Exploring Side-by-Side LLM Evaluation Through Human Alignment and Bias Mitigation

Kseniia Titova[1,2] **Darina Rustamova**[1,3] **Alan-Barsag Gazzaev**[1,3]
Maksim Polushin[1] **Valentin Malykh**[1] **Sergey Zagoruyko**[4]
[1]MWS AI [2]Skoltech [3]ITMO University [4]Independent contributor

Abstract

With the rise of large language models, evaluating their outputs has become increasingly important. While supervised evaluation compares model responses to ground truths, dialogue models often use the Side-by-Side approach, where a judge compares the responses of baseline and candidate models using a predefined methodology. In this paper, we conduct an in-depth analysis of the Side-by-Side approach for evaluating models in text generation as well as for code generation and investigate the circumstances under which LLM-evaluators can be considered an alternative to expert annotation. We propose and publicly release a methodology that can enhance the correlation between automatic evaluation and human annotation through careful prompt engineering and adding model reasoning. We demonstrate the problem of positional bias and propose metrics for measuring it, as well as ways to mitigate it.

1 Introduction

As large language models (LLMs) rapidly advance, evaluating them effectively has become a crucial task that can be approached from various angles. Evaluation methods for these models are typically divided into supervised and unsupervised approaches. The supervised method involves comparing the model's responses to ground-truth answers. Such methods imply a straightforward output format, where the model is required to classify, select, match options, or generate a short answer, after which automatic metrics like accuracy, exact match (EM), and F1-score are used. The model's abilities in tasks such as question answering, common sense, and reasoning are tested in this way.

However, for models intended for interacting with users, providing a perfect answer to every query is not always possible. In these situations, the Side-by-Side (SbS) approach is frequently employed, where an independent judge compares the responses of a candidate model with those of a baseline model. The comparative element in this method helps avoid bias in judges' evaluations while allowing for the assessment of the overall quality of the dialogue agent's response.

Due to the indeterminacy and inconsistency of the evaluation criteria for this method, we decided to explore its characteristics using the example of evaluating language models. In this paper, we examine SbS evaluation by comparing its manual execution with execution using an LLM-based evaluator, and also present ways to improve this approach. We aim to answer two main research questions:

1. Is the issue of positional bias still relevant? How can we address it in SbS evaluation?
2. How does the formulation of the prompt for the LLM-as-judge help, and to what extent?

We propose a methodology that can increase the correlation between model-as-judge assessments and human assessments while exploring ways to significantly improve the performance of a judge with a relatively small number of parameters. We demonstrate that minor changes in the task formulation for the evaluator model can significantly enhance the quality of its evaluation. Our comparative analysis of various open and closed commercial models using our benchmark helps us assess the impact of prompt-engineering techniques on the quality of evaluation.

2 Related Work

Prior to the emergence of robust large language models, evaluations of natural language generation systems often relied on automated metrics such as BertScore Zhang et al. (2020) and GPTScore Fu et al. (2023). Although these metrics offer scalability, they do not fully capture the subtlety and context sensitivity that human judgments can provide. Human evaluators, traditionally considered the "gold standard" for the assessment of NLG Ouyang et al. (2022), remain critical for tasks that require deep linguistic and domain expertise. However, human evaluations introduce issues of subjectivity, potential biases, and reproducibility challenges Clark et al. (2021); Belz et al. (2023). They are also time-consuming, resource-intensive, and limited by the slower processing speed of human annotators. As a result, leveraging powerful LLMs (e.g., GPT-4) to approximate or even replace human annotators has gained prominence Zheng et al. (2023); Chiang et al. (2023); Liu et al. (2023); Zhu et al. (2023). These LLM-based evaluators can achieve high agreement with human preferences, yet they too exhibit specific biases, such as position bias, verbosity bias, and self-enhancement bias Zheng et al. (2023). To address these shortcomings, researchers have introduced strategies such as generating chain of thought plans Wei et al. (2023) for more transparent evaluations. However, this technique has limited effectiveness for tasks that do not involve mathematical or logical reasoning Sprague et al. (2024).

The emergence of "thinking" models, which incorporate reasoning processes before delivering final outputs Wu et al. (2024); Guo et al. (2025); OpenAI (2024b), marks a significant advancement in the evaluation of other large language models Hosseini et al. (2024); Saha et al. (2025). Our research builds upon these developments by focusing on the application of "thinking" models specifically designed to evaluate other LLMs. By examining the alignment of these models with human judgment, we seek to assess their accuracy, fairness, and transparency compared to traditional metrics.

3 Methodology

3.1 Data collection

We manually collect datasets for both text and code generation tasks. The first dataset consists of 2396 instructions, all manually written in Russian. These instructions are sourced from logs of an LLM-based bot specifically designed for dataset collection, and each entry is then manually cleaned. As a result, the dataset is entirely human-authored and curated, reflecting the most typical requests users submit to a conversational LLM. Examples of the dataset can be found in the Appendix C.

We also collect a dataset of 800 instructions with various coding tasks for comparing code model generations. The questions in the dataset are divided into five categories: writing docstrings, creating unit tests, text-to-code, refactoring and explaining a piece of code.

3.2 Side-by-Side method

We use a Side-by-Side evaluation method, where both human experts and large language models act as judges. For each prompt from our dataset, we generate responses from two models — the baseline and the candidate — and then present these response pairs to the judges for comparison.

In many studies employing similar approaches, the judge is limited to a binary choice: indicating whether the first or second model produced the better response. In our approach, we broaden the set of possible outcomes by allowing judges to indicate if both responses are equally good or equally bad. Thus in our case, the judge states that either **a)** whether the response from the candidate model is better than the baseline, **b)** vice versa, **c)** both responses are good or **d)** both responses are bad. The names of the models are concealed from the judges.

Naturally, this approach needs to be formalized to standardize the evaluations. With a well-defined task and properly specified criteria, we aim to align the model-based assessment results as closely as possible with human annotations. In the next section, we describe the design of our evaluation methodology.

3.2.1 Manual evaluation

We prepare directives for the annotators to evaluate pairs of model responses. To determine which response is superior, each pair is assessed and compared against several criteria, listed in order of decreasing importance:

1. **[Safety]** The response should not contain information that could harm an individual.
2. **[Ethics]** The response must adhere to ethical standards: it should not be rude, offensive, biased, or judgmental.
3. **[Truthfulness]** The response should not contain inaccurate or questionable statements. The expert refers to the attached factual reference to verify the truthfulness of the response.
4. **[Relevance]** The response should align with the request: it must follow the instructions, avoid answering unnecessary questions, and be in the required language.
5. **[Completeness]** The response should be thorough and comprehensive.
6. **[Style]** The response should be written with correct spelling, punctuation, and syntax, and should avoid informal language, unless explicitly stated otherwise in the instructions.

The order of the model responses in each pair is randomized. The manual evaluation is conducted with an overlap of three people. To determine the final verdict from expert judges, we employ an ensemble strategy that aggregates three individual assessments into a single unified decision, prioritizing the majority opinion. The details of this strategy are described in 8.

Background information about the team of annotators, including details about their age and education, can be found in the Appendix A.

3.2.2 Automatic evaluation

We develop several prompts for LLM evaluators that take into account the criteria described in the previous section.

Instead of randomizing the order of model responses, we perform two runs through the dataset. In the first run, the prompt places the candidate model's response first followed by the baseline model's response; in the second run, the order is reversed. The scores are averaged after the two runs are completed. We could shuffle the model responses within pairs for the LLM-judge input to save its runtime, as we do for human experts. However, conducting two separate runs allows us to analyze the presence of positional bias in the tested evaluators.

An important task in preparing the evaluator model is the preparation of the prompt. Prompt I is designed to briefly describe the task of SbS evaluation. In Prompt II, we aim to address and describe all the criteria listed for the team of experts. We also attempt to add the following modifications to the prompt.

Reasoning

We ask the model to reflect before reaching a verdict, to analyze responses based on each criterion, and to aggregate scores when providing a comparison result. Some models have been specially trained to reason Guo et al. (2025); OpenAI (2024b), for which such an addition to the prompt presumably will not make any difference.

We find an issue with models trained on reasoning and those evaluated with reasoning prompts — a significant portion of the answers (¿10%) consists not of the expected symbols representing one of four classes, but a different response. Therefore, when using reasoning, we make the model strictly adhere to formatting.

Multi-agent approach

The reasoning of the evaluator's language model helps improve performance when evaluating responses from other models. However, despite the advantages of the Chain-of-thought (CoT) method, when the model reasons step by step, there is a problem called Degeneration-of-thought Liang et al. (2023), when the LLM begins to be confident in its reasoning, even if it is not correct.

	Judge	Parameters	SBS results					CK	MPCC	HF Hub	Citation
			A	B	C	D	E				
	manual	21 experts	4.6	44.5	23.7	27.2	0.0				
English-aligned models	llama3.1-405b	405B	36.6	30.9	31.6	0.8	0.0	0.165	0.507	link	Dubey et al. (2024)
	llama3.3-70b	70B	34.1	39.0	26.0	1.0	0.0	0.116	0.595	link	Dubey et al. (2024)
	gpt-4o	-	17.6	13.6	46.0	22.7	0.0	0.254	0.495	-	OpenAI (2024a)
	o1-mini	-	32.7	40.3	18.1	9.0	0.0	0.194	-	-	OpenAI (2024b)
	gpt4	-	23.8	24.0	51.3	0.8	0.1	0.137	**0.642**	-	Achiam et al. (2023)
	deepseek-r1-dst.	70B	24.2	33.5	31.9	7.3	3.1	0.168	0.640	link	Guo et al. (2025)
	deepseek-v3	671B (37B)	11.5	47.7	34.9	1.0	4.8	**0.374**	0.570	link	Liu et al. (2024)
	claude sonnet	175B	13.6	9.1	71.1	1.0	5.2	0.017	0.427	-	Anthropic (2024)
	claude opus	137B	23.3	35.0	20.2	21.4	0.2	0.205	0.602	-	Anthropic (2024)
Russian	T-lite-it-1.0	7.6B	18.4	26.7	39.5	15.4	0.0	0.079	0.139	link	T-bank (2024)
	T-pro-it-1.0	32.8B	40.6	44.6	4.1	10.7	0.5	0.118	0.397	link	T-bank (2024)
	GigaChat-Max	70-100B	24.9	38.9	30.7	0.9	4.6	0.148	-	-	Sber (2024)
	YandexGPT	-	29.8	43.2	7.4	0.2	19.5	0.125	0.572	-	Yandex (2024)

Table 1: **Comparative analysis of LLMs as judges for SbS Evaluation in Russian.** Various models of different sizes, aligned with both English and Russian languages, were selected as judges. Prompt I was used for obtaining verdicts. The percentage distribution of verdicts across the entire benchmark is represented by symbols: A) the candidate model's answer is better, B) the baseline model's answer is better, C) both models' answers are equally good, D) both models' answers are equally poor. Symbol E refers to cases where the model returns something different from one of the four listed verdicts. The average value for each verdict across two benchmark runs is provided for the LLM evaluators. Additionally, we include a series of metrics that show correlation with expert assessments. Their descriptions and formulas are provided in Section 4.2.

Authors provide an example of a multi-agent approach that avoids this problem. To do this, agent-1 expresses its opinion on a task, agent-2 responds to this, and after the agents' dialogue, the agent-judge analyzes the agents' responses and issues a final verdict.

Based on this research, we propose the following two schemes of a multi-agent approach.

1. **Soft**. Agent-1 makes its assessment regarding a pair of proposals, and agent-2 either agree or disagree with agent-1. Next, the agent-judge makes its verdict based on the two previous verdicts.
2. **Hard**. Agent-1 makes its assessment regarding a couple of proposals, and agent-2 always disagrees with agent-1. After that the agent-judge makes its verdict based on the two previous verdicts.

All variations of the prompts can be found in Appendix B.

4 Experiments

For our experiments in text generation we select Qwen2.5-32B-Instruct Yang et al. (2024) fine-tuned on the Russian language as the candidate model and GPT-4o OpenAI (2024a) as the baseline model. The parameters for generating responses on the benchmark are the same for all models, their values can be found in the Appendix. The paired generations are shuffled and given to a team of experts for annotation (with the model names concealed) along with the evaluation methodology described in Section 3.2.2. These same generations are also evaluated by LLM judges.

4.1 Analysis of manual evaluation

We provide the expert evaluators with universal criteria for assessment through guideline; however, this does not guarantee full correlation among their responses. We believe it is expected and acceptable for annotators to have differing opinions when evaluating pairs of responses, which is precisely why our assessment involved an overlap.

The dataset is divided into parts consisting of 600 questions each, and each of them is evaluated independently by three different people. We calculate the Cohen's kappa (CK) between each pair of

Judge model	MPCC		PCon@AB
	Consistency	Δ	
llama3.1-405b	0.526	0.058	0.336
llama3.3-70b	0.599	0.249	0.476
gpt-4o	0.666	0.035	0.329
gpt4	0.674	0.259	0.339
deepseek-r1-dst.	**0.846**	**0.012**	**0.500**
deepseek-v3	0.164	0.077	0.271
claude sonnet	-0.275	0.241	0.106
claude opus	0.598	0.125	0.431
T-lite-it-1.0	-0.370	0.093	0.122
T-pro-it-1.0	0.363	0.179	0.400
YandexGPT	0.319	0.092	0.409

Table 2: **Comparative analysis of evaluator scores with and without swap of models' answers.** Metrics **MPCC-Consistency**, **MPCC-Δ** and **PCon@AB** indicate the presence of positional bias among LLM-evaluators. The closer the values of metrics **MPCC-Consistency** and **PCon@AB** are to one, the more consistent the model is when the positions of answers in prompt are changed; while lower **MPCC-Δ** indicates lower positional bias.

annotators for each of the four splits of the dataset. The CK ranges from **0.665** to **0.732** depending on the dataset split. We conclude that if the same measure of any model evaluator falls within the specified range, it can likely be considered as a good judge option. The exact metrics can be found in Appendix A.

We acknowledge the possible presence of biases among human annotators. However, in this paper, our focus is primarily on the overall performance of LLMs-as-judges, as well as on examining biases inherent to the models themselves.

4.2 Analysis of automatic evaluation

We select a range of models of different sizes as LLM-as-judges, including both open-source and commercial models. The results of the manual and automatic evaluations for Russian text generation can be found in Table 1. When evaluating judge models, we primarily focus on the consistency of judgments with experts' assessments. We highlight several types of metrics for this task:

1. Nominal metrics: We treat the classes A/B/C/D as unordered categories. When calculating such metrics, we do not impose additional information on specific judgments. This category includes simple metrics like Accuracy, Precision/Recall calculated by class, and Cohen's kappa for measuring agreement among multiple raters.
2. Ordinal and interval metrics: we impose an order on the four classes or, respectively, go further and map the classes onto a real line. This transforms the task into one of binary classification, "model vs model," allowing for metrics that use correlation coefficients.

For analyzing the consistency of LLM judgments with human judgments, we choose the nominal metric Cohen's kappa and propose our own interval metric, the Median Pearson Correlation Coefficient (MPCC). In MPCC, we calculate the median for a set of judgments using the formula:

$$\text{Median} = \frac{\sum \mathbf{A} + \sum \mathbf{C}}{\sum \mathbf{A} + \sum \mathbf{B} + 2 \cdot \sum \mathbf{C}}.$$

This characteristic ranges from 0, 1 and indicates how much better the candidate model is compared to the baseline model. We apply a sliding window with a size of N and a stride of K across all verdicts from the benchmark and calculate the median for each batch. We then obtain a set of medians for both expert and model verdicts and calculate the Pearson Correlation Coefficient (PCC) between them.

Prompt	SBS results				CK	MPCC	PCon@AB
	A	B	C	D			
experts	4.6	44.4	23.7	27.2			
I	24.2	33.5	31.9	7.3	0.169	0.589	**0.500**
II	14.5	26.0	44.1	15.4	0.184	0.623	0.361
II-fact	14.5	26.1	44.2	15.1	0.204	0.606	0.355
II-reason	17.9	29.4	43.9	8.4	**0.219**	**0.639**	0.412
II-fact+reason	17.9	28.9	43.9	8.9	0.202	0.636	0.384

Table 3: **Analysis of deepseek-r1-distill-llama judge model with different prompts.** The proportion of responses and the PCC with expert evaluation are provided for Prompt I, Prompt II, as well as variations of Prompt II with the additions of factual background and reasoning.

Prompt	SBS results				CK	MPCC	PCon@AB
	A	B	C	D			
experts	4.6	44.4	23.7	27.2			
I	13.1	20.3	63.8	2.8	0.163	0.573	0.269
II	9.0	11.5	57.5	21.9	**0.205**	0.505	0.146
II-fact	9.0	11.5	57.6	21.9	0.204	0.504	0.145
II-reason	31.5	31.6	32.0	4.7	0.198	**0.653**	**0.499**
II-fact+reason	30.3	31.7	33.5	4.2	0.195	0.639	0.497

Table 4: **Analysis of llama3.3-70b judge model with different prompts.** The proportion of responses and the PCC with expert evaluation are provided for Prompt I, Prompt II, as well as variations of Prompt II with the additions of factual background and reasoning.

Both metrics FK and MPCC are averaged over two runs: one with the direct order of responses in the prompt and the other with the reverse order. Overall we consider both metrics to assess the correlation between LLM and expert verdicts.

We suggest looking not only at the correlation coefficients but also at the proportions of verdict returned by the judges. In addition to high correlation with manual evaluation, it is important for the LLM to replicate significant statistical patterns. For example, in Table 1 according to expert judgment, we can see that the baseline model answers better significantly more often than the candidate model. For many evaluator models, however, the number of positive (A) statements is often close to the number of negative (B) statements. Judging by both FK and MPCC we conclude that Claude Opus and Deepseek-v3 show the best correlation with manual assessments among all the tested LLM-as-judges for the Russian language.

4.3 Impact of positional bias

Table 2 presents a study of LLM judges for positional bias. We perform two measurements for each evaluator model - without models' answers swap and with - and calculate the PCC of aggregated values with manual annotation. We introduce metric **PCon@AB** that indicate the presence of bias in the evaluator models.

$$\textbf{PCon@AB} = 0.8\frac{\sum_{BM} \mathbb{1}(J_{\text{swap}=0} = J_{\text{swap}=1} | J = \mathbf{A} \vee \mathbf{B})}{\sum_{BM} \mathbb{1}\left((J_{\text{swap}=0} = \mathbf{A} \vee \mathbf{B}) \vee (J_{\text{swap}=1} = \mathbf{A} \vee \mathbf{B})\right)},$$

where *J* stands for *judgement* and can take the values A, B, C, or D. This metric shows the consistency of the model's answers without swap and with - it indicates the proportion of matching answers among answers A and B given the different order of model responses.

The metric **MPCC-Consistency** is calculated as the Pearson correlation coefficient between two sets of medians obtained for the verdicts with and without swap, while the metric **MPCC-**Δ is the difference between the MPCC calculated separately for the verdicts obtained with and without swap.

Judge model	verdicts				CK	MPCC	PCon@AB
	A	B	C	D			
manual	28.4	17.8	23.1	30.8			
llama3.1-405b	29.0	4.4	9.0	57.6	0.174	0.427	0.364
llama3.3-70b	41.3	9.2	17.3	32.2	**0.498**	0.432	**0.439**
deepseek-r1-dst.	35.4	19.3	25.6	19.7	0.395	**0.460**	0.424
deepseek-v3	24.4	6.8	4.1	64.8	0.379	0.089	0.241
claude sonnet	17.4	0.8	6.4	75.4	0.286	0.312	0.098
claude opus	25.4	11.3	27.1	36.3	0.194	0.392	0.089

Table 5: **Comparative analysis of LLMs as judges for SbS Evaluation for code.** Various models of different sizes were selected as judges. Prompt II was used for obtaining verdicts. The average value for each verdict across two benchmark runs and PCC with expert assessments is provided.

PCon@AB, **MPCC-Consistency** and **MPCC-Δ** do not rely on manual annotation, allowing us to determine how prone the model is to positional bias without expert involvement.

4.4 Elevating LLM-as-judge performance

In this section, we address two questions: **a)** how much can we increase the correlation with manual annotation by constructing prompts; **b)** can prompts help with the positional bias issue? As suggested in Section 3.2.2, we create several prompt variations and measure two LLMs-as-judges with each: Deepseek-r1-distill-llama and Llama3.3-70b.

Table 3 shows the comparison for the first model: after updating prompt I to II, the correlation with manual annotation significantly increased. Modifying prompt II by adding a reasoning step increases the model's correlation with expert judgments even further.

The patterns hold for the model Llama3.3-70b, as can be seen in 4. Adding a request to reason in the prompt not only slightly increases correlation with experts but also significantly enhances the model's robustness against positional bias.

We formulate several conclusions that we consider foundational for our methodology based on the results of these experiments. We recommend them as guidelines for performing similar evaluations.

- While the issue of positional bias remains significant for LLM-as-a-judge in the SbS task, **it can be almost entirely avoided by using models trained to reason. For other models, the effect can also be reduced by asking the model to reason beforehand.**
- **A well-crafted prompt can significantly increase correlation, but the prompt should be tailored individually for each model as it is not transferable between different LLM-as-judges.** From Table 3, we see that as the complexity of the prompt increases, the correlation of the Deepseek-r1-dst-llama model with human labeling rises, nearly reaching the quality of a larger model.
- According to the Cohen's kappa metric, **none of the LLMs achieve a sufficiently high level of correlation with manual evaluation** (Section 4.1). We believe this is primarily due to the positional bias of the models and the inconsistency of their verdicts. Further improvement of the prompt may help increase the model's alignment with manual evaluations.

4.5 SbS evaluation for code generation

We also conduct similar experiments for models intended for code generation, using Prompt II for LLM evaluators. From the Table 5 we see that Llama models have generally the best performance in terms of the FK and MPCC metrics, while Claude Opus and Deepseek-r1-dst can still be considered as strong options.

Based on the data presented in Tables 1 and 5, we observe a positional bias in the performance of LLMs when used as judges.

4.6 Analysis of multi-agent approach

Deepseek-r1-distill-llama is chosen as the model-judge for experiments with the multi-agent approach, as it is a model trained on reasoning. As can be seen from Table 6, the hard method yields the same results as the baseline method with Prompt II.

Judge model	method	verdicts					CK
		A	B	C	D	E	Mean
manual		4.6	44.5	23.7	27.2	0.0	
deepseek-r1-dst.	base	24.2	33.5	31.9	7.3	3.1	0.168
	soft	22.0	34.3	32.4	9.3	2.0	**0.210**
	hard	21.7	33.4	36.0	5.4	3.5	0.172

Table 6: **Multi-agent approach.** Measurement results for the Deepseek-r1-distill-llama model as a judge. We managed to increase the correlation with manual annotation using the soft approach.

At the same time, the soft method increases correlation with experts, since it is most likely that the second agent does not necessarily contradict, but sometimes complements the reasoning of the first agent, and the agent judge re-evaluates all statements based on previous reasoning. This variation of Multi-Agent Debate is a strong method that develops the idea of COT.

5 Conclusion

In this work, we present a methodology for conducting Side-by-Side evaluations using language model evaluators, which we apply to compare open and closed commercial large language models as judges. We highlight the significance of the positional bias issue and propose metrics for its evaluation during automatic SbS assessments, as well as suggest methods for mitigating its impact.

Additionally, we suggest ways to make language model evaluations align better with human ratings. This involves demonstrating the importance of prompts in conducting evaluations using our methodology, and emphasize the need for a tailored approach to crafting prompts for each evaluator model. We also assess the impact of adding reasoning on the judging model's capabilities and its influence on correlation with manual annotations.

Ethics Statement

We acknowledge that our study does not directly investigate biases in expert assessments. There is a possibility that patterns inherent in the candidate and baseline model responses may enable annotators to infer model identity, and human evaluators may also exhibit positional biases. We recognize these as ethical considerations and potential limitations in the interpretation of our results.

References

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

Anthropic. Introducing the next generation of claude. 2024. Available at: https://www.anthropic.com/news/claude-3-family.

Anya Belz, Craig Thomson, Ehud Reiter, Gavin Abercrombie, Jose M. Alonso-Moral, Mohammad Arvan, Anouck Braggaar, Mark Cieliebak, Elizabeth Clark, Kees van Deemter, Tanvi Dinkar, Ondřej Dušek, Steffen Eger, Qixiang Fang, Mingqi Gao, Albert Gatt, Dimitra Gkatzia, Javier González-Corbelle, Dirk Hovy, Manuela Hürlimann, Takumi Ito, John D. Kelleher, Filip Klubicka, Emiel Krahmer, Huiyuan Lai, Chris van der Lee, Yiru Li, Saad Mahamood, Margot Mieskes, Emiel van Miltenburg, Pablo Mosteiro, Malvina Nissim, Natalie Parde, Ondřej Plátek, Verena Rieser, Jie Ruan, Joel Tetreault, Antonio Toral, Xiaojun Wan, Leo Wanner, Lewis Watson, and

Diyi Yang. Missing information, unresponsive authors, experimental flaws: The impossibility of assessing the reproducibility of previous human evaluations in nlp, 2023. URL https://arxiv.org/abs/2305.01633.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. All that's 'human' is not gold: Evaluating human evaluation of generated text. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7282–7296, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.565. URL https://aclanthology.org/2021.acl-long.565/.

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

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire, 2023. URL https://arxiv.org/abs/2302.04166.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-star: Training verifiers for self-taught reasoners, 2024. URL https://arxiv.org/abs/2402.06457.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

OpenAI. Hello gpt-4o. 2024a. Available at: https://openai.com/index/hello-gpt-4o/.

OpenAI. Openai o1-mini. 2024b. Available at: https://openai.com/index/introducing-openai-o1-preview/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. Learning to plan reason for evaluation with thinking-llm-as-a-judge, 2025. URL https://arxiv.org/abs/2501.18099.

Sber. Yandexgpt 4. 2024. Available at: https://giga.chat/.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning, 2024. URL https://arxiv.org/abs/2409.12183.

T-bank. T-lite and t-pro - open russian-language open source models with 7 and 32 billion parameters. 2024. Available at: https://habr.com/ru/companies/tbank/articles/865582/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Tianhao Wu, Janice Lan, Weizhe Yuan, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. Thinking llms: General instruction following with thought generation, 2024. URL https://arxiv.org/abs/2410.10630.

Yandex. Yandexgpt 4. 2024. Available at: https://ya.ru/ai/gpt-4/.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020. URL https://arxiv.org/abs/1904.09675.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. Judgelm: Fine-tuned large language models are scalable judges, 2023. URL https://arxiv.org/abs/2310.17631.

A Appendix

The team of human annotators consists of 21 Russian-speaking individuals, comprising 14 women and 7 men. The experts' ages range from 21 to 42 years, with a median age of 29. Eleven members have a higher education degree in linguistics, six have a degree in translation, and two have a degree in philology.

Benchmark split	**Experts 1,2**	**Experts 2,3**	**Experts 1,3**
0-599	0.796	0.545	0.654
600-1199	0.830	0.741	0.547
1200-1799	0.754	0.804	0.604
1800-2396	0.669	0.706	0.823

Table 7: Fleiss' kappa measure between each pair of annotators for each of the four splits of the benchmark.

112: 1	224 : 2	334: 3	444: 4
122 : 2	133 : 1	344: 4	123: 3
113 : 1	144 : 4	111: 1	124: 4
114 : 1	233 : 2	222: 2	134: 1
223 : 2	244: 4	333: 3	234: 2

Table 8: **Ensembling strategy for judge verdicts.** The diagram illustrates how three individual judgments are mapped into a single final verdict.

Generation parameters	**value**
max_tokens	1024
temperature	0.0
top_p	0.1
frequency_penalty	1.0
vllm version	0.6.4

Table 9: **Parameters used for obtaining generations from LLMs-as-judges**. VLLM was used for inferencing open models from huggingface.

B Appendix

B.1 Positional bias metrics

Table 2 presents a study of LLM judges for positional bias. We perform two measurements for each evaluator model- without models' answers swap and with - and calculate the PCC of aggregated values with manual annotation. We also calculate the number of matching verdicts (**accuracy**) and its difference between swaps (Δ), while also introducing metrics **PBias@AB**, **Con@ABCD** and **PCon@AB** that indicate the presence of bias in the evaluator models.

$$\textbf{PBias@AB} = \sum_{\text{swap}=\{0,1\}} \sum_{BM} \mathbb{1}(J = \mathbf{A} | J = \mathbf{A} \vee \mathbf{B}) - 1,$$

where *BM* represents all samples from the benchmark, J is the judge's verdict, and swap $= \{0,1\}$ refers to the order of the test model answers in the prompt ($\{C,B\}$ and $\{B,C\}$ respectively). **PBias@AB** is from the interval $(-1,1)$, where the absolute value indicates the magnitude of the positional bias, and the sign indicates whether the positional bias is direct or reverse. The closer the value is to zero, the more unbiased the model is.

$$\textbf{Con@ABCD} = \sum_{BM} \mathbb{1}(J_{\text{swap}=0} = J_{\text{swap}=1}).$$

Con@ABCD shows the consistency of the model's answers without swap and with swap — it indicates the proportion of matching answers given the different order of model responses.

C Appendix

C.1 Dataset

Tell me what maternity capital is and what it can be spent on.

What is traditionally used to cook borscht?

Examples from a benchmark translated into English. Some of the questions involve aspects of Russian culture or reflect Russian realities.

D Appendix

D.1 Prompt I

prompt:

```
Please act as an objective and strict judge, evaluating the responses of
    two AI assistants to the user's question below. Select the assistant
    that adheres to the user's instructions and responds to the question
    with higher quality. Your evaluation must rigorously consider factors
     such as helpfulness, relevance, accuracy, depth, creativity, and
    detail of the responses. Avoid any biases based on the position or
    order of responses to ensure an unbiased decision. The length of
    responses should not affect your evaluation. Maintain objectivity and
     neutrality towards assistant names. Output exactly one of the
    following symbols: A, B, C, or D. Use the following criteria strictly
    :

- Output 'A' if the first response is notably better.
- Output 'B' if the second response is notably better.
- Output 'C' if both responses are equally good.
- Output 'D' if both responses are equally inadequate.

IMPORTANT!! Be strict in your assessment: if there are errors in both
    responses, even minor ones, treat responses as inadequate and output
    'D'.
```

instruction:

```
Question: {question}

First Response: {answer1}
Second Response: {answer2}
Answer:
```

D.2 Prompt II

prompt:

```
Please act as an objective and strict judge, evaluating the responses of
    two AI assistants to the user's question based on the strict quality
    standards. Assess each response against the following criteria to
    determine which assistant provides the best overall answer. Your
    evaluation must be thorough, reasoned, and step-by-step.


Evaluation Criteria:
1. Safety: The response must not pose any harm, directly or indirectly.
    Even if it meets all other criteria, any unsafe content renders the
    response inadequate.
2. Ethics: Responses must be free from rudeness, offensiveness, bias, or
    evaluative language. Any ethical violations automatically make the
    response unacceptable.
3. Truthfulness: The response must be factually accurate. Any false or
    unverifiable claims should be considered critical errors.
4. Adequacy to the Request: The response must fully address the user's
    query without unnecessary deviations. It should adhere to specific
    instructions such as style, tone, and language. Failure to meet these
     requirements makes the response inadequate.
5. Completeness: The response should cover all relevant aspects of the
    query in a single reply, avoiding the need for follow-ups or
    additional clarifications.
```

```
6. Style: The response should be clear, well-structured, and
    professionally written. Poor readability, incoherence, or
    inappropriate style should result in a lower evaluation.

Evaluation Method:
- Maintain objectivity, avoiding bias based on response position or
    length.
- Penalize any response that fails to meet the standards outlined above.
- Explicitly describe your train of thought for each criterion,
    explaining why one response is better than the other or if they are
    equal.

Decision Rules:
- Output '[[A]]' if the first response is clearly superior across all
    criteria.
- Output '[[B]]' if the second response is clearly superior across all
    criteria.
- Output '[[C]]' if both responses are equally good and fully meet the
    criteria.
- Output '[[D]]' if either response contains any factual inaccuracies,
    ethical violations, safety concerns, or fails to meet the user's
    request in any way, even minor issues.

IMPORTANT:
- Be strict in your assessment - if both responses have any deficiencies,
     even minor ones, output '[[D]]'.
- Focus purely on content quality based on the given evaluation criteria
    }.
- After presenting your
    reasoning, provide the final decision enclosed in double brackets to
    ensure proper parsing, for example: [[A]], [[B]], [[C]] or [[D]].


instruction:

Question: {question}

First Response: {answer1}
Second Response: {answer2}
Answer:
```

D.3 Multi-Agent Debate

```
{AGENT-1}
instruction:

You are a first agent. Please act as an objective and strict judge,
    evaluating the responses of two AI assistants to the user's question
    below. Select the assistant that adheres to the user's instructions
    and responds to the questionwith higher quality. Your evaluation must
     rigorously consider factors such as helpfulness, relevance, accuracy
    , depth, creativity, and detail of the responses. Avoid any biases
    based on the position or order of responses to ensure an unbiased
    decision. The length of responses should not affect your evaluation.
    Maintain objectivity and neutrality towards assistant names. Output
    exactly oneof the following symbols: A, B, C, or D. Use the following
     criteria strictly:

- Output 'A' if the first response is notably better.
- Output 'B' if the second response is notably better.
- Output 'C' if both responses are equally good.
- Output 'D' if both responses are equally inadequate.
```

```
IMPORTANT!! Be strict in your assessment: if there are errors in both
    responses, even minor ones, treat responses as inadequate and output
    'D'.

Question: {question}

First Response: {answer1}
Second Response: {answer2}
Answer:


{AGENT-2}
instruction:
You are the second agent. You always disagree with the first
    agent. Provide your reasons and verdict.


{AGENT-JUDGE}
instruction:
You are the judge agent. Evaluate both agents answers and decide which
    one
is correct and make the final verdict.

Please format your final verdict as follows: [[Selected Answer]]
```