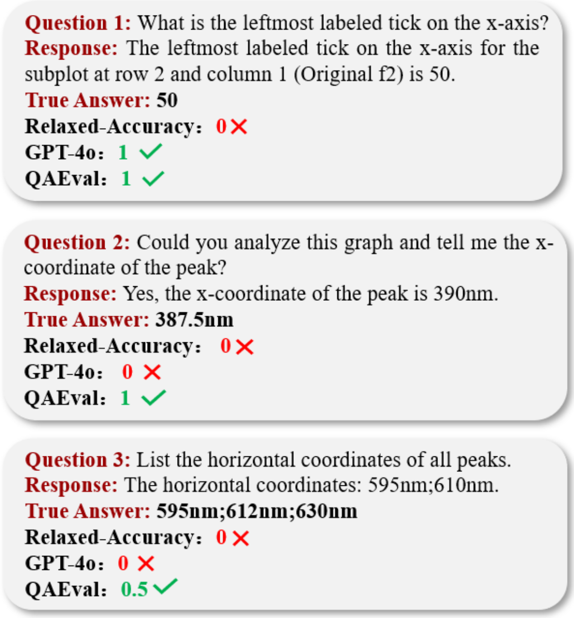
QAEval: Mixture of Evaluators for Question-Answering Task Evaluation

Tan Yue[1], **Rui Mao**[3], **Xuzhao Shi**[4],
Shuo Zhan[3], **Zuhao Yang**[3], **Dongyan Zhao**[1,2]*
[1]Wangxuan Institute of Computer Technology, Peking University
[2]State Key Laboratory of General Artificial Intelligence
[3]Nanyang Technological University
[4]Beijing University of Posts and Telecommunications
{yuetan,zhaodongyan}@pku.edu.cn, rui.mao@ntu.edu.sg,
sxzs@bupt.edu.cn, {zhan0590,yang0756}@e.ntu.edu.sg

Abstract

Question answering (QA) tasks serve as a key benchmark for evaluating generation systems. Traditional rule-based metrics, such as accuracy and relaxed-accuracy, struggle with open-ended and unstructured responses. LLM-based evaluation methods offer greater flexibility but suffer from sensitivity to instructions, robustness issues, and high computational costs. To overcome these challenges, we introduce QAEval, a hybrid framework combining rule-based reliability with LLM-based adaptability. QAEval utilizes two high-quality datasets: QAExtract for short-answer extraction and QAScore for scoring model training. By integrating a Mixture of Evaluators model with Dynamic Load Balancing Optimization, QAEval enables accurate, cost-effective QA evaluation. Experimental results show it outperforms models like GPT-4o and Claude-3, achieving 92.3% accuracy with only 0.6B parameters.



Question 1: What is the leftmost labeled tick on the x-axis?
Response: The leftmost labeled tick on the x-axis for the subplot at row 2 and column 1 (Original f2) is 50.
True Answer: 50
Relaxed-Accuracy: 0 ✗
GPT-4o: 1 ✓
QAEval: 1 ✓

Question 2: Could you analyze this graph and tell me the x-coordinate of the peak?
Response: Yes, the x-coordinate of the peak is 390nm.
True Answer: 387.5nm
Relaxed-Accuracy: 0 ✗
GPT-4o: 0 ✗
QAEval: 1 ✓

Question 3: List the horizontal coordinates of all peaks.
Response: The horizontal coordinates: 595nm;610nm.
True Answer: 595nm;612nm;630nm
Relaxed-Accuracy: 0 ✗
GPT-4o: 0 ✗
QAEval: 0.5 ✓

Figure 1: Comparison of complex QA evaluations using different methods, including our proposed QAEval.

1 Introduction

Question answering (QA) systems (Ojokoh and Adebisi, 2018; Zaib et al., 2022) serve as an important medium of human-computer interaction, with diverse applications, such as information retrieval for solving complex problems (Masry et al., 2022; Liu et al., 2023; Welivita and Pu, 2023). With the advancement of large language models (LLMs), QA tasks have become a crucial benchmark for assessing their performance (Krithara et al., 2023; Mao et al., 2024). Robust QA evaluation becomes essential for understanding the strengths and limitations of LLMs, particularly in reasoning, factual accuracy, and contextual comprehension.

Existing QA evaluation methods exhibit notable limitations in assessing complex and open-ended responses (Wang et al., 2024). Traditional rule-based approaches rely on strict matching criteria to compute accuracy (Kahou et al., 2017), making them struggle with open-ended questions that require contextual understanding. While relaxed-accuracy metrics (Methani et al., 2020) improve flexibility by allowing numerical tolerance, they remain insufficient for capturing contextualized correctness. More recently, LLM-based evaluators have gained popularity due to their ability to assess responses holistically. These methods score the input *[Question, True-answer, Model-response]* in a binary classification ([0, 1]) by using tailored prompts (Xia et al., 2024; Wang et al., 2024). However, these methods introduce new challenges, including sensitivity to prompt variations (Mao et al., 2023), evaluation inconsistencies across different LLM versions, and high computational costs (Shekhar et al., 2024). These constraints hinder their reliability and scalability, particularly for large-scale evaluation tasks where ef-

*Corresponding author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14717–14730
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

ficiency and stability are critical. Typical errors presented in relaxed-accuracy and GPT-4o-based evaluation methods are shown in Fig. 1.

To address these challenges, we propose QAEval[1], a hybrid evaluation framework that combines the robustness of rule-based methods with the adaptability of language model-based evaluation. Unlike existing LLM-based methods, QAEval is designed to be lightweight, plug-and-play, and cost-efficient while maintaining high accuracy. Our method utilizes a Mixture of Evaluators (MOE) model, trained with Dynamic Load Balancing Optimization (DLBO), which leverages Kullback-Leibler (KL) dispersion to enhance evaluation robustness. Additionally, we introduce two manually labeled datasets, QAExtract with 9,889 samples and QAScore with 14,419 samples, to refine both extraction and scoring capabilities. Experimental results on QAScore test set demonstrate that QAEval achieves state-of-the-art performance (92.3%), surpassing models such as GPT-4o and Claude-3 while using only 0.6B parameters, significantly reducing computational overhead without compromising accuracy.

The contributions of this work are summarized as follows: 1) We introduce QAEval, a novel evaluation framework that efficiently and accurately assesses QA tasks by integrating an answer extraction model, a rule-based scoring model, and a scoring model utilizing MOE. The framework is lightweight, and highly robust, supporting a plug-and-play deployment approach. 2) To enhance the robustness of the MOE scoring model, we propose DLBO, which dynamically balances evaluator selection propensity and evaluation accuracy, ensuring more stable and reliable performance. 3) We construct two high-quality datasets: QAExtract for answer extraction and QAScore for scoring, by sampling data from 31 QA datasets and generating responses using 10 mainstream LLMs. All data is manually labeled and thoroughly proofread to ensure dataset quality and reliability.

2 Related works

With the rapid development of LLMs, QA tasks have become an important benchmark for testing model understanding (Yue et al., 2023; Kim et al., 2024) and reasoning capabilities.

Early QA datasets, such as FigureQA (Kahou et al., 2017) and DVQA (Kafle et al., 2018), primarily rely on fixed-answer formats, where responses are constrained to binary ("yes/no") or predefined categorical labels. These datasets are evaluated using accuracy, which directly measure the match between model outputs and ground-truth answers (see Fig. 2A). While effective for structured tasks, this rigid evaluation framework struggle to accommodate more complex QA tasks that involve open-ended or numerical responses. To address this, relaxed-accuracy is introduced (Methani et al., 2020), allowing for a tolerance range in numerical answers and improving flexibility. Beyond numeric relaxation, recent rule-based methods (Bulian et al., 2022; Li et al., 2024) further enhance equivalence judgment through manually defined transformation rules, enabling interpretable matching for paraphrased or reformatted answers. However, as QA tasks continue to grow in complexity (Ma et al., 2024), particularly with the emergence of LLMs capable of generating diverse and unconstrained responses, traditional rule-based methods fail to provide a comprehensive assessment of response quality. The inherent variability in LLM outputs, such as paraphrased responses or contextual interpretations (Yue et al., 2021; Zhu et al., 2024; Verga et al., 2024), makes it difficult for rule-based methods to accurately capture semantic equivalence (Wang et al., 2022; Wang, 2023; Yue et al., 2024).

To overcome these limitations, LLM-based evaluation methods (Achiam et al., 2023; Xia et al., 2024) have been developed, leveraging the contextual interpretation capabilities of LLMs (see Fig. 2B). These approaches instruct an LLM to assess responses by comparing the *[Question, True-answer, Model-response]* tuple, where the evaluation task is often treated as a binary classification problem, assigning a score of "1" for correct answers and "0" for incorrect answers. Some methods, such as Charxiv (Wang et al., 2024), instruct LLMs to first extract key information from lengthy model responses and then score them. While LLM-based evaluation improves flexibility by handling diverse and unstructured responses, it also introduces significant challenges. LLMs are prone to hallucinations and inconsistent responses, leading to unreliable or unexplainable scoring decisions. Furthermore, their sensitivity to prompt variations, dependence on proprietary model versions, and high computational costs hinder their robustness and scalability for large-scale evaluation tasks (Chang et al., 2024).

In summary, rule-based evaluation approaches

[1] https://github.com/yuetanbupt/QAEval

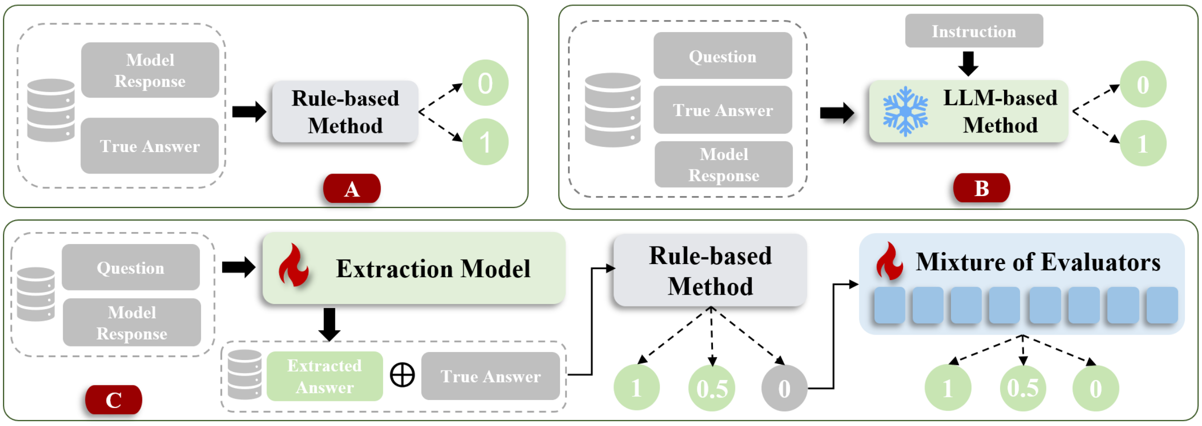


Figure 2: Comparison of the proposed QAEval framework with existing methods. Figure 2A represents the rule-based method, Figure 2B represents the LLM-based method, and Figure 2C shows our QAEval framework.

struggle with model-generated responses that vary in structure and phrasing, leading to underestimation of correct answers due to limited semantic flexibility. While LLMs enable flexible response assessment (Vu et al., 2024), they frequently produce hallucinated errors and lack interpretability, making their scoring inconsistent and difficult to verify. LLM-based evaluators exhibit poor robustness, with significant fluctuations in results due to slight prompt variations (Xu et al., 2023; Chiang and Lee, 2023). Their reliance on proprietary models introduces version inconsistencies, while high API/deployment costs and latency make them impractical for large-scale QA evaluation.

3 Methodology

We propose QAEval, a novel framework designed for efficient and accurate evaluation of QA tasks. As illustrated in Fig. 2C, QAEval comprises three key components: an answer extraction model, a rule-based scoring model, and an MOE scoring model. By integrating extraction with a multi-stage scoring approach, QAEval enhances evaluation accuracy while maintaining a lightweight model architecture.

QAEval evaluates each input sample, consisting of *[Question, True Answer, Model Response]*, via a multi-step process: **1) Answer Extraction.** Short-form answers are extracted from lengthy model-generated responses to standardize the evaluation input. **2) Rule-Based Quick Scoring.** The extracted answers are first assessed using a improved rule-based scoring method for rapid evaluation. **3) Flexible Scoring via MOE.** If the rule-based approach fails to establish a reliable match between the extracted and true answers, an MOE scoring model is employed to provide a more flexible and context-aware assessment. The final evaluation output is a three-tiered scoring system ([0, 0.5, 1]), capturing varying degrees of answer correctness.

3.1 Answer extraction

The answer extraction model is designed to distill concise answers from lengthy QA responses, facilitating a more straightforward comparison with ground-truth answers. To achieve this, we fine-tune a Qwen2.5-0.5B model using our curated QAExtract dataset. Through instruction fine-tuning, the model learns to extract key information from complex and verbose responses, ensuring more efficient and accurate matching during evaluation.

The extraction model takes the input $[Q, R]$, where Q represents the question and R represents the model-generated response. The output A_{pre} is the extracted information from the QA system response. During training, the model learns to minimize the loss between the predicted answer A_{pre} and the true extracted answer A_{ext} with the following the loss $\mathcal{L}_{\text{extract}} = -\sum_{t=1}^{T} \log P(A_t|A_{<t}, Q, R; \theta)$, where T is the length of the true extracted answer A_{ext}; A_t is the t-th token of the true extracted answer; $A_{<t}$ represents the sequence of tokens generated before A_t at step t; $P(A_t|A_{<t}, Q, R; \theta)$ is the probability of generating token A_t conditioned on the previous tokens, question Q, response R, and model parameters θ. During inference, the model extracts the short answer by finding the sequence A_{pre} that maximizes the joint probability of all tokens by $A_{\text{pre}} = \text{argmax}_A \prod_{t=1}^{T} P(A_t|A_{<t}, Q, R; \theta)$.

3.2 Rule-based quick scoring

Based on the relaxed-accuracy method (Methani et al., 2020), we propose an improved rule-based evaluation method (R_{acc}) that aims to measure the prediction results of the model more comprehensively. While retaining the error-tolerance mechanism for numerical answers, the method adds a case-insensitive matching algorithm for string answers, thus increasing the flexibility and applicability of the evaluation. In addition, to address the limitation that the relaxed-accuracy method only provides two scores, 0 or 1, we introduce an intermediate score of 0.5 to identify partially correct answer cases. When dealing with multiple-answer samples, we match each model-predicted answer individually with the corresponding true answer to ensure evaluation accuracy. The implementation details are shown in Algorithm 1.

The rule-based method provides a fast and reliable preliminary evaluation using predefined matching rules. In the initial scoring phase, it calculates a score S, based on predefined matching rules. The scoring function is given by:

$$S = \begin{cases} 1, & \text{if } R_{acc}(A_{\text{pre}}) = A_{\text{true}} \\ 0.5, & \text{if } R_{acc}(A_{\text{pre}}) \in A_{\text{true}} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

If the rule-based score S is 0.5 or 1, the extracted answer is deemed a match, and the score is finalized. If $S = 0$, indicating no match or an incorrect response, the filtered sample (denoted as F) is passed to the MOE model for further assessment.

$$\text{FinalScore} = \begin{cases} S, & \text{if } S = [0.5, 1] \\ \text{MOE}(F), & \text{if } S = 0 \end{cases} \quad (2)$$

This two-stage scoring strategy merges the reliability and efficiency of the rule-based method with the flexibility of the MOE model, offering a balanced solution for complex QA tasks. It reduces computational costs while maintaining accuracy.

3.3 Mixture of evaluators

We develop a scoring model using MOE (Fan et al., 2024; Jiang et al., 2024; Cai et al., 2024; Guo et al., 2025) to handle complex QA responses that rule-based methods cannot match. Unlike LLM-based evaluators with a generative paradigm, we treat QA scoring as a classification task and train a classifier. The advantage is that the classification approach directly maps the evaluation outcomes to predefined categories, which enhances stability and transparency. It also avoids the unpredictability of generative models and reduces the risk of hallucinations. To address the evaluation accuracy drop caused by diverse QA types, we introduce an MOE architecture with DLBO, which improves scoring accuracy by assigning specific QA types to evaluators (see Fig. 3). The input is defined as *E_input* for evaluators and *G_input* for the gating network, where *G_input* $= [..., [Q, A_{\text{true}}]_i, ...]$, *E_input* $= [..., [Q, A_{\text{true}}, R, A_{\text{pre}}]_i, ...], \forall i \in [1, N]$. N is the batch size.

Algorithm 1 Rule-based Scoring Algorithm

```
Function Correctness($t, p, \delta = 0.05$)
Convert $t$ and $p$ to lowercase (for string comparison)
Define toFloat(x): parse $x$ (original or lowercased) as a
  float; if ends with ‘%‘, remove ‘%‘ and divide by 100
$t_f \leftarrow$ toFloat($t$), $p_f \leftarrow$ toFloat($p$)
if $t_f$ and $p_f$ are not None then
    if $t_f == 0$ then
        return ($p_f == 0$)
    end if
    Compute $\Delta \leftarrow \frac{|p_f - t_f|}{|t_f|}$
    return ($\Delta \leq \delta$)
else
    return ($t == p$)
end if

Function Score($R, A$)
Trim leading/trailing spaces in $R, A$, then split $R$ and $A$
  into lists (by delimiter ‘;‘)
if $|R| > |A|$ then
    return 0
end if
Initialize match ← [ ]
for $r$ in $R$ do
    $m \leftarrow$ any(Correctness($a, r, \delta$) for $a$ in $A$)
    Append $m$ to match
end for
if all(match) and $|R| == |A|$ then
    return 1
else if all(match) then
    return 0.5
else
    return 0
end if
```

First, a gating network has been designed, which assigns a weight distribution based on the QA type of the input. It dynamically directs different sample types to the most suitable evaluators, ensuring specialization. Features $\mathbf{H}_{\text{CLS}} = \mathcal{M}^A(\text{G\_input})$, are extracted from the ALBERT model ($\mathcal{M}^A(\cdot)$) (Lan, 2019) using the [CLS] token representation, where $\mathbf{H}_{\text{CLS}} \in \mathbb{R}^{N \times d}$ is the [CLS] feature vector. d is the hidden layer size. The gating network takes $\mathbf{H}_{\text{CLS}}$ as input and computes the weight distribution across evaluators. $\mathbf{G} = \sigma(\frac{\mathbf{H}_{\text{CLS}} \cdot \mathbf{W}_g + \mathbf{b}_g}{2})$, where $\mathbf{W}_g \in \mathbb{R}^{d \times K}$is the weight matrix of the gating network. $\mathbf{b}_g$ is the bias vector. K denotes the

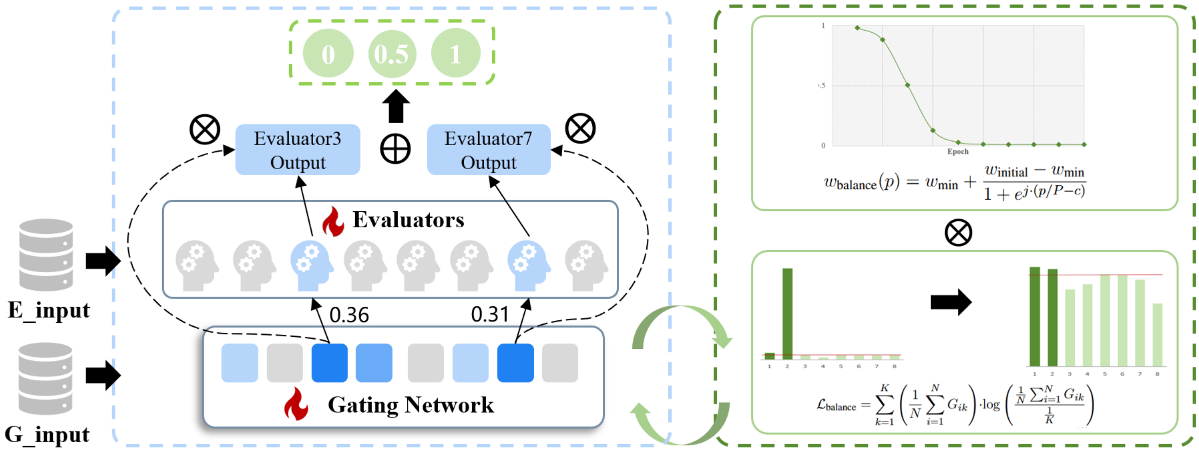

0
0.5
1
Evaluator3 Output
Evaluator7 Output
Evaluators
E_input
G_input
0.36
0.31
Gating Network
1
.5
0
Epoch
$w_{\text{balance}}(p) = w_{\min} + \frac{w_{\text{initial}} - w_{\min}}{1 + e^{j\cdot(p/P-c)}}$
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
$\mathcal{L}_{\text{balance}} = \sum_{k=1}^{K}\left(\frac{1}{N}\sum_{i=1}^{N} G_{ik}\right)\cdot\log\left(\frac{\frac{1}{N}\sum_{i=1}^{N} G_{ik}}{\frac{1}{K}}\right)$

Figure 3: The proposed MOE scoring model. The left side of the figure illustrates the model scoring process, and the right side represents the DLBO for the gating network.

number of evaluators. $\mathbf{G} \in \mathbb{R}^{N\times K}$ is the weight distribution generated by the gating network. σ is a softmax function.

Next, we develop a gating network to select the best evaluators for processing the evaluation task. Based on the gating network's output (the weight values), we select the Top2 evaluators with the highest weights. Each evaluator independently performs the classification. The outputs of the selected evaluators are fused with their weights: $\mathbf{y}_{\text{fused}} = \sum_{k=1}^{K} \sigma(\text{Top2}(\mathbf{G})) \cdot \mathcal{M}^A(\text{E\_input}_i)$.

Finally, during MOE training, the model may over-rely on one evaluator, diminishing the utility of others. To address this, we introduce Kullback-Leibler (KL) divergence to create a DLBO function. By applying a weight decay mechanism, the gating network's weights are dynamically adjusted across epochs, ensuring balanced load distribution and enhancing overall model performance. The KL divergence measures the difference between the evaluator distribution and a uniform distribution:

$$\mathcal{L}_{\text{balance}} = \sum_{k=1}^{K}\left(\frac{1}{N}\sum_{i=1}^{N} G_{ik}\right)\cdot\log\left(\frac{\frac{1}{N}\sum_{i=1}^{N} G_{ik}}{\frac{1}{K}}\right), \tag{3}$$

where k is the indexes of evaluators. An improved weight decay function dynamically adjusts the balancing weight by

$$w_{\text{balance}}(p) = w_{\min} + \frac{w_{\text{initial}} - w_{\min}}{1 + e^{j\cdot(p/P-c)}}, \tag{4}$$

where $w_{\min}$ and w_{initial} are the minimum and initial weights. p is the current epoch, P is the total number of epochs. $c = 0.5 - f$, where f is the early adjustment factor. j is the curve steepness factor. The overall loss is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + w_{\text{balance}}(p) \cdot \mathcal{L}_{\text{balance}}$.

The DLBO function aims to constrain the average evaluator weight $\frac{1}{N}\sum_{i=1}^{N} G_{ik}$ to close to the target uniform distribution $\frac{1}{K}$, avoiding over-reliance on specific evaluators. The function is defined as the weighted logarithmic difference between the average evaluator weight and the target distribution. During the optimization, load balancing is achieved by updating the gradient of each sample weight G_{ik} (we define $\mu_k = \frac{1}{N}\sum_{i=1}^{N} G_{ik}$):

$$\frac{\partial \mathcal{L}_{\text{balance}}}{\partial G_{ik}} = \frac{\partial \mathcal{L}_{\text{balance}}}{\partial \mu_k} \cdot \frac{\partial \mu_k}{\partial G_{ik}}, \tag{5}$$

$$\frac{\partial \mathcal{L}_{\text{balance}}}{\partial \mu_k} = \log\left(\frac{\mu_k}{\frac{1}{K}}\right) + 1, \tag{6}$$

$$\frac{\partial \mathcal{L}_{\text{balance}}}{\partial G_{ik}} = \frac{1}{N}\left(\log\frac{\frac{1}{N}\sum_{i=1}^{N} G_{ik}}{\frac{1}{K}} + 1\right). \tag{7}$$

In the early training stages, higher dynamic weights cause greater gradient adjustments when evaluator weights deviate from the target distribution. DLBO ensures that evaluators are exposed to more data initially. The weights decay with training. In the later stages, gradients focus on prediction differences, allowing better-performing evaluators to receive higher weights for their specialized QA tasks, ultimately improving model accuracy.

4 Dataset

4.1 Dataset collection

To train the extraction and scoring models, we build two quality, manually labeled datasets: QAExtract

Dataset	Train	Val.	Test	Total
QAExtract	7889	2000	-	9,889
QAScore	12419	1000	1000	14,419

Table 1: QAExtract and QAScore statistics.

and QAScore. Both are collected from 31 widely-used datasets for QA, covering a wide range of domains, including figure/chart QA, textbook QA, visual QA, etc (see Table A.2 in Appendix A for details). In the initial data collection phase, 60,000 QA samples are drawn from 31 datasets. However, the prevalence of homogeneous and simple QA pairs pose a risk of ineffective training. To address this, similar samples within the same dataset are randomly selected, followed by manual filtering, reducing the dataset to 24,000 samples while ensuring quality and diversity (shown in Table 1). The selection process adhered to three key principles: 1) maintaining a balanced data distribution by controlling the proportion of similar QA samples; 2) limiting the inclusion of simple QA pairs to a small set of representative examples; 3) increasing the proportion of complex QA samples to enhance dataset difficulty and model generalization.

4.2 Response generation

To further improve the reality and diversity of the dataset, we use 10 popular LLMs (e.g., GPT-4o) to generate responses (model list shown in Appendix B). During the generation, we use manually constructed diverse instruction templates to guide the LLMs to generate rich answer types.

4.3 QAExtract-dataset

The QAExtract dataset is used for instruction fine-tuning the extraction model of QAEval to generate concise and precise answers from longer responses.

To construct the dataset, simple cases where short answers can be derived through rule-based matching or LLM-generated extraction are first processed preliminarily, followed by manual verification. For more complex QA samples, manual extraction is conducted to ensure accuracy and quality. Three independent annotators (Appendix A.2) with expertise in NLP and QA research participate in this process. In cases where manually extracted answers differ among annotators, discussions are conducted to reach a consensus. The final dataset consists of 9,889 samples, each containing *[Question, Model Response]* as input and *[True Extracted Answer]* as the ground truth output.

To standardize the annotation task, the extraction process adheres to strict guidelines: 1) Human annotators extract short answers by referring to the given context *[Question, Model Response]*. 2) Extracted answers must be strictly derived from the content of *[Model Response]* without adding external information. 3) Extracted answers should be as concise as possible, retaining only numerical values when applicable, with multiple answers separated by semicolons.

4.4 QAScore-dataset

The QAScore dataset is designed to train a scoring model that evaluates the alignment between *[Model Responses]* and *[True Answers]*. It comprises 14,419 samples, each structured as *[Question, True Answer, Model Response]* as input and *[True Score]* as the target output. To ensure high quality and accuracy, all samples are human-scored based on a standardized three-level scale: Complete Match (1.0) for responses identical to the true answer, Partial Match (0.5) for responses with partial correctness but some omissions or biases, and Mismatch or Excess Content (0.0) for entirely incorrect or extraneous responses. A rigorous annotation process is implemented to maintain consistency and reliability. Three expert annotators (Appendix A.2) with backgrounds in NLP and QA evaluation are selected and instructed with clear scoring guidelines and examples to minimize subjectivity. Each sample is independently scored by the three annotators, with discrepancies resolved through discussion to reach a consensus.

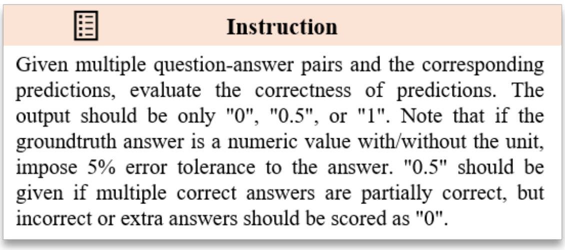
5 Experiment

5.1 Methods

We perform extensive experiments using the rule-based and LLM-based methods. **1) Rule-based:** Accuracy, Relaxed-accuracy (Methani et al., 2020), BEM (Bulian et al., 2022), and PEDANTS (Li et al., 2024). **2) LLM-based:** Qwen2.5 (Team, 2025), Claude-3 (Anthropic, 2024), Gemini-1.5 (Team et al., 2024a), GPT-4o-mini (OpenAI, 2024), ChartX (Xia et al., 2024), Charxiv (Wang et al., 2024), and GPT-4o (Hurst et al., 2024). More details are shown in Appendix C.

5.2 Settings

For the rule-based and open-source LLM-based methods, we use publicly available code and pre-



Instruction

Given multiple question-answer pairs and the corresponding predictions, evaluate the correctness of predictions. The output should be only "0", "0.5", or "1". Note that if the groundtruth answer is a numeric value with/without the unit, impose 5% error tolerance to the answer. "0.5" should be given if multiple correct answers are partially correct, but incorrect or extra answers should be scored as "0".

Figure 4: The instruction for proprietary LLM-based methods.

Name	Variable	Value
Number of trained evaluators	K	8
Number of selected evaluators	Top-k	2
Minimum weight	w_{min}	0.01
Initial weight	w_{initial}	1
Early adjustment factor	f	0.3
Curve steepness factor	j	20
Hidden layer size	d	768
Batch size	N	16

Table 2: Hyper-parameter statistics.

Methods	RT(s)	Cost	Acc.(%)
Human	-	-	**99.1**
Rule-based			
Accuracy	4.17 (O)	Free	47.5
Relaxed-accuracy	4.36 (O)	Free	47.6
BEM	56.34 (O)	Free	63.5
PEDANTS	15.89 (O)	Free	70.2
LLM-based			
ChartX(GPT-3.5)	1281.74 (P)	\$0.5(Q)+\$1.5(A)	64.4
GPT-4o-mini	1886.73 (P)	\$0.15(Q)+\$0.6(A)	84.5
Claude-3	1418.14 (P)	\$0.25(Q)+\$1.25(A)	85.5
Gemini-1.5	1172.81 (P)	\$0.15(Q)+\$0.6(A)	86.1
ChartX(GPT-4o)	1381.60 (P)	\$2.5(Q)+\$10(A)	87.5
Charxiv	3024.51 (P)	\$2.5(Q)+\$10(A)	90.6
Fine-tune			
Qwen2.5-0.5B	49.69 (O)	Free	44.5
Qwen2.5-0.5B(FT)	129.97 (O)	Free	58.1
QAEval (0.6B)	139.62 (O)	Free	**92.3**

Table 3: Comparison of different methods in accuracy (Acc.), running time (RT), and cost. \$0.5(Q) + \$1.5(A) means that input 1 million tokens cost \$0.5 and output (response) 1 million tokens cost \$1.5. O=Open source (RTX4090D for running). P=Proprietary.

trained weights with default hyperparameters. For the proprietary closed-source LLM-based methods, we conduct experiments by calling the API through the official website (More details in Appendix D). As shown in Fig. 4, we use the same instructions to ensure a fair comparison. The QAEval framework comprises an extraction model (Qwen2.5-0.5B[2]), a rule-based model (Algorithm 1), a mixture of evaluators model (ALBERT[3] as evaluator). We train QAEval for 10 epochs with a learning rate of 1e-6. Hyper-parameter statistics are shown in Table 2.

5.3 Evaluation

In our experiments, **"Accuracy"** measures whether the final prediction scores match the true labels, while **"Score"** serves as a complementary metric to assess the gap between each method's total evaluation score and the true total score, which is 51.6% in the test set. Since there is no publicly available test set, we evaluate performance using QAScore-Test with 1,000 samples.

6 Results

We aim to develop an accurate and efficient QA scoring method with minimal computational cost. Therefore, our evaluation focuses on accuracy, running time, and cost-effectiveness. As shown in Table 3, QAEval achieves the highest 92.3% accuracy (with 97.06%/477 in our rule-based model and 87.95%/523 in our MOE model) among all methods, closely approaching human-level performance (99.1%). Unlike LLM-based methods, QAEval does not rely on external API calls, eliminating additional costs. Although its execution time surpasses LLM-based methods with comparable parameter sizes and rule-based methods due to the complexity of its hybrid framework, QAEval exceeds these methods in accuracy by a significant margin.

Rule-based methods like Accuracy and Relaxed-accuracy show high efficiency, completing a single evaluation in approximately 4 seconds without requiring an answer extraction model. However, their accuracy falls below 50%, highlighting their limitations in handling diverse QA tasks using predefined rules alone. The BEM and PEDANTS methods perform better, but there is still a significant gap with the QAEval (-22.1% PEDANTS, and -28.8% BEM). Both models perform significantly worse in the evaluation of complex QA samples and are not flexible enough to adapt to diverse QA samples, especially for samples with very long responses but short answers. LLM-based methods exhibit strong accuracy, with Charxiv achieving 90.6%. However, their high API costs and time constraints limit scalability. Smaller LLMs perform poorly under default settings, as seen with Qwen2.5-0.5B, which

[2] https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct

[3] https://huggingface.co/albert

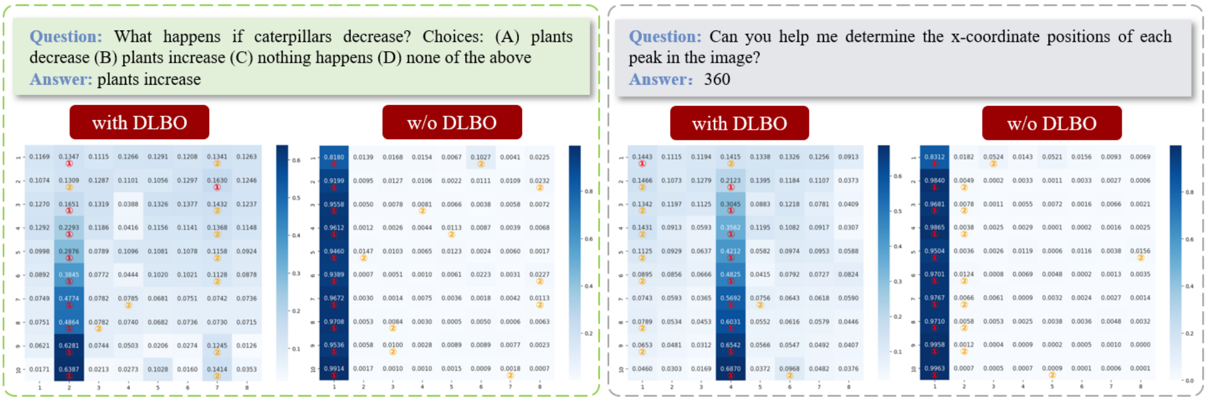


Figure 5: Visualization of the effect of DLBO on the gating network. X-axis denotes the indexes of evaluators; Y-axis denotes the number of training epochs. Red and yellow numbers indicate the top2 selected evaluators.

	Instruction Settings					Ground
	1	2	3	4	5	Truth
Acc.	79.9	85.5	77.1	79.2	79.4	100
Score	47.8	52.6	52.1	50.1	51.9	51.6

Table 4: Claude-3 performance by different instructions.

	Version	Accuracy	Score
GPT-4o	GPT-4o-0513	81.7	48.8
	GPT-4o-0816	87.5	51.2

Table 5: Different GPT-4o version comparison

Methods	RS (%)	ES (%)	Imp. (%)
Accuracy	47.5	79.6	(↑)32.1
Relaxed-accuracy	47.6	87.9	(↑)40.3
ChartX(GPT-3.5)	64.4	67.1	(↑)2.7
Claude-3	85.5	88.1	(↑)2.6
ChartX(GPT-4o)	87.5	90.6	(↑)3.1

Table 6: Effect of key information extraction on scoring results (Acc.). RS denotes using response and true answer to score; ES denotes using extracted answer and true answer to score; Imp. denotes improvements.

attains only 44.5% accuracy. After fine-tuning, its accuracy improves to 58.1%, demonstrating the effectiveness of task-specific fine-tuning for smaller models in the evaluation task.

6.1 LLM-based solution analysis

LLM-based scoring lacks robustness to instruction variations. To assess the robustness and consistency of LLMs in QA scoring, we conduct experiments using the Claude-3 model with five semantically equivalent but differently phrased instructions (provided in Appendix E). As shown in Table 4, the model exhibits notable sensitivity to instruction phrasing, with accuracy varying between 77.1% and 85.5%, a gap of 8.4%. Similarly, the final scores ranged from 47.8% to 52.6%, differing by 4.8%. These results indicate that even minor modifications in instruction wording can significantly impact scoring outcomes, highlighting the lack of robustness in LLM-based scoring methods.

Scoring performance varies across LLM versions. LLM-based evaluation methods often depend on proprietary APIs, and different model versions can yield substantially different results. To examine this effect, we compare two versions of the GPT-4o model (0513 and 0816) on the QA scoring task, as detailed in Table 5. The GPT-4o-0816 version achieves an accuracy of 87.5%, outperforming the 0513 version (81.7%) by 5.8%. These findings underscore the importance of version stability in ensuring consistent evaluation results.

6.2 Extraction and DLBO utility analysis

Answer extraction significantly enhances scoring performance. This experiment evaluates the impact of a paradigm that extracts concise answers from lengthy responses before scoring them against true answers. In Table 6, results demonstrate substantial performance gains across all evaluation methods. For LLM-based methods, extraction improves accuracy by 2–3%, while rule-based methods experience a notable 40.3% increase in Relaxed-accuracy, achieving a final accuracy of 87.9%, comparable to leading LLM-based methods. These findings indicate that integrating an answer extraction model with rule-based method enhances QA scoring accuracy, simplifies processing of lengthy responses, and improves overall reliability, validating the effectiveness of our method.

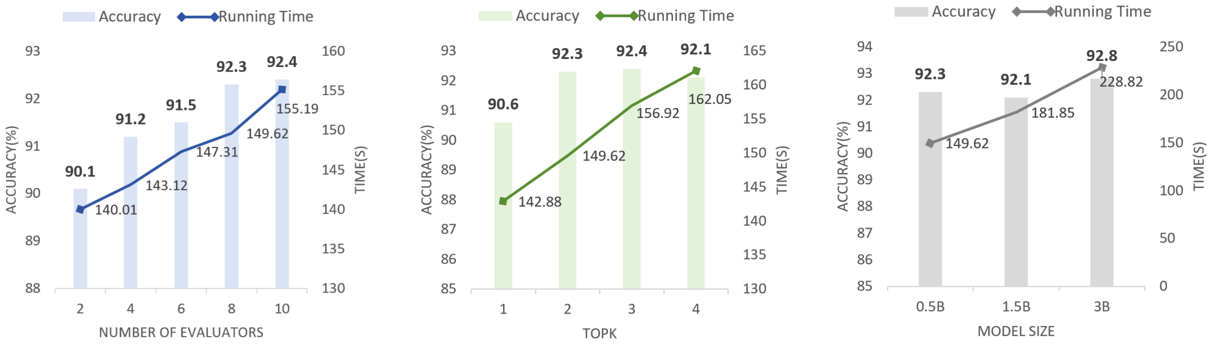


Figure 6: The impact of the number of evaluators, the number of selected evaluators (Top-k) for final decision-making, and different sizes of extraction models on scoring results and runtime.

DLBO enhances specialized knowledge distribution among evaluators. To prevent over-reliance on a single evaluator for all samples, we introduce DLBO, which encourages diverse evaluator specialization. A case study visualizing the gating network's output during different training epochs (8 evaluators) highlights this effect in Fig. 5. DLBO assigns tasks to Evaluators 2 (left) and 4 (right), whereas without DLBO, the model consistently relies on Evaluator 1. This demonstrates that DLBO enables a more balanced and task-specific allocation of evaluators in our MOE architecture, improving the robustness of the evaluation process.

6.3 Hyperparameter analysis

Number of evaluators. Our findings indicate that increasing the number of evaluators improves performance when there are fewer evaluators (see Fig. 6 left). However, beyond 8 evaluators, the performance gains become marginal, while the computational cost increases. This occurs because, with more than 8 evaluators, the gating network faces difficulty in effectively distinguishing the roles of each evaluator, resulting in diminishing returns in performance.

Number of selected evaluators (Top-k). We further examine the impact of the number of evaluators selected by the gating network in generating the final output (see Fig. 6 middle). Selecting only one evaluator results in incorrect predictions for a small subset of complex samples, due to a reduction in robustness. However, due to the similarity and redundancy of the outputs, as more than two evaluators are selected, the accuracy stabilizes, with little improvement beyond this point. Conversely, the computational time increases substantially as the number of evaluators grows, highlighting a trade-off between accuracy and efficiency.

6.4 Model scaling analysis

Given that LLMs often enhance performance through an increase in parameters, we conduct experiments with the Qwen-2.5 model, testing variants with 0.5B, 1.5B, and 3B parameters. The results reveal that while a higher number of parameters yields some performance improvement, the gains are marginal relative to the increase in computational resource requirements, which negatively impacts the efficiency (see Fig. 6 right). These findings suggest that high-quality data annotations, like QAExtract, are more crucial than model size in improving QAEval performance, stressing the value of effective labeling over model complexity.

7 Conclusion

This work tackles the limitations of current QA evaluation methods by introducing QAEval, which effectively combines the reliability of rule-based approaches with the adaptability of LLM-based models. Our approach demonstrates superior performance relative to advanced LLMs, while significantly reducing computational costs. In addition, the QAExtract and QAScore datasets provide training and evaluation resources for QA evaluation.

Limitations

While QAEval solves QA tasks with unambiguous answers effectively, e.g., visual QA, math problem solving, textbook QA, it is limited when applied to tasks involving ambiguous or subjective responses, such as literary dialogues, multi-turn conversations, and descriptive generation. In these tasks, answers are often not unique and may vary depending on context or interpretation. As a result, the quality evaluation of such responses is more prone to subjective influences. Therefore, further works are needed for ambiguous QA tasks.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 1(2):3.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Boerschinger, and Tal Schuster. 2022. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. *arXiv preprint arXiv:2202.07654*.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. 2024. A survey on mixture of experts. *arXiv preprint arXiv:2407.06204*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Cheng-Han Chiang and Hung-yi Lee. 2023. A closer look into automatic evaluation using large language models. *arXiv preprint arXiv:2310.05657*.

Dongyang Fan, Bettina Messmer, and Martin Jaggi. 2024. Towards an empirical understanding of moe design choices. *arXiv preprint arXiv:2402.13089*.

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. 2025. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. *Preprint*, arXiv:2405.14297.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*.

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

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. 2023. Bioasq-qa: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(1):170.

Zhenzhong Lan. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Zongxia Li, Ishani Mondal, Yijun Liang, Huy Nghiem, and Jordan Lee Boyd-Graber. 2024. Pedants: Cheap but effective and interpretable answer equivalence. *arXiv preprint arXiv:2402.11161*.

Qian Liu, Rui Mao, Xiubo Geng, and Erik Cambria. 2023. Semantic matching in machine reading comprehension: An empirical study. *Information Processing and Management*, 60(2):103145.

Jie Ma, Pinghui Wang, Dechen Kong, Zewei Wang, Jun Liu, Hongbin Pei, and Junzhou Zhao. 2024. Robust visual question answering: Datasets, methods, and future challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. 2024. GPTEval: A survey on assessments of ChatGPT and GPT-4. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, page 7844–7866, Torino, Italia. ELRA and ICCL.

Rui Mao, Qian Liu, Kai He, Wei Li, and Erik Cambria. 2023. The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection. *IEEE Transactions on Affective Computing*, 14(3):1743–1753.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1527–1536.

Bolanle Ojokoh and Emmanuel Adebisi. 2018. A review of question answering systems. *Journal of Web Engineering*, 17(8):717–758.

OpenAI. 2024. GPT-4o mini: advancing cost-efficient intelligence.

Shivanshu Shekhar, Tanishq Dubey, Koyel Mukherjee, Apoorv Saxena, Atharv Tyagi, and Nishanth Kotla. 2024. Towards optimizing the costs of llm usage. *arXiv preprint arXiv:2402.01742*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, et al. 2024a. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Qwen Team. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Reka Team, Aitor Ormazabal, Che Zheng, Cyprien de Masson d'Autume, Dani Yogatama, Deyu Fu, Donovan Ong, Eric Chen, Eugenie Lamprecht, Hai Pham, Isaac Ong, Kaloyan Aleksiev, Lei Li, Matthew Henderson, Max Bain, Mikel Artetxe, Nishant Relan, Piotr Padlewski, Qi Liu, Ren Chen, Samuel Phua, Yazheng Yang, Yi Tay, Yuqi Wang, Zhongkai Zhu, and Zhihui Xie. 2024b. Reka core, flash, and edge: A series of powerful multimodal language models. *Preprint*, arXiv:2404.12387.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

Tu Vu, Kalpesh Krishna, Salaheddin Alzubi, Chris Tar, Manaal Faruqui, and Yun-Hsuan Sung. 2024. Foundational autoraters: Taming large language models for better automatic evaluation. *arXiv preprint arXiv:2407.10817*.

Heng Wang, Tan Yue, Xiang Ye, Zihang He, Bohan Li, and Yong Li. 2022. Revisit finetuning strategy for few-shot learning to transfer the emdeddings. In *The Eleventh International Conference on Learning Representations*.

Yifei Wang. 2023. Deciphering the enigma: A deep dive into understanding and interpreting llm outputs. *Authorea Preprints*.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. 2024. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *arXiv preprint arXiv:2406.18521*.

Anuradha Welivita and Pearl Pu. 2023. A survey of consumer health question answering systems. *Ai Magazine*, 44(4):482–507.

Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Min Dou, Botian Shi, Junchi Yan, et al. 2024. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint arXiv:2402.12185*.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023. A critical evaluation of evaluations for long-form question answering. *arXiv preprint arXiv:2305.18201*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. MiniCPM-V: A GPT-4V level MLLM on your phone. *arXiv preprint 2408.01800*.

Tan Yue, Yong Li, and Zonghai Hu. 2021. Dwsa: An intelligent document structural analysis model for information extraction and data mining. *Electronics*, 10(19):2443.

Tan Yue, Rui Mao, Heng Wang, Zonghai Hu, and Erik Cambria. 2023. KnowleNet: Knowledge fusion network for multimodal sarcasm detection. *Information Fusion*, 100:101921.

Tan Yue, Xuzhao Shi, Rui Mao, Zonghai Hu, and Erik Cambria. 2024. Sarcnet: a multilingual multimodal sarcasm detection dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14325–14335.

Munazza Zaib, Wei Emma Zhang, Quan Z Sheng, Adnan Mahmood, and Yang Zhang. 2022. Conversational question answering: A survey. *Knowledge and Information Systems*, 64(12):3151–3195.

Luyao Zhu, Rui Mao, Erik Cambria, and Bernard J. Jansen. 2024. Neurosymbolic AI for personalized sentiment analysis. In *Proceedings of International Conference on Human-Computer Interaction (HCII)*, pages 269–290.

A Details of datasets

A.1 Collection

As shown in Table A.1, to efficiently train our extraction and scoring models, we build two high-quality manually labeled datasets: QAExtract and QAScore, which are sampled from 31 widely-used QA-related datasets covering a wide range of domains, including textbook question answering, visual question answering, figure question answering, etc (shown in Table A.2). We show the sample in QAScore dataset in Fig. A.1.

Dataset	Que.	Res.	Ext.A	Tru. A
QAExtract	19.71	37.05	1.65	-
QAScore	18.16	39.13	-	1.61

Table A.1: The average token statistics.

A.2 Annotation

A total of 6 annotators participate in the dataset annotation (3 for QAExtract dataset and 3 for QAScore dataset). The annotators, all native Chinese speakers from top universities, possessed advanced English proficiency (CET-6). The annotators have expert experience in the field of NLP and QA, and we make the annotation rules for both QAExtract and QAScore datasets as detailed instructions for annotators. The average hourly payment for each annotator is 129 CNY, exceeding the minimum wage. Additionally, annotators are given a 10-minute break after 30 minutes of work.

The dataset and code will be open-sourced under the MIT License. The data used in this study are derived from publicly available datasets and it has been ensured that the relevant data use policies and privacy protections have been followed. The data will provide support to the academic field and the use has been explained in detail in the instructions.

Dataset	Year	Task
GEOS	2015	Geometry Problem
VQA-AS	2015	Visual Question Answering
AI2D	2016	Textbook Question Answering
FigureQA	2017	Figure Question Answering
TQA	2017	Textbook Question Answering
VQA2.0	2017	Visual Question Answering
DVQA	2018	Figure Question Answering
VizWiz	2018	Visual Question Answering
VQA-RAD	2018	Visual Question Answering
KVQA	2019	Visual Question Answering
TextVQA	2019	Visual Question Answering
PlotQA	2020	Figure Question Answering
Geometry3K	2021	Geometry Problem
IconQA	2021	Math Problem
GeoQA+	2022	Geometry Problem
UniGeo	2022	Geometry Problem
CLEVR-Math	2022	Math Problem
ChartQA	2022	Figure Question Answering
MapQA	2022	Figure Question Answering
DocVQA	2022	Figure Question Answering
ScienceQA	2022	Textbook Question Answering
A-OKVQA	2022	Visual Question Answering
ParsVQA-Caps	2022	Visual Question Answering
TabMWP	2023	Math Problem
SciBench	2023	Textbook Question Answering
TheoremQA	2023	Textbook Question Answering
PMC-VQA	2023	Visual Question Answering
Super-CLEVR	2023	Visual Question Answering
SciChart	2024	Figure Question Answering
Charvix	2024	Figure Question Answering
MathVista	2024	Figure Question Answering

Table A.2: Datasets used for developing QAExtract and QAScore.

B Details of the models used to generate responses

Claude-3.5-Sonnet (Anthropic, 2024) model performs well in reasoning, knowledge acquisition, and coding skills. The model has strong visual processing capabilities and is able to interpret charts and graphs.

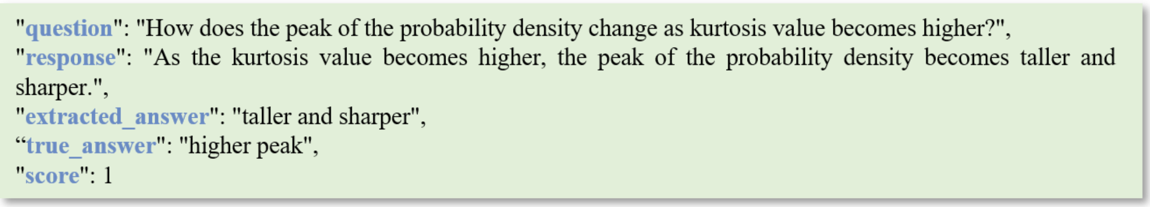
Gemini-1.5-Pro (Team et al., 2024a) is a multimodal model from Google with a sparse mixture of experts (MoE) architecture with a contextual window for processing up to millions of tokens for complex tasks requiring advanced reasoning and analysis.

Qwen-VL-Plus (Bai et al., 2023) is an enhanced version of the visual language model introduced by Alibaba, which significantly improves the recognition of image details and text, and excels in a wide range of visual tasks.

Qwen-VL-Max (Bai et al., 2023) is an ultra-large-scale visual language model from Alibaba that excels in visual reasoning and command following capabilities, supports processing of ultra-megapixel high-resolution images, and is capable of accurately recognizing image details and text.

Phi-3-Vision (Abdin et al., 2024) is a lightweight, multimodal state-of-the-art model supporting 128K context length, trained on high-quality textual and visual inference data, and rigorously supervised fine-tuned and preference-optimized for excellent command adherence and security.

InternVL2-76B (Chen et al., 2024) is a multimodal



"question": "How does the peak of the probability density change as kurtosis value becomes higher?",
"response": "As the kurtosis value becomes higher, the peak of the probability density becomes taller and sharper.",
"extracted_answer": "taller and sharper",
“true_answer": "higher peak",
"score": 1

Figure A.1: The sample in QAScore dataset.

large language model with capabilities of document and chart comprehension, infographics QA, scene text understanding, etc.

MiniCPM-V2.5 (Yao et al., 2024) is the latest multimodal large model of MiniCPM-V series, supporting more than 30 languages, with excellent document understanding, OCR extraction, complex reasoning, etc., while optimized for efficient edge device deployment.

Reka-Core (Team et al., 2024b) is a 67 billion-parameter multimodal language model supporting 32 languages with the ability to process text, images, video and audio for complex application scenarios and model distillation.

GPT-4V-Turbo (Achiam et al., 2023) is a multimodal model from OpenAI that combines natural language processing and visual understanding capabilities to analyze images and answer related questions. The model offers a significant increase in processing speed over GPT-4.

GPT-4o (Achiam et al., 2023) is a SOTA multimodal model that can process multimodal inputs (e.g., text and images) as well as perform excellent understanding and reasoning.

C Methods

C.1 Rule-based methods

Accuracy is a metric that measures the consistency between predicted and target answers, calculated as the ratio of correctly predicted samples to the total number of samples. The formula is:

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_{\text{total}}}, \tag{8}$$

where N_{correct} denotes the number of samples where the predicted answer matches the target answer exactly, and N_{total} represents the total number of samples.

Relaxed-accuracy provides an additional tolerance for errors in numerical answers. For example, with a tolerance of 5%, 102 (Predict answer) = 100 (True answer).

$$\text{Relax-accuracy} = \frac{N_{\text{relaxed-correct}}}{N_{\text{total}}}, \tag{9}$$

where $N_{\text{relaxed-correct}}$ denotes the number of samples where the predicted numerical answer $\hat{y}$ satisfies:

$$|\hat{y} - y| \leq \epsilon \cdot y, \tag{10}$$

with y being the true answer and ϵ the allowed tolerance fraction (e.g., $\epsilon = 0.05$ for 5%). N_{total} represents the total number of samples.

BEM (Bulian et al., 2022) introduces a rule-based evaluation method that maps answers into predefined equivalence classes based on transformations such as numerical formatting, unit conversions, and common paraphrases, enabling more flexible matching beyond token-level accuracy.

PEDANTS (Li et al., 2024) proposes a lightweight rule-based system that performs answer equivalence checking through manually designed transformations and pattern matching, aiming to achieve interpretable and cost-efficient evaluation.

C.2 LLM-based methods

Qwen2.5-0.5B (Team, 2025) is the base model of the Qwen2.5 family with 500 million parameters and a decoder-only architecture. The model is pre-trained on large-scale multilingual and multimodal datasets with multilingual support and structured data comprehension.

Claude-3-Haiku (Anthropic, 2024) model has multimodal capabilities and is able to fast and efficiently process visual and textual tasks.

Gemini-1.5-Flash (Team et al., 2024a) is an efficient multimodal pre-training model introduced by Google that aims to improve the performance of visual and language understanding tasks through fast reasoning and low computational cost.

GPT-4o-mini (OpenAI, 2024) is a lightweight version of the GPT-4o model, designed to provide multimodal capabilities in a more efficient model. It maintains a better ability to process multimodal data and greatly reduces the model size.

GPT-4o (Achiam et al., 2023) is a SOTA multimodal model that can process multimodal inputs (e.g., text and images) as well as perform excellent understanding and reasoning.

ChartX (Xia et al., 2024) designs the GPT-acc evaluation method for tasks with clear answers such as QA, which uses the GPT-4 model to evaluate responses and true answers and outputs binary classification results (True/False). Similar to relaxed-accuracy, GPT-acc sets a 5% error margin for numerical answers.

Charxiv (Wang et al., 2024) uses the GPT-4o model to evaluate model responses and true answers, designed with instructions that first ask the model to extract short answers from long responses and then evaluate them with true answers.

D Experimental settings

In this study, for different types of methods, we have given careful consideration to the experimental setup to ensure that each method is comparable under a fair evaluation framework.

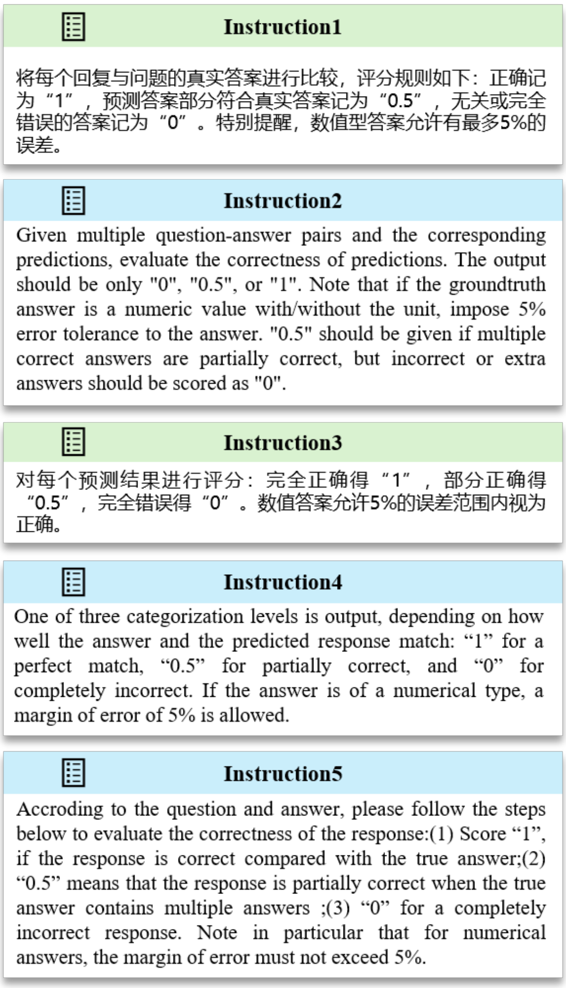
Rule-based methods: based on the formulas in Appendix C.1, we implement the Accuracy and Relaxed-accuracy assessment methods. We set a 5% tolerance for relaxed-accuracy.

Proprietary LLM-based methods: for the experiments of proprietary LLM-based methods, including Claude-3, Gemini-1.5, GPT-4o-mini and GPT-4o, we conduct experiments by calling the API through the official website. The official recommended hyperparameter settings are strictly followed for each model. ChartX[4] and Charxiv[5] are based on GPT-4 and GPT-4o, respectively. We conducted experiments using the code, instructions and parameters provided in their papers.

Open-source LLM-based methods: for Qwen2.5-0.5B, we use publicly available code and pre-trained weights (downloaded from Huggingface[6]) with default hyperparameters. And for Qwen 2.5-0.5B(FT), we use the LoRA method (Hu et al., 2021) to fine-tune the model on the QAScore training dataset with 10 epochs.

E Instructions for robustness testing

Given that current LLM-based methods typically exhibit high sensitivity to instruction input, i.e., instructions with different expressions may lead to significantly different evaluation results, we design and implement a multi-instruction experiment. Specifically, we design five semantically consistent instructions that differ in terms of expression structure, wording choice, and linguistic style to systematically evaluate the stability and robustness of the model under different instruction expressions.



Instruction1

将每个回复与问题的真实答案进行比较，评分规则如下：正确记为“1”，预测答案部分符合真实答案记为“0.5”，无关或完全错误的答案记为“0”。特别提醒，数值型答案允许有最多5%的误差。

Instruction2

Given multiple question-answer pairs and the corresponding predictions, evaluate the correctness of predictions. The output should be only "0", "0.5", or "1". Note that if the groundtruth answer is a numeric value with/without the unit, impose 5% error tolerance to the answer. "0.5" should be given if multiple correct answers are partially correct, but incorrect or extra answers should be scored as "0".

Instruction3

对每个预测结果进行评分：完全正确得“1”，部分正确得“0.5”，完全错误得“0”。数值答案允许5%的误差范围内视为正确。

Instruction4

One of three categorization levels is output, depending on how well the answer and the predicted response match: “1” for a perfect match, “0.5” for partially correct, and “0” for completely incorrect. If the answer is of a numerical type, a margin of error of 5% is allowed.

Instruction5

Accroding to the question and answer, please follow the steps below to evaluate the correctness of the response:(1) Score “1”, if the response is correct compared with the true answer;(2) “0.5” means that the response is partially correct when the true answer contains multiple answers ;(3) “0” for a completely incorrect response. Note in particular that for numerical answers, the margin of error must not exceed 5%.

Figure A.2: Different instructions for robustness testing.

These instructions cover different linguistic features, including concise, lengthy, and language, etc., so as to simulate as much as possible the diversified forms of instruction inputs that the model may face in real application scenarios. Examples of the instructions are shown in Fig. A.2, and the model outputs under different instructions are compared through multiple rounds of experiments to further analyze their impact on the model performance and potential adaptive capability.

[4] https://github.com/Alpha-Innovator/ChartVLM

[5] https://github.com/princeton-nlp/CharXiv

[6] https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct