generated rubric is inconsistent with the human-provided rubric, errors can accumulate in the scoring pipeline, leading to a larger deviation in the final score. By incorporating a filtering strategy, the overall performance will improve.

E.3 Joint Training vs. Expert Training

We also explored whether to combine data from different categories for joint training when training the evaluator LM or rubric generation model, or to train a separate expert model for each category individually. We found that using joint training can achieve better results than expert training.

Model	Type	ACC	ACC_t	Pearsonr	GSB
GPT-4	w self-adaptive rubrics	0.3241	0.7025	0.7283	0.9211
GPT-4	wo	0.2500	0.5035	0.4863	0.8684
GPT-4-turbo	w	-	-	-	-
GPT-4-turbo	wo	0.1995	0.5473	0.5326	0.9000
GPT-3.5	w	0.2121	0.5569	0.3888	0.8947
GPT-3.5	wo	0.1677	0.3872	0.1286	0.5158

Table 4: Ablation Study

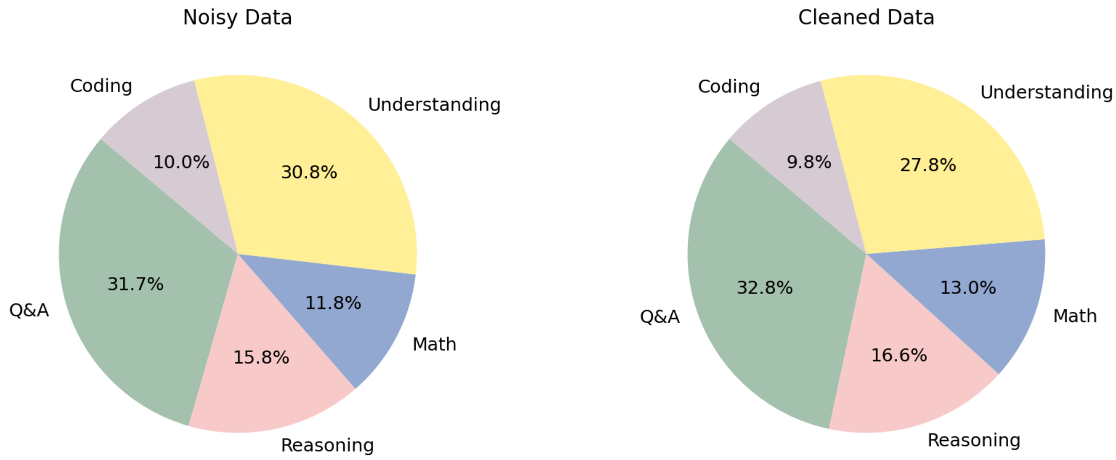

Figure 4: Distribution change of the evaluator LM format data after applying the Human-AI Consistency strategy.

Type	OOD	ID	Random
GPT4-turbo	0.399	0.488	0.417
GPT4	0.602	0.613	0.613
xdg-turbo	0.606	0.607	0.603

Table 5: Rubric Generation Results.

Type	GSB	ACC	ACC_t	pearsonr
GPT-4	0.921	0.324	0.702	0.728
GPT-3.5	0.894	0.212	0.556	0.388

Table 6: General Rubrics with Ground Truth.

Type	GSB	ACC	ACC_t	pearsonr
GPT-4-turbo	0.9	0.199	0.547	0.532
GPT-4	0.868	0.250	0.503	0.486
GPT-3.5	0.515	0.167	0.0.387	0.128

Table 7: General Rubrics without Groud Truth.