

Re-evaluating Polish LLMs in RAG: The Evolution of Bielik and PLLuM and the Limits of Open-Source Evaluators

Marcin Szewczyk, Krzysztof Maksim, Tymon Pluciński, Szymon Bartanowicz, Krzysztof Jassem
0009-0003-9825-258X, 0009-0003-4094-1905, 0009-0002-2623-0800, 0009-0002-0385-2332, 0000-0001-7122-9206
Adam Mickiewicz University, Poznań, Poland
E-mail: {marsze19, krzimak2, tymplu, szybar4}@st.amu.edu.pl, krzysztof.jassem@amu.edu.pl

Abstract—This paper extends a prior evaluation of a RAG system in an industrial environment. We measure generational progress by comparing Polish LLMs against their latest iterations and analyze the "LLM-as-a-Judge" paradigm by evaluating the evaluators themselves. The evaluation was conducted using two distinct classes of open-weight judges: small (under 10B parameters) and large (over 20B parameters), employing both quantitative scoring and order-controlled pairwise A/B testing. The results reveal that small judges exhibit severe position bias, making them unreliable. Furthermore, they confirm that *Bielik-11B-v3.0-Instruct* consistently outperforms *PLLuM-12B-nc-chat-250715* in accuracy and relevance. We conclude that small judges (under 10B) are unreliable for local RAG evaluation, while larger judges (over 20B) are more robust though not entirely free of order effects. In this technical documentation setting, Bielik remains the stronger of the two evaluated Polish models.

Index Terms—Retrieval-Augmented Generation (RAG), large language models (LLMs), LLM-as-a-Judge, evaluation

I. INTRODUCTION

THE RAPID ADVANCEMENT of Large Language Models (LLMs) has revolutionized natural language processing, yet their utility in specialized domains remains limited by the risk of hallucinations and the lack of access to up to date proprietary information. Retrieval-Augmented Generation (RAG) has emerged as the standard architectural solution to this problem, grounding model outputs in relevant documents retrieved from a controlled knowledge base. While English language RAG systems have been extensively documented, the development of robust RAG pipelines for the Polish language presents unique challenges due to linguistic complexity and the limited availability of high-quality, instruction tuned open source models [1], [2].

Building directly on the findings presented in [3], this study extends its evaluation methodology. In that baseline paper, the authors designed an optimized Retrieval-Augmented Generation (RAG) pipeline tailored for the technical documentation of a low-code business platform. They successfully identified *OrlikB/KartonBERT-USE-base-v1* [4] as the optimal retrieval engine and provided empirical evidence that *Bielik-11B-v2.3-Instruct* outperformed *PLLuM-12B-nc-chat*. Nevertheless, the Polish LLM ecosystem is subject to rapid, continuous development. The recent introduction of *Bielik-11B-v3.0-Instruct* and *PLLuM-12B-nc-chat-250715* motivates a re-evaluation to

assess whether the generative performance gap between these competing models has expanded or narrowed.

Furthermore, the methodology used to evaluate these models, predominantly the "LLM-as-a-Judge" paradigm requires closer scrutiny. While previous studies often relied on large, proprietary models or established open source weights as evaluators, the impact of the judge's model size on the reliability of the results lacks comprehensive analysis in the context of the Polish language. Biases such as "leniency" (overly positive scoring) and "position bias" (favoring the first presented response in A/B testing) can significantly distort findings, especially when using smaller, more efficient models for local evaluation.

Our research methodology is structured into two distinct phases. Initially, we utilized the evaluation data from the original study (baseline) to analyze the performance and reliability of "small" judges (under 10B parameters) in comparison to established benchmarks. In the second phase, we evaluated the latest versions of the leading Polish models—*Bielik-11B-v3.0-Instruct* and *PLLuM-12B-nc-chat-250715* to measure chronological progress and verify the evaluative stability of the "large" judges (over 20B parameters). By examining whether these high-capacity evaluators maintain consistent scoring trends when faced with improved model outputs, we aim to define the necessary scale for reliable, locally hosted model assessment and provide an updated analysis of the current state of Polish generative AI in this RAG scenario.

II. RELATED WORK

A. RAG Evaluation and Retrieval Optimization

Domain-specific benchmarks for Retrieval-Augmented Generation (RAG) emphasize the necessity of explicitly measuring both the retrieval accuracy (e.g., using NDCG) and the grounded generation quality [5], [6]. In the context of English language systems, dense BERT-based retrievers have long been established as superior to sparse methods [7], with hybrid search becoming the industry standard. For the Polish language, the baseline study [3] systematically evaluated these methods and concluded that pure vector search utilizing the Polish specific *OrlikB/KartonBERT-USE-base-v1* encoder outperforms hybrid configurations. By adopting this optimized retrieval baseline, our study isolates the generative

phase, ensuring that any variance in performance is strictly attributable to the LLMs rather than the retrieval mechanics.

B. The Evolution of Polish LLMs

The sector of Polish LLM models tailored for instructional tasks is growing at a rapid pace. The original *Bielik-11B-v2.3-Instruct* technical report [2] highlighted significant improvements in Polish text comprehension and grounded generation. Similar initiatives, such as the government supported PLLuM models [8], have provided comparable benchmarks for 12 billion parameter foundation and instruction models. While previous evaluations favored the v2.3 iteration of Bielik, the recent deployment of *Bielik-11B-v3.0-Instruct* introduces enhanced architectural optimizations. The rapid iteration cycle of these models necessitates continuous benchmarking in real world scenarios to accurately track state-of-the-art performance.

C. LLM-as-a-Judge: Bias and Model Scale Limitations

The "LLM-as-a-Judge" paradigm, introduced as a scalable alternative to human evaluation, utilizes advanced language models to score or compare the outputs of other models [9]. However, this methodology is fraught with well-documented vulnerabilities. Research by [10] and [11] identifies multiple cognitive biases inherent in model based evaluation, including "leniency" (a tendency to assign unjustifiably high scores) and "position bias" or "order effects" [12], where evaluators disproportionately favor the first response in A/B testing formats. Crucially, while the original "LLM-as-a-Judge" framework was validated using big models, there is a growing trend toward using smaller, open weight models locally for cost effective evaluation. The current literature lacks a clear agreement on the minimum parameter threshold required for an open source model to effectively resist these biases. While the original study used older, large-scale open evaluators with reversed A/B protocols to mitigate order effects, our work directly investigates the correlation between evaluator model scale (e.g., <10B vs. >20B parameters) and the severity of positional bias, addressing a critical gap in modern LLM evaluation methodology.

III. EXPERIMENTS

A. Baseline and Retrieval Setup

To ensure strict comparability with the baseline study, we utilized the identical golden set comprising 146 real world technical queries. The retrieval pipeline was also locked to the previously optimized configuration. Specifically, document chunks were retrieved utilizing the Polish tailored *OrlikB/KartonBERT-USE-base-v1* embedding model combined with a pure vector based search strategy. By keeping all elements of the study constant, we effectively isolated the generation phase, guaranteeing that any observed performance deltas are strictly attributable to the generative language models rather than the retrieval mechanics.

B. Generative Models

The generation component of our RAG pipeline was powered by the latest iterations of two leading Polish instruction tuned models. We selected *Bielik-11B-v3.0-Instruct*, which introduces substantial architectural and alignment refinements over the v2.3 version tested in prior benchmarks. Against this, we benchmarked *PLLuM-12B-nc-chat-250715*, the most recent 12B scale checkpoint that represents a public administration effort in Polish NLP. Both models operate within a comparable parameter footprint, making them ideal candidates for a direct head to head evaluation.

C. Evaluator Configurations

A central part of our experimental design is the careful validation we applied to the evaluator models themselves. While the "LLM-as-a-Judge" methodology is effective, it relies heavily on a model's ability to stay impartial. However, using massive proprietary models for constant local testing is often too expensive or impractical. To test the limits of what open-source evaluators can do, we split our judges into two tiers based on their size. The "Small Judges" group included models under 10 billion parameters:

- Llama-3.1-8B-Instruct,
- Qwen2.5-7B,
- Qwen3.5-9B.

The "Large Judges" group, covered models over 20 billion parameters:

- Mistral-Small-3.2-24B-Instruct-2506,
- Qwen3-30B-A3B-Instruct-2507,
- gpt-oss-20b.

This division allows us to see how model size directly impacts reliability and the cognitive biases that typically affect automated scoring.

D. Evaluator Protocol

The evaluation process was divided into a quantitative scoring phase and a comparative preference assessment phase. Initially, each judge model was instructed to independently rate the generated responses on a discrete scale of 1 to 5, evaluating both the factual precision and contextual completeness based exclusively on the retrieval chunks provided.

Following this, we conducted exhaustive pairwise A/B tests. To best detect and mitigate the well-documented position bias, where an evaluator favors the first presented option, we implemented a reversed B/A testing protocol. Every pair of answers was evaluated twice, with their positions in the prompt swapped. This approach ensured that a model's preference was genuinely based on output quality rather than prompt order.

E. Implementation Details

All experiments were conducted on a dedicated computing node equipped with NVIDIA A100 GPU (80GB VRAM). The models were deployed using the *vLLM* framework [13] for optimized inference, orchestrated via *Docker* containers. The connection between the evaluation scripts and the models

```

1 prompt = f"""
2 Question: {question}
3 Answer: {model_answer}
4 Context: {full_context}
5 Evaluate the answer (Answer) to the
  → question (Question) based on the
  → provided context (Context).
6 Scoring instructions:
7 Rate the answer on a scale of 1-5,
  → where:
8 {scoring_instructions}
9 """

```

Code listing 1: Prompt structure used for computing the average score

```

1 prompt = f"""
2 Compare the two answers (Answer A and
  → Answer B) and
3 choose the one that is more complete
  → and accurate
4 based on the provided context.
5 Question: {question}
6 Answer A: {answer_a}
7 Answer B: {answer_b}
8 Context: {context}
9 Which answer is better? Provide a
  → brief explanation and
10 then output 'Best Answer: A' or 'Best
  → Answer: B'.
11 """

```

Code listing 2: Prompt structure used for pairwise A/B testing

was established using an *OpenAI-compatible API*, ensuring low latency and consistent request handling across different evaluator scales.

IV. RESULTS

A. Score Evaluation Outcome

To evaluate the reliability of small open-source judges, we first compared their scoring behavior against the baseline results of large evaluators [3] using the older model iterations (*Bielik-11B-v2.3-Instruct* and *PLLuM-12B-nc-chat*). As presented in Table I, the baseline large judges consistently identified a distinct performance gap, with *gpt-oss-20b* awarding Bielik an average score of 4.192 compared to PLLuM’s 3.466. In contrast, models from the "Small Judges" tier exhibited a severe leniency bias and failed to distinguish between the models. For instance, *Llama-3.1-8B-Instruct* assigned nearly identical, inflated scores to both PLLuM (4.705)

and Bielik (4.623). Similarly, *Qwen2.5-7B* followed a near-identical trend, awarding inflated scores of 4.603 to PLLuM and 4.548 to Bielik. A notable exception was *Qwen3.5-9B*, which was the only small model to identify the performance gap (4.555 for Bielik versus 3.774 for PLLuM).

B. A/B tests and Position Bias

The pairwise testing exposed a critical vulnerability in the evaluation pipeline, effectively disqualifying most sub-10B parameter models from acting as reliable automated judges. As shown in Tables II and III, models from the "Small Judges" tier exhibited a severe position bias. For example, *Llama-3.1-8B-Instruct* selected the first option in 67.8% of cases in the standard setup (Table II) and continued to favor the first position in 82.2% of cases in the reversed protocol (Table III). In both the standard A/B test and the reversed B/A protocol, these evaluators significantly favored "Answer A" regardless of the underlying semantic quality. Their inability to track the response across swapped positions reveals that models at this scale default to ordering patterns rather than conducting actual content evaluation.

However, a notable exception within the "Small Judges" tier is *Qwen3.5-9B*. Although it is more sensitive to position bias than the 20B+ evaluators, it shows more accurate results than other models in the same size category. It was the only small judge that preferred Bielik’s answers more often in both A/B and B/A tests, showing that its judgment was based more on content than on the order of responses. This suggests that being an effective judge is not only about parameter count; instead, model improvements in newer series like Qwen3.5 may partially compensate for smaller scale, making them a more reliable option for local evaluation than older models of similar size.

While such exceptions exist, the models in the "Large Judges" group are, on average, more consistent than the small tier, though their robustness is not uniform. Among the large evaluators, *gpt-oss-20b* demonstrated the highest resistance to position bias. It consistently preferred the superior model, maintaining a 77.4% preference in the standard setup (Table II) and 70.5% in the reversed protocol (Table III). While *Mistral-Small-3.2-24B-Instruct-2506* also identified the better model, its tracking accuracy was more sensitive to prompt order, dropping from 76.7% to 56.2% when positions were swapped. Moreover, *Qwen3-30B-A3B-Instruct-2507*, despite a strong 90.4% preference for Bielik in the A/B setup, reversed to favour PLLuM (63.7%) in the B/A setup, confirming that even some 20B+ judges remain sensitive to answer order. These results provide strong evidence that a parameter threshold of at least 20 billion reduces the likelihood of position bias, but this result still cannot be considered sufficient to overcome order effects and achieve reproducible content-based evaluations.

C. Generational Progress and Evolution

To address the rapid development of Polish LLMs, we compared the current iterations against their predecessors,

Table I
AVERAGE EVALUATION SCORES ON A 1–5 SCALE

Evaluator Model	<i>PLLuM-12B-nc-chat</i>	<i>Bielik-11B-v2.3-Instruct</i>
<i>Baseline Large Judges [3]</i>		
gpt-oss-20b	3.466	4.192
Mistral-Small-3.2-24B-Instruct-2506	4.219	4.548
Qwen3-30B-A3B-Instruct-2507	4.390	4.822
Average	4.025	4.521
<i>Small Judges</i>		
Llama-3.1-8B-Instruct	4.705	4.623
Qwen2.5-7B	4.603	4.548
Qwen3.5-9B	3.774	4.555
Average	4.361	4.575

Table II
A/B TEST RESULTS (A BEFORE B)

Evaluator Model	<i>PLLuM-12B-nc-chat</i> (A)	<i>Bielik-11B-v2.3-Instruct</i> (B)
<i>Baseline Large Judges [3]</i>		
gpt-oss-20b	33 (22.6%)	113 (77.4%)
Mistral-Small-3.2-24B-Instruct-2506	34 (23.3%)	112 (76.7%)
Qwen3-30B-A3B-Instruct-2507	14 (9.6%)	132 (90.4%)
Average	27.0 (18.5%)	119.0 (81.5%)
<i>Small Judges</i>		
Llama-3.1-8B-Instruct	99 (67.8%)	47 (32.2%)
Qwen2.5-7B	102 (69.9%)	44 (30.1%)
Qwen3.5-9B	22 (15.1%)	124 (84.9%)
Average	74.3 (50.9%)	71.7 (49.1%)

Table III
A/B TEST RESULTS (B BEFORE A)

Evaluator Model	<i>Bielik-11B-v2.3-Instruct</i> (B)	<i>PLLuM-12B-nc-chat</i> (A)
<i>Baseline Large Judges [3]</i>		
gpt-oss-20b	103 (70.5%)	43 (29.5%)
Mistral-Small-3.2-24B-Instruct-2506	82 (56.2%)	64 (43.8%)
Qwen3-30B-A3B-Instruct-2507	53 (36.3%)	93 (63.7%)
Average	79.3 (54.3%)	66.7 (45.7%)
<i>Small Judges</i>		
Llama-3.1-8B-Instruct	120 (82.2%)	26 (17.8%)
Qwen2.5-7B	106 (72.6%)	40 (27.4%)
Qwen3.5-9B	79 (54.1%)	67 (45.9%)
Average	101.7 (69.7%)	44.3 (30.3%)

relying exclusively on the stable "Large Judges" tier (Tables IV, V, and VI). The quantitative scoring demonstrates a clear generational improvement for both model families. The updated *PLLuM-12B-nc-chat-250715* model saw its overall mean score increase from 4.025 to 4.192. Similarly, *Bielik-11B-v3.0-Instruct* advanced from an already strong baseline of 4.521 (v2.3) to 4.591 (v3.0). The pairwise A/B testing further corroborates this progress, with large judges consistently registering a preference for the newer checkpoints over the older ones. These findings validate the rapid evolution of the Polish open source ecosystem, while confirming that Bielik maintains its position as the state-of-the-art for local RAG deployments.

V. CONCLUSIONS

This study extended the evaluation methodology established in the baseline study [3] to assess the latest generation of Polish language models and rigorously test the "LLM-as-a-Judge" methodology. Our results confirm the continued advantage of the Bielik model family in the evaluated industrial RAG scenario. In a direct comparison of the latest

Table IV
OLD VS NEW MODEL ITERATIONS: AVERAGE EVALUATION SCORES.
NOTE: "OLD" ITERATIONS DENOTE BASELINE RESULTS RETRIEVED FROM [3], WHEREAS "NEW" ITERATIONS REPRESENT FINDINGS FROM OUR CURRENT STUDY.

Large Evaluator	PLLuM-12B		Bielik-11B	
	Old	New	Old	New
gpt-oss-20b	3.466	3.897	4.192	4.240
Mistral-Small-3.2-24B-Instruct-2506	4.219	4.199	4.548	4.753
Qwen3-30B-A3B-Instruct-2507	4.390	4.479	4.822	4.781
Overall Mean	4.025	4.192	4.521	4.591

Table V
OLD VS NEW MODEL ITERATIONS: A/B TESTS RESULTS (A BEFORE B).
NOTE: "OLD" ITERATIONS DENOTE BASELINE RESULTS RETRIEVED FROM [3], WHEREAS "NEW" ITERATIONS REPRESENT FINDINGS FROM OUR CURRENT STUDY.

Large Evaluator	PLLuM-12B (A)		Bielik-11B (B)	
	Old	New	Old	New
gpt-oss-20b	33	15	113	131
Mistral-Small-3.2-24B-Instruct-2506	34	40	112	106
Qwen3-30B-A3B-Instruct-2507	14	3	132	143
Overall Mean	27.0	19.3	119.0	126.7

Table VI
OLD VS NEW MODEL ITERATIONS: A/B TESTS RESULTS (B BEFORE A).
NOTE: "OLD" ITERATIONS DENOTE BASELINE RESULTS RETRIEVED FROM [3], WHEREAS "NEW" ITERATIONS REPRESENT FINDINGS FROM OUR CURRENT STUDY.

Large Evaluator	Bielik-11B (B)		PLLuM-12B (A)	
	Old	New	Old	New
gpt-oss-20b	103	119	43	27
Mistral-Small-3.2-24B-Instruct-2506	82	137	64	9
Qwen3-30B-A3B-Instruct-2507	53	46	93	100
Overall Mean	79.3	100.7	66.7	45.3

iterations, *Bielik-11B-v3.0-Instruct* demonstrated a significant and consistent advantage over *PLLuM-12B-nc-chat-250715*, providing answers with higher factual accuracy and stronger grounding in the provided context. More importantly, our systematic stress testing of evaluator models revealed a critical reliability threshold for local evaluation. We empirically demonstrated that small open source models (under 10 billion parameters) are highly susceptible to position bias, consistently failing reversed order A/B tests. Consequently, these models are unsuitable for serving as objective judges. Large judges performed slightly better on position bias in this experiment, *gpt-oss-20b* performed best, effectively resisted order effects and evaluated the actual semantic content, while the other two judges performed worse on this task. Future deployments of RAG pipelines and evaluation frameworks must account for this scale threshold to ensure scientific validity.

VI. LIMITATIONS AND FUTURE WORK

While our findings regarding position bias and Bielik's performance are supported by empirical evidence, several limitations remain. The evaluation was conducted on a single, highly

domain specific dataset involving technical documentation for a low-code platform. The golden set of 146 queries cannot be publicly released due to commercial confidentiality. Given the modest size of the set, we interpret small score differences as indicative trends rather than definitive effects, and we leave formal significance testing (e.g., bootstrap confidence intervals) for future work. Extending this benchmark to more diverse fields, such as complex legal texts or medical data analysis, would provide a more comprehensive assessment of the generative capabilities of Polish LLMs across different registers. Furthermore, our testing protocol relied on standard zero shot prompting for the evaluator models to simulate the most common use case. Future research should investigate whether advanced prompt engineering techniques such as few-shot prompting or enforcing logical reasoning through Chain-of-Thought (CoT) can mitigate the critical position bias observed in small judges (under 10B parameters). Finally, although the 20B+ judges were generally more robust than the smaller ones, their stability was not uniform, and human expert judgment remains the gold standard. Crucially, without a human reference we cannot determine whether the large judges are genuinely accurate or merely more consistent with one another.

REFERENCES

- [1] S. Dadas, M. Perelkiewicz, and R. Poświata, “Pirb: A comprehensive benchmark of polish dense and hybrid text retrieval methods,” 2024. [Online]. Available: <https://arxiv.org/abs/2402.13350>
- [2] K. Ociepa, Ł. Flis, K. Wróbel, A. Gwoździej, and R. Kinas, “Bielik 11b v2 technical report,” arXiv, Tech. Rep. arXiv:2505.02410, 2025.
- [3] S. Bartanowicz and K. Jassem, “Evaluation of two leading polish language models in a real-world rag scenario,” Adam Mickiewicz University, Tech. Rep., 2025.
- [4] B. Orlik, “KartonBERT-USE-base-v1: A polish universal sentence encoder,” <https://huggingface.co/OrlikB/KartonBERT-USE-base-v1>, 2024, hugging Face model repository.
- [5] R. Friel *et al.*, “Ragbench: Explainable benchmark for retrieval-augmented generation,” in *Proceedings of a 2024 venue (preprint: arXiv:2407.11005)*, 2024.
- [6] G. Xiong *et al.*, “Benchmarking retrieval-augmented generation for medicine,” in *Findings of ACL 2024*, 2024.
- [7] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, “Dense passage retrieval for open-domain question answering,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, 2020, pp. 6769–6781. [Online]. Available: <https://aclanthology.org/2020.emnlp-main.550/>
- [8] CYFRAGOVPL, “Pllum-12b-instruct: Model card,” <https://huggingface.co/CYFRAGOVPL/PLLuM-12B-instruct>, 2025.
- [9] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, “Judging llm-as-a-judge with mt-bench and chatbot arena,” 2023. [Online]. Available: <https://arxiv.org/abs/2306.05685>
- [10] J. Ye *et al.*, “Justice or prejudice? quantifying biases in llm-as-a-judge,” in *OpenReview (2024/2025)*, 2025.
- [11] Y. Zhang *et al.*, “Llms-as-judges: A comprehensive survey on llm-based evaluation,” *arXiv preprint arXiv:2412.05579*, 2024.
- [12] H. Yin, S. Vardi, and V. Choudhary, “Fragile preferences: A deep dive into order effects in large language models,” in *Proceedings of the 14th Conference on Computational Natural Language Learning (CoNLL 2025)*, 2025.
- [13] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, “Efficient memory management for large language model serving with pagedattention,” in *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023.