Automatic Large Language Model Evaluation via Peer Review

Anonymous ACL submission

Abstract

The impressive performance of large language models (LLMs) has attracted considerable attention from the academic and industrial communities. Besides how to construct and train LLMs, how to effectively evaluate and compare the capacity of LLMs has also been well recognized as an important yet difficult problem. Existing paradigms rely on either human annotators or model-based evaluators to evaluate the performance of LLMs on different tasks. However, these paradigms often suffer from high cost, low generalizability, and inherited biases in practice, which make them incapable of supporting the sustainable development of LLMs in long term. In order to address these issues, inspired by the peer review systems widely used in academic publication process, we propose a novel framework that can automatically evaluate LLMs through a peer-review process. Specifically, for the evaluation of a specific task, we first construct a small qualification exam to select "reviewers" from a couple of powerful LLMs. Then, to actually evaluate the "submissions" written by different candidate LLMs, i.e., the evaluatees, we use the reviewer LLMs to rate or compare the submissions. The final ranking of evaluatee LLMs is generated based on the results provided by all reviewers. We conducted extensive experiments on both text summarization and non-factoid question answering tasks with eleven LLMs including GPT-4. The results demonstrate the existence of biasness when evaluating using a single LLM. Also, our PRE model outperforms all the baselines, illustrating the effectiveness of the peer review mechanism.

1 Introduction

The continuous development of large-scale language models (LLMs) such as GPT-3 (Brown et al., 2020), PALM (Chowdhery et al., 2022), and Llama (Touvron et al., 2023a) has sparked people's passion on Artifical General Intelligence in both academia and industry. A new generation of LLMs, led by GPT-4 (OpenAI, 2023) and Claude (Anthropic, 2023b,a), can achieve competitive performance on a wide range of natural language processing tasks, even in zero-shot scenarios. Ever since the release of ChatGPT (OpenAI, 2022), a large number of LLMs have been developed, many of which can produce high-quality responses that achieve or even surpass human-level performance in many cases (Mao et al., 2023; Ali et al., 2022).

With rapid development of LLMs, how to evaluate the performance of LLMs both effectively and efficiently has become a crucial bottleneck that restricts LLMs' progress. A reliable and reusable LLM evaluation method not only helps us better select the best LLMs for each task, but also provides important guidelines for LLM optimization.

To the best of our knowledge, there are two types of evaluation paradigms for LLM: human evaluation and model-based evaluation. The former hires human annotators to judge the quality of responses generated by LLMs directly or create gold references to evaluate the outputs of LLMs. The later trains separate evaluators for each task or uses a powerful LLM (e.g., GPT-4) to evaluate the performance of other LLMs. Unfortunately, due to their intrinsic characteristics, these methods often suffer from one or more of the following three problems:

(1) **High cost**: Human annotations have been considered the most effective and reliable data to evaluate the quality of LLM outputs (Zheng et al., 2023; Frieder et al., 2023; Jang et al., 2022). However, in commonly-used generation tasks, such as text summarization and question answering, different LLMs would output diverse responses, leading the cost of evaluation be approximately proportional to the number of evaluated LLMs. Reference-based methods (Lin, 2004; Papineni et al., 2002; Zhang et al., 2019) try to avoid this problem by requiring the annotators to provide gold references for each tasks instead of judging the quality of each

LLM's outputs directly, but this could significantly increase the load and difficulty of the annotation jobs. Also, since LLMs are extremely powerful in terms of memorization, any public reference-based datasets can easily be incorporated and optimized by LLMs in the training process and thus become useless for evaluation after a short period of time. All these make the cost of annotation-based LLM evaluation methods prohibitive in long term.

(2) **Low generalizibility**: Existing evaluation methods, such as reference-based or model-based evaluators, often requires task-specific dataset construction and evaluator pre-training (Xu et al., 2023; Sun et al., 2023; Kryściński et al., 2019; Hendrycks et al., 2020; Gu et al., 2023). For example, Xu et al. (Xu et al., 2023) designed multiple-choice questions based on human references to evaluate LLMs. Similarly, studies like (Sun et al., 2023) fine-tune pretrained language models on each specific task with large-scale supervised data to create model-based evaluators. However, the evaluators created in these methods cannot be generalized to tasks beyond the target task of the references or the training data. Considering the large number and variety of LLM applications, the low generalizability of these evaluation methods make them not preferable for LLM evaluation.

(3) **Inherent bias**: Due to their intrinsic model structure or algorithm design, many evaluation methods are inherently biased in the evaluation process. For example, reference-based word similarity metrics (such as ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), and BERTScore (Zhang et al., 2019)), which are commonly used to evaluate the outputs of LLMs in generation tasks, steer LLM outputs to be as similar as possible to the reference text, discriminating against LLMs that create qualified but different responses. Recently, many studies have adopted the state-of-the-art LLM, GPT-4, as their evaluation tools (Liu et al., 2023; Kocmi and Federmann, 2023). Although several works have demonstrated that GPT-4 has decent evaluation capabilities (Liu et al., 2023; Kocmi and Federmann, 2023), we found that GPT-4, so as other LLMs, often prefers responses of LLMs from its own series (i.e., the GPT models) over other LLMs despite of the actual quality of the responses. In other words, if we use GPT-4 as the evaluator, its inherent bias may make it difficult, if possible, to develop an LLM outside the GPT family that outperforms GPT-4.

To address the aforementioned issues, we propose a novel framework, Peer Review Evaluator (PRE) [1], to evaluate the performance of LLMs automatically. Inspired by the peer review mechanism in academic community, we propose to use LLMs as reviewers to evaluate the performance of LLMs directly. Specifically, we first develop a qualification exam to filter out LLMs that fail to provide reliable evaluation results. Then, qualified reviewer LLMs are required to assess the outputs of the evaluatee LLMs, and the final evaluation results are aggregated from all reviewer LLMs' ratings or preferences. To verified the effectiveness of our framework, we conducted extensive experiments on two representative text generation tasks, i.e., document summarization and non-factoid question answering. The experimental results show that the results of PRE model have the highest consistency with human preferences (ground truth) compared to all the baseline models including GPT-4. Comparing to previous evaluation methods, PRE can easily be generalized to different tasks and is highly cost efficient. Also, experiment results show that PRE provides much more robust evaluation results than methods that rely on specific model structures or LLMs.

2 Related Work

2.1 Large Language Models

Large Language Models (LLMs) typically refer to language models that contain more than a hundred billion parameters and have been pre-trained on large amounts of textual data. The large-scale parameters and large amounts of training data of LLMs bring impressive capabilities, such as few-shot and zero-shot learning, where they can generate high-quality and reasonable text output with limited prompts.

According to open source or not, LLMs can be divided into two categories: closed source LLMs and open source LLMs. Closed source LLMs include ChatGPT (OpenAI, 2022), GPT-4 (OpenAI, 2023), Claude (Anthropic, 2023b), Claude 2 (Anthropic, 2023a) and Gemini (Team et al., 2023) which only offer API services instead of publicly available models. They tend to have enormous parameter sizes, and therefore reach top performance on all types of tasks.

For open source models, the most renowned model is LLaMa (Touvron et al., 2023a,b) and its derivative models, e.g., Alpaca (Taori et al.,

[1] https://anonymous.4open.science/r/PRE-D66A

2023), Koala (Geng et al., 2023), and Vicuna (Chiang et al., 2023). ChatGLM series (Zeng et al., 2022), in addition to instruction tuning, are committed to utilizing quantitative methods to reduce model memory footprint and improve inference efficiency. Other popular LLMs, such as FastChat-T5 (LMSYS, 2023), Baichuan (Technology, 2023), RWKV (Peng et al., 2023) also attract much attention.

2.2 Evaluation of Large Language Models

With the rapid development of LLMs, how to effectively evaluate the quality of the LLM generative texts has also become an urgent research question. We can simply categorize the existing evaluation approaches into the following categories:

Human Annotations: Human annotation has usually been regarded the most effective and reliable means for evaluating the outputs of LLMs. Recently, LMSYS has built a benchmark platform, Chatbot Arena (Zheng et al., 2023; Frieder et al., 2023; Jang et al., 2022), which allows different LLMs to engage in a fair, anonymous and random battle through crowdsourcing manner. Then, they adopted the ELO rating system to aggregate for the final leaderboard. However, as the number of evaluatee LLMs and evaluation tasks sharply increase, human annotation becomes increasingly unsustainable. Thus, effective semi-automatic or automatic evaluation methods become an urgent need.

Reference-based Word Similarity Metrics: Before the emergence of LLMs, there are a number of similarity metrics that could assess the quality of the generative text based on the reference text. As two series of widely used word similarity metrics, BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) determine the similarity between reference text and generative text by counting the matching units like n-gram. BERTScore (Zhang et al., 2019) is an embedding-based metric that maps texts to embedding vectors and evaluates via cosine similarity.

Evaluation with Multi-Choice Questions: Multiple-choice questions, as a category of questions with fixed output formats, whose easy-evaluation properties lead a great deal of work (Xu et al., 2023; Zhang et al., 2023) to construct benchmarks with such formatting. Such evaluation results are often more intuitive and aligned with human values.

Evaluation using LLMs: Due to the stunning performance of LLMs, many studies attempted to employ one or multiple LLMs as evaluators for the evaluation of LLMs' outputs. GPTScore(Fu et al., 2023) evaluates the quality of generative texts using the generation probability of LLMs. PandaLM (Wang et al., 2023) trains LLaMa (Touvron et al., 2023a) to evaluate the results generated by LLMs through instruction tuning. Its training data is generated by ChatGPT via self-instruct (Wang et al., 2022). PRD (Li et al., 2023) and CHAT-EVAL (Chan et al., 2023), the two recent works, attempt to integrate multiple LLMs into the evaluation system to provide an aligned evaluation result by ranking, discussing, and debating among LLMs.

3 LLM Peer Review Evaluation

In this section, we provide a detailed introduction to the design motivation and specifics of our proposed LLMs evaluation framework.

3.1 Motivation

As discussed in Sec 1, a good LLM evaluation system should be affordable, generalizable and unbiased. Existing evaluation methods powered by a strong LLM (e.g., GPT-4) have been shown to be both effective and cost-efficient (Mao et al., 2023; Ali et al., 2022), but they also suffer from intrinsic limitations like inherent bias as discussed in Section 1. To this end, we propose to employ the peer review mechanism to integrate the evaluation results of multiple LLMs.

Peer review mechanisms are widely used in the academic field for paper reviewing. Journal editors or conference chairs invite experienced researchers in such research fields to act as reviewers, providing feedback and ratings on submitted papers. Editors or chairs take into account the reviewers' comments to make final decisions. Inspired by this mechanism, we apply it to the scenario of LLMs evaluation. Specifically, we consider multiple LLMs as potential reviewers. The evaluation framework, acting as the chair, selects qualified LLM reviewers to rate the outputs of each LLM on the task, and ultimately aggregates the reviewers' rates to provide the final evaluation results.

3.2 Framework Architecture

Figure 1 shows the architecture of our overall LLM evaluation framework: Peer Review Evaluator (PRE). The whole process can be divided into three modules: (1) Qualification exam module: conducting a qualification exam for all reviewer candidate LLMs to select qualified LLMs exceeding a certain level of evaluation capability; (2) Peer

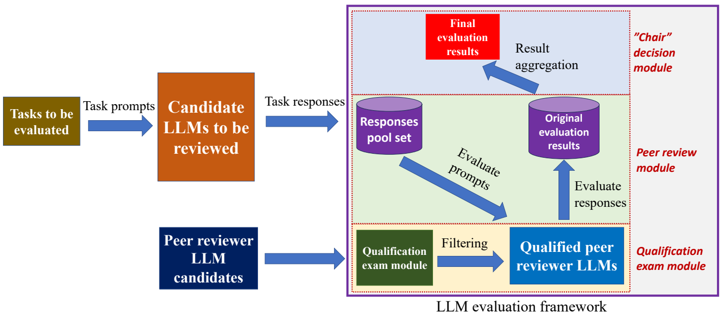


Figure 1: The architecture of our evaluation framework for large language models

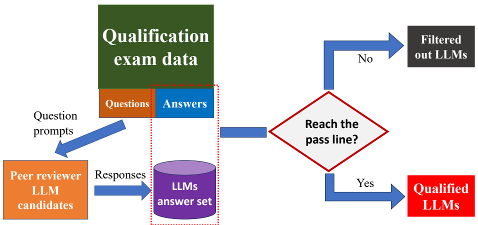


Figure 2: The process of the qualification exam module in our evaluation framework

review module: collecting the outputs of evaluatee LLMs on the given assessment tasks, and then rating the outputs of all evaluatee LLMs by qualified reviewer LLMs; (3) "Chair" decision module: aggregating the ratings provided by all reviewer LLMs to obtain the final evaluation results. Below, we will provide detailed information regarding the design details of each module.

3.2.1 Qualification exam module

Previous work has already demonstrated that LLMs have certain evaluation capabilities (Mao et al., 2023; Ali et al., 2022). Based on this finding, in our framework, any LLMs are allowed to participate in the evaluation process as reviewer candidates. Through the qualification examination module, we select LLMs whose evaluation capability is strong enough from the reviewer candidates to participate as reviewers in the peer review stage. Figure 2 illustrates the specific process of the qualification exam. This process relies on a set of qualification examination data, which should include a set of test cases to assess the evaluation capability of LLMs. We require each reviewer candidate to complete these evaluation tasks, and then compare candidates' outputs with the standard answers to obtain the evaluation scores for each candidate's capability. Only when the evaluation score of a reviewer candidate LLM reaches the admission threshold, will we add it to the reviewer pool.

There are two points worth further discussion regarding the qualification exam data: (1) Data acquisition: In order to closely approximate the application scenarios of reviewer LLMs, in our experimental setup, we used the outputs of a subset of LLMs in the evaluated task as the evaluation objects, constructing the qualification exam data. For simplicity, we use human annotations to create the qualification exam data (described in Sec 4.5), but please note that other unsupervised or semi-supervised methods (Yue et al., 2023; Kryściński et al., 2019) could also be used to create the exam. (2) Data reusability: The purpose of the qualification exam data is to assess the evaluation abilities of reviewer candidate LLMs. With proper design, a single set of qualification exam data can reflect the general evaluation abilities of reviewer candidate LLMs, thus making the reviewer selection results generalizable to multiple tasks. Also, the exam data is designed to evaluate LLM's ability as a reviewer. After the exam data are published, even if an LLM manages to trick the exam and become a reviewer by using the data in training, it doesn't mean that the LLM could stand out as an evaluatee in the actual testing stage (i.e., the peer reviewing process). This makes the whole framework more robust and reusable.

3.2.2 Peer review module

In this module, we first collect the responses of all evaluatee LLMs to the given tasks. Then, each qualified reviewer LLM is required to rate the outputs in the response pool set. Specifically, organizers need to design prompts in advance for rating, and feed them into the reviewer LLMs. Then, they extract corresponding rating information from the reviewers' outputs. It is worth noting that the rating method here is not limited to pointwise evaluation of each (task, response) pair, but can also be in pairwise or even listwise format.

3.2.3 "Chair" decision module

After collecting the comments (ratings) from all the LLM reviewers, the "chair" (evaluation system) needs to aggregate the ratings to generate the final evaluation results. Specifically, we adopt a weighted voting strategy for rating aggregation, as shown in Eq 1.

$$R_x = \frac{1}{W} \sum_{l \in L} w_l r_x^{(l)} \quad (1)$$

In Eq 1, L denotes the whole reviewer LLM set, and $r_x^{(l)}$ denotes the LLM l's rating on sample x. The vote weight w_l of each reviewer LLM is determined by its performance in the qualification exam, while W is the normalization term with $W = \sum_{l \in L} w_l$.

4 Experimental Setup

In this section, we provide a quick introduction to the experimental setup, more details could be found in Appendix.

4.1 Tasks and LLMs Selection

Given the limitations of multiple-choice questions, we chose two representative generation tasks that are more generalizable and more closely matched to real-world needs: text summarization and non-factoid question answering (QA).

As for the text summarization task, we adopted Extreme Summarization (XSum) (Narayan et al., 2018) dataset to construct evaluation tasks. XSum is a real-world single-document news summary dataset collected from online articles by the British Broadcasting Corporation (BBC) and has been widely used in previous research (Chowdhery et al., 2022; Li and Liang, 2021). The entire XSum dataset contains over 220 thousand news documents.

As for the non-factoid QA task, we used the NF-CATS dataset (Bolotova et al., 2022) to create evaluation tasks. NF-CATS is an emerging non-factoid QA dataset that contains 11,984 non-factoid questions as well as their categorizations. We removed the questions belonging to the types “FACTOID” and “NOT-A-QUESTION” to construct the sample pooling set.

To validate the effectiveness of each evaluation method, we need to collect the most reliable evaluation data, i.e., human preferences over the LLM's outputs for each test case, as our ground truth. Due to our limited budget, we randomly sampled 100 samples from the XSum and NF-CATS datasets and used them as our testbed.

We selected eleven powerful LLMs to conduct experiments, including LLMs in both closed-source (e.g. GPT-4 and Claude-1) and open-source (e.g. Llama-2-70b-chat and RWKV-4-Raven-7B) settings. In our experiments, these LLMs play dual roles as both evaluatees and reviewer candidates. Table 1 shows some basic information about these LLMs, as well as their ratings and rankings in the ELO leaderboard (i.e., a leaderboard of LLMs created based on human annotations) of LMSYS released in September 2023 (Zheng et al., 2023). GPT-4 and Claude-1, recognized as two of the strongest existing LLMs, are ranked in the top two positions on the ELO leaderboard.

4.2 Baselines

We compare the performance of the PRE model with several baselines, including:

ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), and BERTScore (Zhang et al., 2019): They are all reference-based word similarity metrics

PandaLM (Wang et al., 2023): A fine-tuned language model based on Llama-7b (Touvron et al., 2023a) for the preference judgment tasks

GPTScore (Fu et al., 2023): Evaluate the quality of generative text based on its generation probability feeding into particular LLMs right after the given prompts

Single LLM: Only use a single LLM as an evaluator to assess the quality of the generative text. Its prompt setting is the same as the PRE model. The LLMs selected here are listed in Table 1

4.3 Meta-evaluation Metrics

In our experiments, we collected manual annotations as the gold standard for the quality of LLM-generated summaries to evaluate the evaluation performance of the PRE model and baselines. Our annotation data includes two parts: pointwise labels as well as auxiliary pairwise preferences.

For these two different formats of labels, we proposed various evaluation metrics to measure the performance of LLM evaluation models. Specifically, (1) for pairwise labels, we use **Precision** (P) to measure the proportion of identical preference results between the model and human cognition. (2) for pointwise labels, we use **Kendall's tau** (τ) (Kendall, 1938) and **Spearman correlation coefficient** (S) (Lehman et al., 2013) to measure the consistency between the model's outputs $\hat{y}(s,t)$ and labels $y(s,t)$.

4.4 Framework Details

Due to the space limit, here we provide a coarse introduction to the implementation of the framework. More details can be found in Appendix A.

Table 1: The basic information of the large language models used in our experiments

Model	Developer	Size (B)	ELO rate (rank)	Evaluatee	Evaluator candidate	Annotation	Exam provider
GPT-4 (OpenAI, 2023)	Openai	/	1193 (1 / 28)	✓	✓		
Claude-1 (Anthropic, 2023b)	Anthropic	/	1161 (2 / 28)	✓	✓	✓	
GPT-3.5-turbo (OpenAI, 2022)	Openai	/	1118 (5 / 28)	✓	✓	✓	✓
Llama-2-70b-chat (Touvron et al., 2023b)	Meta	70	1060 (7 / 28)	✓	✓		
Vicuna-7b (Chiang et al., 2023)	LMSYS	7	1003 (14 / 28)	✓	✓	✓	
ChatGLM2-6B (Zeng et al., 2022)	Tsinghua	6	965 (18 / 28)	✓	✓	✓	
RWKV-4-Raven-7B (Peng et al., 2023)	BlinkDL	7	14B: 939 (21 / 28)	✓	✓	✓	
Alpaca-7b (Taori et al., 2023)	Stanford	7	13B: 919 (22 / 28)	✓	✓	✓	✓
FastChat-t5-3b (LMSYS, 2023)	LMSYS	3	888 (25 / 28)	✓	✓	✓	✓
ChatGLM-Pro (Zeng et al., 2022)	Tsinghua	/	N/A	✓	✓		
Baichuan-2-13b (Yang et al., 2023)	Baichuan Inc.	13	N/A	✓	✓		

We designed three rating methods: **5-level pointwise**, **100-level pointwise**, and **pairwise** (or called **preference**). In the qualification exam module, we select three evaluatee LLMs (GPT-3.5-turbo, Fastchat-t5-3b, and Alpaca-7b) as "questioners" to test the ability of reviewer candidate LLMs. Only when an LLM's Precision exceeds the threshold of ξ, it will be retained as a reviewer for the peer review process. In our experiments, we set ξ to be 60%.

In the peer review module, each reviewer LLM is required to rate all the pointwise (or pairwise) samples. We designed the same prompt templates, and fed them into all the reviewer LLMs. Totally, each reviewer LLM is required to generate $11 \times 100 = 1,100$ rates or $Perm(11, 2) \times 100 = 11,000$ preferences, respectively.

In the "chair" decision module, we adopted the weighted voting strategy. Specially, we set $w_l = \log\left(\frac{p_l}{1-p_l}\right)$, where p_l is the Precision of the LLM l in qualification exam.

4.5 Manual Annotations

We conducted manual annotations serving for two purposes: (1) as ground truth for the LLM qualification exam (only use a small subset of annotations); (2) as a gold standard for evaluating the performance of different evaluation methods. Due to cost considerations, we conducted annotations on 7 out of the 11 LLMs that diversify in quality, developers, and model structure, as shown in Table 1.

To meet the requirements of both pointwise and pairwise evaluation metrics, we conducted pointwise annotation as well as auxiliary preference annotation [2]. Both two types of annotations were conducted under the guidance of Likert scale (Jebb et al., 2021). The difference is the scale: 5-level for pointwise while 7-level for pairwise. We recruited annotators through Amazon's MTurk Crowdsourcing platform [3], assigning 5 different annotators for each Human Intelligence Task (HIT). Overall, we collected 1,400 pointwise HITs and 1,704 preference HITs, collecting a total of 15,520 annotations at a cost of approximately 1,600 dollars. After removing the maximum and minimum labels, the annotations achieve fair annotation agreement: the mean intra-task Krippendorff's α (Krippendorff, 2011) for pointwise and preference annotations are 0.4581 and 0.2983, respectively.

5 Results and Analysis

In this section, we present the experimental results and attempt to answer the following three research questions (RQs): (1) How does the performance of our proposed PRE model compare to other baseline methods? (2) Does the inherit bias really exist when evaluating with a single LLM? (3) How robust is the PRE model in evaluating LLMs?

5.1 Overall Results (RQ1)

Table 2 presents the overall experimental results, evaluated by Precision metric [4], where *PRE w/o GPT-4* represents the PRE variant model that excludes GPT-4 (currently recognized as the strongest LLM) out of the reviewer list. The results show that our proposed PRE model outperforms all the baselines including GPT-4. Even the variant without the GPT-4 model (*PRE w/o GPT-4*) achieves comparable evaluation results with GPT-4. This indicates that the peer review mechanism could effectively evaluate.

Experimental results show that GPT-4 performs the best among all LLMs in terms of evaluating the outputs of evaluatee LLMs. GPT-3.5-turbo, Claude-1 and ChatGLM-Pro also perform well in the evaluation task, as we expect. Surprisingly, FastChat-t5-3b, a model with only 3 billion pa-

[2]Due to the high cost, we conducted preference annotation only on the pairs with a tied pointwise label.

[3]https://www.mturk.com/

[4]Due to the space limit, here we only present the results on pairwise metric, i.e., Precision. The results on pointwise metrics are quite similar to it, please refer to Appendix D if interested.

Table 2: The overall performance of our proposed PRE models and baselines, evaluated by **Precision** metric. The bold text indicates the best performing model. †/†† indicates p-value of paired sample t-test where the method outperforms GPT-4 is less than 0.05/0.01. The methods in the above part of table are compared with GPT-4 under pairwise setting. The underlined text denotes that the LLM passes the pairwise / 5-level / 100-level qualification exam, respectively.

(a) PRE and single LLM models

Evaluation Models	XSum			NF-CATS		
	pairwise	5-level point	100-level point	pairwise	5-level point	100-level point
RWKV-4-Raven-7B	0.4972	0.0000	0.0000	0.5021	0.5083	0.4958
Alpaca-7b	0.5056	0.3249	0.3940	0.5286	0.5455	0.5155
Vicuna-7b	0.4948	0.4721	0.4732	0.5296	0.5557	0.5574
ChatGLM2-6B	0.5619	$\underline{0.5839}$	$\underline{0.6135}$	0.5414	0.5735	0.5958
Baichuan-2-13b	$\underline{0.6057}$	0.5471	0.5653	0.5515	0.5521	0.5500
Llama-2-70b-chat	$\underline{0.5719}$	$\underline{0.5848}$	$\underline{0.6704}$	$\underline{0.5891}$	0.5515	$\underline{0.6798}$
GPT-3.5-turbo	$\underline{0.6470}$	$\underline{0.6676}$	$\underline{0.6361}$	$\underline{0.6080}$	0.5586	0.5592
Claude-1	$\underline{0.6729}$	$\underline{0.6484}$	$\underline{0.6467}$	$\underline{0.6613}$	$\underline{0.5774}$	$\underline{0.5881}$
FastChat-t5-3b	$\underline{0.6921}$	$\underline{0.6291}$	$\underline{0.6302}$	$\underline{0.6537}$	0.5411	0.5708
ChatGLM-Pro	$\underline{0.6951}$	$\underline{0.6701}$	$\underline{0.7158}$	$\underline{0.7042}$	$\underline{0.6485}$	$\underline{0.6887}$
GPT-4	$\underline{0.7369}$	$\underline{0.6958}$	$\underline{0.7206}$	$\underline{0.7815}$	$\underline{0.6330}$	$\underline{0.6801}$
PRE w/o GPT-4 (ours)	0.7328	0.7242†	0.7334	0.7402	0.6604††	0.7074†
PRE (ours)	**0.7443**	**0.7331**††	**0.7390**††	**0.7842**	**0.6935**††	**0.7113**††

(b) Other baseline models

Evaluation Models	XSum	NF-CATS	models	XSum	NF_CATS
BERTScore (roberta)	0.5728	/	BLEU-1	0.5505	/
BERTScore (deberta)	0.5901	/	BLEU-2	0.5558	/
PandasLM	0.6350	0.7205	ROUGE-1	0.5884	/
GPTScore (flan-t5-xl)	0.6023	0.4762	ROUGE-2	0.5636	/
GPTScore (text-davinci-003)	0.6910	0.5940	ROUGE-l	0.5798	/

rameters, achieves a comparable evaluation level to larger-scale LLMs such as Claude-1 and ChatGLM-Pro when evaluating text summarization tasks. We speculate that this is caused by the detailed design of its instruct tuning strategy during training, which makes it effective in dealing with such specific rating tasks. By contrast, it performs relatively average in evaluating non-factoid QA tasks.

When comparing three different prompt settings, we find that the pairwise setting is slightly better than the pointwise ones, while the performance difference between the 5-level and 100-level pointwise settings is not significant. Therefore, we recommend using the pairwise setting when resources permit.

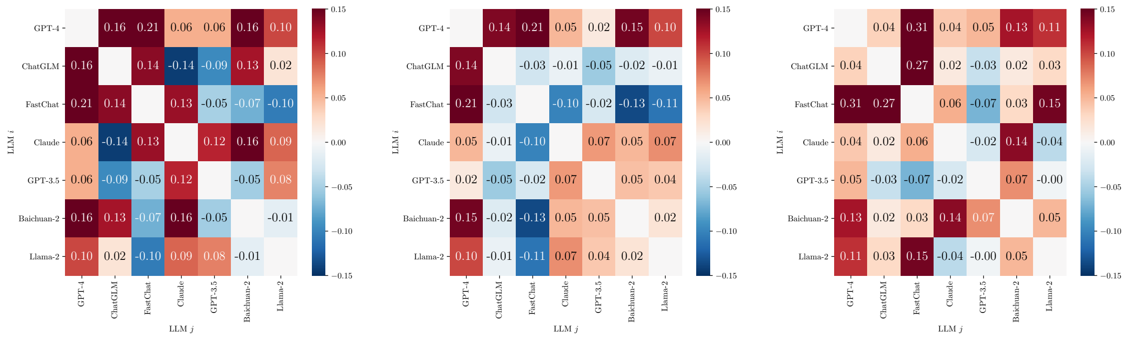
Table 2 also shows the performance of reference-based word similarity metrics such as ROUGE, BLUE, and BERTScore. We find that these metrics have positive correlations with human annotations, but the overall evaluation performance is worse compared to LLM-based methods like GPT-4 and PRE. PandaLM and GPTScore (text-davinci-003) show competitive performance in NF-CATS and XSum tasks respectively, but not in the other one. This phenomenon shows their performance is not robust across different tasks.

5.2 Bias Analysis (RQ2)

In this section, we dive into the results provided by different evaluation methods and investigate whether we could observe any type of evaluation bias in each of them. To measure potential bias in the evaluation using a single LLM, we propose the preference gap (PG) as an evaluation metric. Specifically, for LLMs i and j, we define the preference gap between LLM i and j ($\text{PG}(i,j)$) as the proportion of i's outputs that are better than j's outputs from i's perspective, subtracted by the same proportion from j's perspective, as shown in Eq 2. Naturally, the larger $\text{PG}(i,j)$ is, the more likely the bias exists between LLMs i and j. Ideally, for models without bias, the distribution of PG in the set $\{PG(i,j)|i \in L, j \in L, i \neq j\}$ is a random noise with a mean of 0.

$$\text{PG}(i,j) = P_i(i \succ j) - P_j(i \succ j) \qquad (2)$$

We conducted experiments under XSum tasks. Figure 3 shows the heatmap distribution of the PG metric among seven powerful LLMs under different settings in XSum tasks. In the pairwise, 5-level pointwise and 100-level pointwise settings, the proportions of PG values greater than 0 (i.e., i has stronger preferences than j on the output of i) are



((a)) Pairwise Setting ((b)) 5-level Pointwise Setting ((c)) 100-level Pointwise Setting

Figure 3: The severity of bias among seven powerful LLMs (measured with metric Preference Gap). Larger value (greater than 0) indicates a higher potential for bias between those two LLMs.

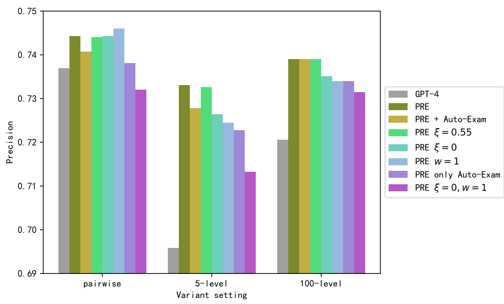


Figure 4: The performance of several PRE variants under different settings on XSum

66.67%, 57.14% and 76.19% respectively, which are all significantly higher than the 50% of the unbiased scenario. The results indicate significant bias in evaluation using individual LLMs under the pairwise and 100-level pointwise settings.

5.3 Robust Analysis (RQ3)

In this section, we aim to explore the robustness of the PRE model, that is, whether it still performs well when hyperparameters and qualification methods vary.

Here, we mainly attempted to adjust two hyperparameters: the pass threshold (ξ) for the qualification exam and the weight (w_l) used during rating aggregation. We adjusted ξ to 55% and 0, where $\xi = 0$ indicates all candidate reviewers are allowed to participate in the peer review process. We also adjusted w_l to be 1, which means all reviewers have equal rating weight.

We also tried an unsupervised qualification exam method called *Auto-Exam*, in which we evaluated the consistency of the LLM outputs before and after changing the order of content in the prompt. When the consistent proportion of such LLM exceeds a threshold η, this LLM is regarded as a reliable one to join in the reviewer set. In our experiments, we set $\eta = 55\%$.

We conducted experiments in XSum tasks, Figure 4 shows the performance of the PRE model under different hyperparameter and qualification settings, with GPT-4 used as the baseline. *PRE + Auto-Exam* denotes the variant of PRE method with both the original exam and Auto-Exam, while *PRE only Auto-Exam* denotes the variant with only Auto-Exam as the qualification exam and $w_l = 1$. Results show that the performance of the PRE model is not sensitive to the changes in its hyperparameters. Only when we remove all the effects of the qualification exam (i.e., $\xi = 0, w_l = 1$), does the performance of PRE noticeably decrease. This finding corroborates the necessity of LLM qualification filtering.

Figure 4 also shows the effect of *Auto-Exam* method. We find that PRE with only *Auto-Exam* outperforms the non-exam one ($\xi = 0, w = 1$), but its performance is lower than the qualification exam with a subset of manual annotation as ground truth. This finding indicates the potential of *Auto-Exam*, which deserves further exploration.

6 Conclusion

In this paper, we propose a novel framework, Peer Review Evaluator (PRE), for automatically evaluating the performance of large language models (LLMs). Inspired by the peer-review mechanism in the academic community, we introduce a mutual evaluation mechanism among LLMs in our framework. By setting reasonable qualification exams and model aggregation criteria, our PRE model outperforms all baseline methods including GPT-4. In the experiments, we also validate the existence of bias when using a single model like GPT-4 as an evaluation tool. PRE could reduce this bias to some extent. We believe that our proposed PRE, an automatic LLM evaluation method, can be adaptable to various evaluation tasks and scenarios.

7 Limitations

In this paper, we proposed a novel automatic LLM evaluation method based on the peer review mechanism. Here present some limitations:

(1) The qualification exam module deserves to be more explored. In the main experiment, we used part of the manual annotations as exam criteria, which is still some distance from the ideal purely automatic evaluation. In Sec 5.3, we attempted unsupervised metrics for qualification exams. The experimental results validate the potential of the automated exam.

(2) Due to the resource limit, our experiments were conducted only on two representative generative tasks, i.e., text summarization and non-factoid QA. The validity of our method warrants validation on more diverse datasets.

8 Potential Risks

The peer review mechanism proposed in this paper allows LLMs to take on the role of both evaluatee and evaluator when evaluating. While this mechanism improves evaluation efficiency, it also offers LLMs an opportunity to "cheat" in the qualification exams to increase their evaluation weights. This also poses a challenge to the designers of qualification exams: how to design an exam mechanism that avoids the potential for cheating to the greatest extent.

References

Rohaid Ali, Oliver Y Tang, Ian D Connolly, Patricia L Zadnik Sullivan, John H Shin, Jared S Fridley, Wael F Asaad, Deus Cielo, Adetokunbo A Oyelese, Curtis E Doberstein, et al. 2022. Performance of chatgpt and gpt-4 on neurosurgery written board examinations. *Neurosurgery*, pages 10–1227.

Anthropic. 2023a. Claude 2. https://www.anthropic.com/index/claude-2.

Anthropic. 2023b. Introducing claude. https://www.anthropic.com/index/introducing-claude.

Valeriia Bolotova, Vladislav Blinov, Falk Scholer, W. Bruce Croft, and Mark Sanderson. 2022. A non-factoid question-answering taxonomy.

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

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. *arXiv 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)*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, Alexis Chevalier, and Julius Berner. 2023. Mathematical capabilities of chatgpt. *arXiv preprint arXiv:2301.13867*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. *Blog post, April*, 1.

Zhouhong Gu, Xiaoxuan Zhu, Haoning Ye, Lin Zhang, Jianchen Wang, Sihang Jiang, Zhuozhi Xiong, Zihan Li, Qianyu He, Rui Xu, et al. 2023. Xiezhi: An ever-updating benchmark for holistic domain knowledge evaluation. *arXiv preprint arXiv:2306.05783*.

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*.

Myeongjun Jang, Deuk Sin Kwon, and Thomas Lukasiewicz. 2022. Becel: Benchmark for consistency evaluation of language models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3680–3696.

Andrew T Jebb, Vincent Ng, and Louis Tay. 2021. A review of key likert scale development advances: 1995–2019. *Frontiers in psychology*, 12:637547.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. *arXiv preprint arXiv:2302.14520*.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*.

Ann Lehman, Norm O'Rourke, Larry Hatcher, and Edward Stepanski. 2013. *JMP for basic univariate and multivariate statistics: methods for researchers and social scientists*. Sas Institute.

Ruosen Li, Teerth Patel, and Xinya Du. 2023. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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

LMSYS. 2023. Fastchat. https://github.com/lm-sys/FastChat.

Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. 2023. Gpteval: A survey on assessments of chatgpt and gpt-4. *arXiv preprint arXiv:2308.12488*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745.

OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt.

OpenAI. 2023. Gpt-4 technical report. *arXiv 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 of the Association for Computational Linguistics*, pages 311–318.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, et al. 2023. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*.

Hao Sun, Zhexin Zhang, Jiawen Deng, Jiale Cheng, and Minlie Huang. 2023. Safety assessment of chinese large language models. *arXiv preprint arXiv:2304.10436*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

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*.

Baichuan Intelligent Technology. 2023. Baichuan-7b. https://github.com/baichuan-inc/Baichuan-7B.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. 2023. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Liang Xu, Anqi Li, Lei Zhu, Hang Xue, Changtai Zhu, Kangkang Zhao, Haonan He, Xuanwei Zhang, Qiyue Kang, and Zhenzhong Lan. 2023. Superclue: A comprehensive chinese large language model benchmark. *arXiv preprint arXiv:2307.15020*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Xiang Yue, Boshi Wang, Kai Zhang, Ziru Chen, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. *arXiv preprint arXiv:2305.06311*.

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. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

A More Details about Framework Implement

Here, we extend more details of the Sec. 4.4.

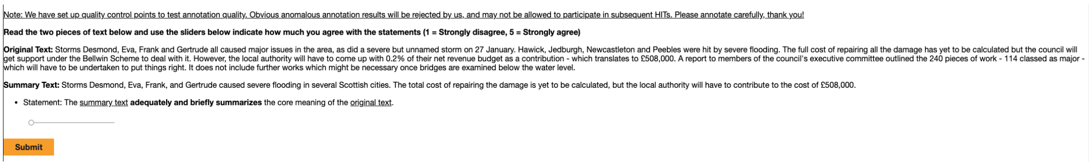
A.1 Qualification exam module

To test the ability of reviewer candidate LLMs, we selected the outputs (i.e., summaries of test documents) of three evaluatee LLMs with varying quality: GPT-3.5-turbo, Fastchat-t5-3b, and Alpaca-7b, as "questioners". Reviewer candidates are asked to rate these summaries. We designed three rating methods: **5-level pointwise**, **100-level pointwise**, and **pairwise** (or called **preference**). In both the 5-level and 100-level pointwise rating methods, the candidate LLMs need to rate an integer number for each (text, summary) pair to indicate its summarization quality. The differences between 5-level and 100-level settings are not only in the rating scale and granularity (1-5 levels and 0-100 levels), but also in the guidance style: the 5-level method offers detailed definition of each level, while the 100-level method only provides a general description on the quality tendency. The pairwise rating method requires candidate LLMs to rate the preference for each (text, summary 1, summary 2) tuple, determining which summary better summarize the text. To reduce bias caused by word position and frequency, we constructed two prompt samples ((t, s_1, s_2) and (t, s_2, s_1)) for each text-summary-summary tuple (t, s_1, s_2) in our experiments.

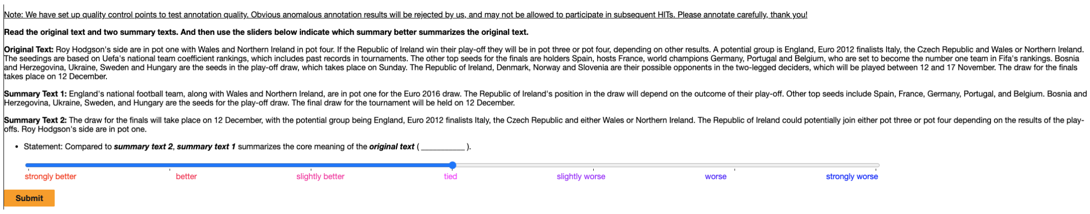
We uniformly designed prompts for these three rating methods, as specified in the Appendix C. Additionally, we collected human preferences as the ground truth for the exam, and then used Precision (e.g., in the pointwise cases like 5-level or 100-level ratings, convert the rates to pairwise preferences first) in the pairwise mode as the evaluation metric to rate the evaluation ability of candidate LLMs. Only when an LLM's Precision exceeds the threshold of ξ, it will be retained as a reviewer for the peer review process. In our experiments, we set ξ to be 60%.

A.2 Peer review module

For the text summarization task, we have devised a unified set of prompts to be fed into the whole eleven evaluatee LLMs. Specifically, we utilized the prompt template "*Task: Generate a short summary of the text in at most 64 words. Text: {original text} Summary:*". Then, only the LLMs that pass the qualification exam are deployed to rate the



((a)) Pointwise assessment



((b)) Pairwise assessment

Figure 5: Examples of the MTurk annotation user interface

outputs of evaluatee LLMs. We fed the prompts designed in the Appendix C into reviewer LLMs and collected their scoring results. Overall, in the pointwise and pairwise modes, each reviewer LLM is required to generate $11 \times 100 = 1,100$ rates or $Perm(11, 2) \times 100 = 11,000$ preferences, respectively.

A.3 "Chair" decision module

In Sec 3.2.3, Eq 1 already demonstrates the core idea of the weighted voting strategy. For pointwise and pairwise modes, we have different implementation details:

For the pointwise mode, each text-summary pair is treated as a sample x. We first need to normalize the original LLM output score $r_x^{(l)}$ using mean-variance normalization to eliminate its weighting effect. The weight of reviewer LLM w_l is determined by its Precision p_l in the qualification exam. In the experiments, we set $w_l = \log\left(\frac{p_l}{1-p_l}\right)$, just as Eq 3 shows, where W is the normalization term.

$$R_x = \frac{1}{W}\sum_{l \in L} w_l \tilde{r}_x^{(l)} = \frac{1}{W}\sum_{l \in L} \log\left(\frac{p_l}{1-p_l}\right) \frac{r_x^{(l)} - \mu_l}{\sigma_l} \tag{3}$$

For the pairwise mode, let x represent a text-summary-summary tuple $(t_x, s_{1,x}, s_{2,x})$. Each reviewer LLM l votes for its preference output $r_x^{(l)}$ (either $s_{1,x}$ or $s_{2,x}$) with weight w_l. The preference result of the aggregated PRE model is determined by the summary with the higher votes. In our experiments, we also set $w_l = \log\left(\frac{p_l}{1-p_l}\right)$, just as shown in Eq 4. The function $I(\cdot)$ denotes as the 0-1 Indicator function.

$$R_x = \arg \max_{s \in \{s_{1,x}, s_{2,x}\}} \sum_{l \in L} \log\left(\frac{p_l}{1-p_l}\right) I(r_x^{(l)} = s) \tag{4}$$

B Details of Manual Annotations

Just as Sec 4.5 discussed, we recruited annotators to conduct judgments under both pointwise and pairwise settings.

Here let us use the XSum dataset as an instance to introduce the annotation details. The design of NF-CATS dataset is quite similar to it. Under the pointwise setting, for each assessment task, the annotators are provided with a (text, summary) pair, then they need to give a rating on the range from integer 1 to 5. We adopted the Likert scale (Jebb et al., 2021) as the 5-level annotation guidance: For the statement "*The summary text adequately and briefly summarizes the core meaning of the original text*", levels $1 \sim 5$ respectively represent annotator strongly-disagrees/disagrees/neutralizes/agrees/strongly-agrees with the above statement. Figure 5(a) shows examples of the UI used in pointwise assessments.

Under the pairwise setting, we designed a 7-level annotation rule ranging from -3 to 3. Specifically, levels $-3 \sim 3$ respectively represent "*compared to summary text 2, summary text 1 summarizes the core meaning of the original text strongly-better/better/slightly-better/tied/slightly-worse/worse/strongly-worse*". The difference compared with general 7-level setting is that the assessment results are allowed to be any real number within the range $[-3, 3]$ to improve the annotation experience of annotators. Figure 5(b) shows examples of the UI used in pairwise assessments.

C The Design of Evaluation Prompt

The evaluation prompts are adopted for both *qualification exam* and *peer review* modules (detailedly introduced in Sec 3.2). These prompts are fed to the reviewer (or reviewer candidate) LLMs, allowing them to generate ratings or preferences. In our experiments, we have proposed three different prompt settings (pairwise, 5-level pointwise and 100-level pointwise), and then seperately designed the prompt template for each setting, as the following shows. Here we show the design in XSum dataset under each setting, the design of NF-CATS dataset is almost the same.

C.1 Pairwise setting

###Task: Evaluate two summaries of a given passage and determine which one better summarizes the main points of the passage considering accuracy and conciseness. You only need to output ‘one‘ or ‘two‘ directly to indicate which summary summarizes the passage better.

###Passage: { *passage* }

###Summary one: { *summary 1* }

###Summary two: { *summary 2* }

###Output:

C.2 5-Level pointwise setting

###Task: Evaluate the summary of a given passage and determine how it summarizes the main points of the passage considering accuracy and conciseness. Directly output a number between 1 and 5 to indicate the quality score of this summary:
- 1 means the summary is not relevant to the passage,
- 2 means the summary is neither accurate nor concise but it is relevant to the passage,
- 3 means the summary is only a fair summary of the passage considering accuracy and conciseness,
- 4 means the summary is a good summary of the passage but still has room for improvement in accuracy and conciseness,
- 5 means the summary is a perfect summary of the passage considering accuracy and conciseness.

###Passage: { *passage* }

###Summary: { *summary* }

###Score of the summary:

C.3 100-Level pointwise setting

###Task: Evaluate the summary of a given passage and determine how it summarizes the main points of the passage considering accuracy and conciseness. Directly output a number between 0 and 100 to indicate the score of this summary. The higher the score, the more accurate and concise the summary is.

###Passage: { *passage* }

###Summary: { *summary* }

###Score of the summary:

D Overall Performance on Pointwise Metrics

Table 3 shows the overall performance on pointwise metrics. The results are quite similar to the findings on Table 2: Our proposed PRE model outperforms all the baseline models.

Table 3: The overall performance of our proposed PRE models and baselines, evaluated by pointwise metrics, i.e., **Kendall's tau** (τ) and **Spearman correlation coefficient** (S). The bold text indicates the best performing model. †/†† indicates p-value of paired sample t-test where the method outperforms GPT-4 is less than 0.05/0.01. The methods in the above part of table are compared with GPT-4 under 5-level point setting.

(a) PRE and single LLM models

models	XSum				NF-CATS			
	5-level point		100-level point		5-level point		100-level point	
	τ	S	τ	S	τ	S	τ	S
RWKV-4-Raven-7B	/	/	/	/	0.0277	0.0482	-0.0277	-0.0334
Alpaca-7b	0.0335	0.0500	0.0306	0.0506	0.0489	0.0856	0.0250	0.0390
Vicuna-7b	-0.0028	-0.0135	0.0175	0.0330	0.0854	0.1738	0.0925	0.1512
ChatGLM2-6B	0.1305	0.2268	0.1406	0.2172	0.0990	0.1664	0.1394	0.2333
Llama-2-70b-chat	0.1386	0.3079	0.2628	0.4503	0.0950	0.1908	0.2184	0.3576
Baichuan-2-13b	0.0941	0.2255	0.1185	0.2271	0.0865	0.1735	0.0833	0.1381
GPT-3.5-turbo	0.2495	0.4199	0.2029	0.3165	0.0757	0.1577	0.0929	0.1520
Claude-1	0.2491	0.4650	0.2702	0.4321	0.1023	0.1949	0.0795	0.1355
FastChat-t5-3b	0.2090	0.3935	0.2195	0.3295	0.0638	0.1439	0.1107	0.1716
ChatGLM-Pro	0.2662	0.4898	0.3129	0.4868	0.2038	0.3605	0.2357	**0.3893**
GPT-4	0.3098	0.4929	0.3290	0.4845	0.1776	0.3318	0.2052	0.3287
PRE w/o GPT-4 (ours)	0.3271	0.4835	0.3052	0.4543	0.2043	0.3451	0.2329	0.3601
PRE (ours)	**0.3452††**	**0.4998**	**0.3319**	**0.4947**	**0.2348**	**0.3843**	**0.2438**	0.3735

(b) Other baseline models

models	XSum		NF-CATS		models	XSum	
	τ	S	τ	S		tau	S
BERTScore (roberta)	0.1562	0.2379	/	/	BLEU-1	0.1371	0.1969
BERTScore (deberta)	0.1829	0.2715	/	/	BLEU-2	0.1252	0.1864
PandaLM	/	/	/	/	ROUGE-1	0.1662	0.2448
GPTScore (flan-t5-xl)	0.1486	0.2286	-0.0048	0.0033	ROUGE-2	0.1214	0.1789
GPTScore (text-davinci-003)	0.2848	0.4203	0.1238	0.1966	ROUGE-l	0.1524	0.2329