Towards Better Multilingual Side-by-Side LLM Evaluation

Anonymous ACL submission

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 Russian, Arabic, 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 models in different languages. 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 Works

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 (LLMs) (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 each of the SbS setups: Russian, Arabic, and code generation. The first dataset contains 2396 instructions, each written manually, taking into account the specifics of Russian culture and patterns characteristic of Russian users. The instructions are classified by task type, including question answering, creative writing, information extraction, summarization, classification, and chat-based questions. For those instructions requiring factual accuracy in responses, we also gathered factual references. Later, these references help judges make correct decisions based on the provided knowledge sources.

For the second setup, we asked native Arabic speakers to translate and adapt the instructions from this dataset to fit Arabic culture.

We also manually 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 utilize a Side-by-Side method in which human experts and large language models serve as judges. We generate responses to instructions from the collected dataset for two models — the baseline and the candidate. After that the selected judge compares pairs of responses and delivers a verdict.

In many studies using a similar evaluation method, the judge performs a binary classification, indicating either that the response of the first model is better or the second one is. We decided to expand the set of options, implying that both responses could be 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 instructions for the human experts 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. In cases where the verdicts for a particular pair of responses did not match, they are sent for reassessment with discussion among the experts.

Background information about the team of experts, 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 succinctly 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.

Factology

We include factual information in the prompt and ask the model to rely on a knowledge source when evaluating responses where necessary. Factual references were collected from Wikipedia.

Multi-agent approach

The reasoning of the evaluator's language model really 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. Authors provide an example of a multi-agent approach that avoids this problem. To do this, agent-1 expresses his 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 his assessment regarding a pair of proposals, and agent-2 either agree or disagree with agent-1. Next, the agent-judge makes his verdict based on the two previous verdicts.

2. **Hard**. Agent-1 makes his assessment regarding a couple of proposals, and agent-2 always disagrees with agent-1. After that the agent-judge makes his verdict based on the two previous verdicts.

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

4 Experiments

For our experiments in Russian we select Qwen2.5-32B-Instruct (Yang et al., 2024) fine-tuned on the

	Judge	Parameters	SBS results					APCC	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	-0.229	0.507	link	(Dubey et al., 2024)
	llama3.3-70b	70B	34.1	39.0	26.0	1.0	0.0	0.041	0.595	link	(Dubey et al., 2024)
	gpt-4o	-	17.6	13.6	46.0	22.7	0.0	-0.144	0.495	-	(OpenAI, 2024a)
	o1-mini	-	32.7	40.3	18.1	9.0	0.0	0.165	-	-	(OpenAI, 2024b)
	gpt4	-	23.8	24.0	51.3	0.8	0.1	-0.081	**0.642**	-	(Achiam et al., 2023)
	deepseek-r1-dst.	70B	24.2	33.5	31.9	7.3	3.1	0.227	0.640	link	(Guo et al., 2025)
	deepseek-v3	671B (37B)	11.5	47.7	34.9	1.0	4.8	0.624	0.570	link	(Liu et al., 2024)
	claude sonnet	175B	13.6	9.1	71.1	1.0	5.2	-0.122	0.427	-	(Anthropic, 2024)
	claude opus	137B	23.3	35.0	20.2	21.4	0.2	**0.681**	0.602	-	(Anthropic, 2024)
Russian	T-lite-it-1.0	7.6B	18.4	26.7	39.5	15.4	0.0	0.238	0.139	link	(T-bank, 2024)
	T-pro-it-1.0	32.8B	40.6	44.6	4.1	10.7	0.5	0.070	0.397	link	(T-bank, 2024)
	GigaChat-Max	70-100B	24.9	38.9	30.7	0.9	4.6	0.265	-	-	(Sber, 2024)
	YandexGPT	-	29.8	43.2	7.4	0.2	19.5	0.231	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 presented 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. For the LLM evaluators, the average value for each verdict across two benchmark runs is provided. Additionally, we include the **APCC** with expert assessments, where all verdict are aggregated by verdict class and **MPCC**, where we use a sliding window to go through the verdicts and calculate the median for each batch.

Russian language as the candidate model and GPT-4o (OpenAI, 2024a) as the baseline model. For the Arabic language we use Llama-3.0-70b (Dubey et al., 2024) fine-tuned on the Arabic language as the candidate 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 Pearson correlation coefficient (PCC) between each pair of annotators for each of the four splits of the dataset. The PCC ranges from **0.667** to **0.937** depending on the dataset split. We conclude that if the correlation 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.

4.2 Analysis of automatic evaluation

We select a range of models of different sizes for the evaluation, including both open-source and commercial models. The results of the manual and automatic evaluation in Russian and Arabic can be found in Table 1 and Table 5 respectively. We measure the correlation between the assessments of the models and the experts using the following metrics.

Aggregated Pearson Correlation Coefficient (APCC). We count how many verdicts fall into each class A/B/C/D and calculate the correlation between LLM-as-judges and experts assessments based on these four values. While calculating this metric, we lose a lot of information about individual verdicts, but we can estimate how close the model is to the experts in delivering a final verdict for the entire benchmark.

Median Pearson Correlation Coefficient (MPCC). We apply a sliding window with a size of 10 and a stride of 5 across all verdicts from the benchmark. For teach batch we calculate the median using formula: $Median = \frac{\sum \mathbf{A} + \sum \mathbf{C}}{\sum \mathbf{A} + \sum \mathbf{B} + 2 \cdot \sum \mathbf{C}}$. We obtain a set of medians for the expert and model

Judge model	MPCC		PCon@AB
	Cons.	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.

Prompt	SBS results				APCC	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	0.226	0.589	**0.500**
II	14.5	26.0	44.1	15.4	**0.277**	0.623	0.361
II-fact	14.5	26.1	44.2	15.1	0.275	0.606	0.355
II-reason	17.9	29.4	43.9	8.4	0.226	**0.639**	0.412
II-fact+reason	17.9	28.9	43.9	8.9	0.218	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				APCC	MPCC	PCon@AB
	A	B	C	D			
experts	7.6	41.4	23.7	27.2			
I	13.1	20.3	63.8	2.8	**0.041**	0.573	0.573
II	9.0	11.5	57.5	21.9	0.014	0.505	0.146
II-fact	9.0	11.5	57.6	21.9	0.013	0.504	0.145
II-reason	31.5	31.6	32.0	4.7	-0.089	**0.653**	**0.499**
II-fact+reason	30.3	31.7	33.5	4.2	-0.051	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.

verdicts and calculate the PCC between them. This method retains almost all information for all verdicts but imposes a linear relationship between verdict classes, which may be somewhat incorrect to establish.

Both metrics 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 APCC and MPCC 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 APCC and BPCC we conclude that Claude Opus and Deepseek-r1-dst - distillation of Deepseek-r1 into Llama-70B - 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 10 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.

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

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.

PCon@AB, **MPCC-Consistency** and **MPCC-Δ** do not rely on manual annotation,

Judge model	verdicts				APCC	MPCC	PCon@AB
	A	B	C	D			
manual	27.6	14.8	46.4	11.2			
llama3.1-405b	21.6	13.6	37.8	26.8	0.735	**0.390**	**0.464**
llama3.3-70b	15.7	7.8	39.8	36.5	0.413	0.337	0.306
gpt-4o	2.4	3.1	20.4	74.1	-0.379	-0.153	0.112
deepseek-r1-dst.	23.1	15.4	36.8	24.5	**0.835**	0.369	0.354
deepseek-v3	4.9	3.0	19.4	72.7	-0.390	0.280	N/A
claude sonnet	21.7	16.4	21.3	40.5	-0.423	0.259	0.448
claude opus	22.0	15.3	32.3	30.1	0.485	0.347	0.168

Table 5: **Comparative analysis of LLMs as judges for SbS Evaluation in Arabic.** 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.

Judge model	verdicts				APCC	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	0.819	0.427	0.364
llama3.3-70b	41.3	9.2	17.3	32.2	0.898	0.432	**0.439**
deepseek-r1-dst.	35.4	19.3	25.6	19.7	0.343	**0.460**	0.424
deepseek-v3	24.4	6.8	4.1	64.8	0.818	0.089	0.241
claude sonnet	17.4	0.8	6.4	75.4	0.797	0.312	0.098
claude opus	25.4	11.3	27.1	36.3	**0.903**	0.392	0.089

Table 6: **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.

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. However, modifying prompt II by adding factual information and reasoning cause the correlation to decrease. This is expected when requesting the model to reason - Deepseek-r1-distill-llama is already trained to do reasoning, therefore the additional step is redundant.

The patterns do not hold for the model Llama3.3-70b, as can be seen in 4. Adding factual information and reasoning for this model significantly increases the correlation with experts. It is noteworthy that 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 model more parameters (Claude opus).

4.5 SbS evaluation for Arabic

We conduct similar studies on the Arabic version of our benchmark with Prompt II for a range of evaluator models.

From the Table 5, it follows that models Deepseek-r1-distill-llama and Llama3.1-405b show the best correlation with human judgment, while models Llama3.1-405b and Claude Sonnet are the least prone to positional bias.

4.6 SbS evaluation for code generation

In addition to SbS assessments in two languages, we also conduct similar experiments for models intended for code generation, using Prompt II for LLM evaluators. From the Table 6 we see that Llama models have generally the best performance in terms of the APCC and MPCC metrics, while Claude Opus and Deepseek-r1-dst can still be considered as strong options.

From Tables 1, 5 and 6, we conclude that regardless of the language in which the SbS task is performed, there is an issue with positional bias in

LLMs-as-judges. For each language (and for the programming languages), there are different families of models that perform best at evaluating texts in that language, and there isn't a single model that excels in everything.

Judge model	method	verdicts					PCC
		A	B	C	D	E	Mean
manual		7.6	41.4	23.7	27.2	0.0	
T-pro-it-1.0	base	40.6	44.6	4.1	10.7	0.5	0.070
	soft	37.9	45.6	9.5	6.0	1.0	**0.118**
	hard	36.1	36.7	14.82	3.6	8.78	-0.041

Table 7: **Multi-agent approach.** Measurement results for the T-pro-it model as a judge. We managed to increase the correlation with manual annotation using the soft approach.

4.7 Analysis of multi-agent approach

As can be seen from the Table 7, the hard method shows weak correlation with human markup. This is most likely due to the fact that T-pro-it already gives a response similar to the markup in the first iteration, and the other agents (the agent who disagrees or agrees with the statement and the agent judge) confuse the verdict of the first agent.

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 factual information and reasoning on the judging model's capabilities and its influence on correlation with manual annotations.

Limitations

We acknowledge that there are other strong open and proprietary models that we do not consider as evaluators for the SbS task. We also do not research on biases in expert assessments; there are patterns in the candidate and baseline models' responses inherent to these models that could lead experts to guess which model produced a given response. Additionally, humans can also have positional biases.

References

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

Anthropic. 2024. Introducing the next generation of claude. 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. 2023. Missing information, unresponsive authors, experimental flaws: The impossibility of assessing the reproducibility of previous human evaluations in nlp. *Preprint*, arXiv: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. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *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)*, pages 7282–7296, Online. Association for Computational Linguistics.

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

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

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

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-star: Training verifiers for self-taught reasoners. *Preprint*, arXiv:2402.06457.

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

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

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

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

OpenAI. 2024b. Openai o1-mini. 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. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. 2025. Learning to plan reason for evaluation with thinking-llm-as-a-judge. *Preprint*, arXiv:2501.18099.

Sber. 2024. Yandexgpt 4. 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. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *Preprint*, arXiv:2409.12183.

T-bank. 2024. T-lite and t-pro - open russian-language open source models with 7 and 32 billion parameters. 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. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Tianhao Wu, Janice Lan, Weizhe Yuan, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024. Thinking llms: General instruction following with thought generation. *Preprint*, arXiv:2410.10630.

Yandex. 2024. Yandexgpt 4. 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. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv: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. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

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

A Appendix

The team of human experts 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	0.989	0.809	0.844
600-1199	0.873	0.751	**0.376**
1200-1799	0.914	**0.932**	0.966
1800-2396	0.985	0.866	0.939

Table 8: Pearson correlation coefficient between each pair of annotators for each of the four splits of the benchmark.

B Appendix

B.1 Positional bias metrics

Table 10 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.

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.

$$\textbf{PBias@AB} = \sum_{\text{swap}=\{0,1\}} \sum_{BM} \mathbb{1}(J = \mathbf{A} | J = \mathbf{A} \vee \mathbf{B}) - 1, \quad (2)$$

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.

$$\textbf{Con@ABCD} = \sum_{BM} \mathbb{1}(J_{\text{swap}=0} = J_{\text{swap}=1}). \quad (3)$$

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 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,
```

Judge model	swap	verdicts					PCC		PBias@AB	Con@ABCD	PCon@AB
		A	B	C	D	E		Mean			
miqu	{C, B}	36.7	14.8	43.5	3.3	1.8	0.008	0.367	0.455	0.413	0.233
	{B, C}	15.5	46.0	34.9	3.6	0.0	0.726				
llama3.1-405b	{C, B}	43.9	19.9	35.5	0.7	0.0	0.016	0.272	0.274	0.411	0.336
	{B, C}	29.3	42.0	27.8	1.0	0.0	0.528				
llama3.3-70b	{C, B}	23.3	39.0	36.6	1.0	0.0	0.560	0.405	-0.173	0.542	0.476
	{B, C}	44.8	39.0	15.3	0.8	0.0	0.250				
gpt-4o	{C, B}	22.3	7.0	48.4	22.3	0.0	0.176	0.376	0.359	0.606	0.329
	{B, C}	12.9	20.3	43.7	23.1	0.0	0.576				
gpt4	{C, B}	25.8	9.1	63.9	1.1	0.1	0.068	0.314	0.373	0.547	0.339
	{B, C}	21.9	38.9	38.7	0.5	0.0	0.560				
deepseek-r1-distill-llama	{C, B}	25.1	30.4	35.0	7.3	2.3	0.503	0.564	0.074	0.574	0.500
	{B, C}	23.3	36.6	28.8	7.3	4.1	0.625				
deepseek-v3	{C, B}	0.7	64.3	32.2	0.8	2.2	0.784	0.558	-0.477	0.376	0.271
	{B, C}	22.3	31.2	37.7	1.4	7.4	0.392				
claude sonnet	{C, B}	7.8	14.9	72.1	0.7	4.4	0.209	0.100	-0.480	0.631	0.106
	{B, C}	19.5	3.3	70.0	1.3	5.9	-0.009				
claude opus	{C, B}	19.3	41.7	23.1	15.6	0.3	0.866	0.766	-0.170	0.508	0.431
	{B, C}	27.3	28.2	17.2	27.1	0.2	0.666				
T-lite-it-1.0	{C, B}	6.1	36.1	46.7	11.2	0.0	0.716	0.504	-0.512	0.397	0.122
	{B, C}	30.8	17.2	32.3	19.6	0.0	0.292				
T-pro-it-1.0	{C, B}	56.1	28.4	3.9	11.1	0.5	-0.01	0.336	0.386	0.454	0.400
	{B, C}	24.0	60.8	4.3	10.3	0.6	0.674				
GigaChat-Max	{C, B}	24.9	38.9	30.7	0.9	4.6	0.25	0.25			
	{B, C}										
YandexGPT	{C, B}	34.9	38.9	8.7	0.3	17.2	0.084	0.166	0.140	0.480	0.409
	{B, C}	24.6	47.4	6.0	0.2	21.8	0.247				

Table 10: **Comparative analysis of evaluator scores with and without swap of models' answers.** Metrics **Accuracy-Δ**, **PBias@AB**, **Con@ABCD** and **PCon@AB** indicate the presence of positional bias among LLM-evaluators. The values of metrics **Accuracy-Δ** and **PBias@AB** being close to zero indicate an absence of positional bias, while the values of metrics **Con@ABCD** and **PCon@AB**, on the contrary — the closer they are to one, the more consistent the model is when the positions of prompt answers are changed.

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:

C.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 provided factual information and 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 and aligned with the provided factual information. 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.
- Compare each response to the factual information and the above criteria.
- 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 factual information and 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:

Knowledge source: {}
Question: {question}

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

C.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
    desagree 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]]
```