Evaluating Cost-Efficiency of LLMs in a RAG Setup on Polish Wikipedia: Quality vs. Energy Consumption

Patrycja Smits and Tomasz Walkowiak
Faculty of Information and Communication Technology
Wrocław University of Science and Technology, Poland
272940@student.pwr.edu.pl, tomasz.walkowiak@pwr.edu.pl

Abstract

Retrieval-augmented generation has become the dominant paradigm for deploying large language models in knowledge-intensive applications, yet practitioners lack guidance on model selection when both quality and costs matter. We evaluate language models from 4B to 70B parameters, including PLLuM and Bielik families of Polish LLM, within a Polish Wikipedia-based RAG pipeline. Quality assessment uses GPT-4o pairwise comparison across 1,000 PolQA questions with bias mitigation and Bradley-Terry ranking, while energy measurements capture inference costs on NVIDIA H100 hardware. Our findings challenge conventional scaling assumptions: parameter scaling beyond 12B offers minimal quality gains, with mid-size PLLuM-12 matching 70B performance while reducing energy consumption by 83%.

1 Introduction

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) systems that combine large language models (LLMs) with external knowledge sources are increasingly being deployed in both industry and public institutions. By grounding generation in retrieved documents, RAG significantly reduces hallucinations and enables efficient adaptation to evolving knowledge without retraining the underlying models. With the increasing deployment of LLMs in real-world applications, energy consumption has become a critical limiting factor (Chung et al., 2025). In practical deployments, RAG systems must balance response quality with economic efficiency. Although larger models often achieve higher accuracy, they also require substantial GPU resources for inference, leading to high operational costs driven by both hardware investment and energy consumption. As generative models are particularly energy-intensive, inference efficiency has become a key constraint for scalable real-world applications.

This work is motivated by the need to systematically analyze the trade-off between answer quality and energy consumption in long-context processing tasks. While we employ a RAG pipeline as our experimental framework, the core challenge is long-context understanding: models must process 10 Wikipedia passages and synthesize information across them. This setup mirrors any scenario requiring multi-document comprehension.

We evaluated multiple LLMs in an RAG setup over Wikipedia, providing empirical insights into cost-effective model selection under realistic deployment conditions.

There is a substantial body of research that evaluates the quality of RAG systems (Chen et al., 2024b; Wojtasik et al., 2025), as well as studies that focus on the energy efficiency of large language models (Chung et al., 2025; Kwon et al., 2023). However, only a limited number of works jointly analyze both aspects (Vrettos and Klontzas, 2025). In addition, two recently introduced families of Polish-language models - PLLuM (Kocoń and et al., 2025) and Bielik (Ociepa et al., 2025b) - have not yet been systematically evaluated or compared within RAG pipelines in terms of both answer quality and energy efficiency.

This paper addresses a practical question for industry, government, and public institutions in Poland: which large language model should be selected for a fixed RAG pipeline to balance answer quality and energy consumption. We do not explore pipeline design or retrieval strategies; instead, we assume a fixed RAG setup and focus solely on model selection. The Polish Wikipedia was chosen as the knowledge source because it is open and publicly available, ensuring the reproducibility of our experiments.

This paper makes the following contributions:

1. **Empirical evaluation of Polish RAG systems:** We systematically assess seven large

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 4: Student Research Workshop, pages 747–759

language models (4B–70B parameters), including the Polish-language models PLLuM and Bielik, within a fixed RAG pipeline grounded in the Polish Wikipedia.

2. **Joint analysis of answer quality and energy efficiency:** While prior studies have considered either RAG performance or LLM energy consumption separately, we provide a combined assessment, highlighting trade-offs and identifying models that achieve near-optimal quality with minimal computational cost.

3. **Robust pairwise evaluation methodology:** Using GPT-4o as an LLM-as-judge in a self-consistent, bias-mitigated pairwise comparison framework combined with the Bradley-Terry model (Bradley and Terry, 1952). This methodology can serve as a blueprint for future quality–efficiency trade-off studies in RAG pipelines.

The paper is structured as follows. Section 2 reviews related work on RAG systems, LLM evaluation, and energy efficiency studies. Next, section 3 describes the RAG pipeline, dataset, and models used in our experiments. Section 4 details the pairwise evaluation framework, GPT-4o judging procedure, and aggregation via the Bradley-Terry model. Section 5 presents the experimental results, including model rankings, pairwise win probabilities, energy consumption, and the trade-off between quality and efficiency. Section 6 discusses key findings, including diminishing returns from model scaling, quantization effects, and performance-consistency patterns. Finally, Section 7 summarizes our contributions, practical implications, and directions for future work.

2 Related work

Retrieval-Augmented Generation (RAG) has emerged as a prominent approach to enhance language model outputs by grounding responses in retrieved external documents (Lewis et al., 2020). However, evaluating RAG systems presents unique challenges compared to traditional language model evaluation, as assessment must account for both retrieval quality and generation fidelity.

The RAGAS (Retrieval Augmented Generation Assessment) framework (Es et al., 2024) proposes component-level metrics for fine-grained RAG evaluation: faithfulness (whether responses are grounded in retrieved context), answer relevancy (whether responses address the question), context precision (ranking quality of retrieved documents), and context recall (whether all necessary information was retrieved).

Based on these evaluation principles, several comprehensive benchmarks have been developed to standardize the assessment of RAG. The RGB (Retrieval-augmented Generation Benchmark) (Chen et al., 2024b) evaluates four fundamental RAG capabilities: noise robustness, negative rejection, information integration, and counterfactual robustness, using QA pairs constructed from recent news articles to minimize bias from models parametric knowledge.

The use of large language models as evaluators (LLM-as-judge) provides a scalable alternative to expensive human evaluation. (Zheng et al., 2023) demonstrated that GPT-4 achieves more than 80% agreement with human judges on the MT-Bench benchmark, effectively assessing multiple quality dimensions, including helpfulness, relevance and precision.

However, LLM-as-judge approaches exhibit systematic biases. Studies identify position bias (favoring specific response positions), verbosity bias (preferring longer responses regardless of quality), and self-enhancement bias (Wang et al., 2024; Panickssery et al., 2024). To address these limitations, validated mitigation strategies include position swapping (randomizing response order) (Zheng et al., 2023), self-consistency (aggregating multiple independent evaluations) (Wang et al., 2022), and explicit anti-bias instructions. Pairwise comparison, in which judges compare two responses rather than assign absolute scores, is particularly effective in obtaining robust rankings (Zheng et al., 2023). Combined with the Bradley-Terry model (Bradley and Terry, 1952), which estimates latent quality from comparison outcomes, this approach enables a reliable ranking even with noisy judgments (Chiang et al., 2024).

The need to optimize the power consumption of LLMs is increasingly recognized by industry (Patel et al., 2024), and numerous studies have addressed the energy and time efficiency of these models (Kwon et al., 2023; Lin et al., 2025; Fernandez et al., 2025; Chung et al., 2025).

3 Data and Experimental Setup

3.1 Retrieval-Augmented Generation

To evaluate the performance of different LLMs, we employed a testbed based on a Retrieval-Augmented Generation (RAG) pipeline (Wojtasik et al., 2025). The pipeline utilizes a vector database constructed from a Polish Wikipedia dump. BAAI/bge-m3 (Chen et al., 2024a) was used as the embedding model, while BAAI/bge-reranker-v2-m3 (Li et al., 2023; Chen et al., 2024a) served as the reranking model. For each query, the retriever returns the 100 nearest passages, from which the reranker selects the top 10 passages that are subsequently provided to the LLM, which acts as the generator. These retrieved passages from the Polish Wikipedia constitute the contextual input for each evaluated model.

Importantly, the RAG pipeline serves primarily as a controlled framework for evaluating long-context processing. The retrieval component ensures consistent input length and relevance, but the evaluation focuses on how models handle the resulting multi-passage context rather than retrieval quality itself.

3.2 Dataset and models

The evaluation was conducted using the PolQA (Polish Question Answering) dataset, which consists of questions designed to assess the retrieval and reasoning capacities of factual knowledge in Polish (Rybak et al., 2024). The evaluation dataset inherits licensing terms from its source components: the PolQA dataset (CC BY-SA 4.0) (Rybak et al., 2024) and Polish Wikipedia content (CC BY-SA 3.0). Following standard practice for combined datasets, the resulting dataset is released under CC BY-SA 4.0, ensuring compatibility with both source licenses.

For each question, the RAG testbed pipeline (described in the previous section) retrieves 10 relevant documents from Polish Wikipedia, which are then provided as contextual input to the evaluated models. The models generated responses by synthesizing information from these sources. They were instructed to refer to references where appropriate, allowing assessment of both response quality and proper use of context. The prompt is shown in Figure 1.

A total of 1,000 questions from the PolQA dataset test split were used, ensuring diversity across question types and difficulty levels. To balance computational efficiency with statistical robustness, the evaluation was conducted in 10 batches of 100 questions each.

```
The numbered list of documents is below:
<results>
Document: 1
....
</results>
Answer the user's question using only the
information contained in the documents,
not prior knowledge. Provide a high-quality,
grammatically correct answer in Polish.
The answer should include citations to the
documents from which the information originates.
Cite  a document using the symbol [doc_no],
referring to a fragment, e.g., [1] for a fragment
from  document 1. If the documents do not contain
the information needed to answer the question,
return the text: "Could not find an answer
to the question.". Question:
....
```

Figure 1: Prompt used for RAG pipeline evaluation, originally written in Polish.

Seven large language models were evaluated, spanning a broad range of parameter scales and memory requirements. We included two families of Polish models, namely PLLuM (Kocoń and et al., 2025) and Bielik (Ociepa et al., 2025b,a), as well as Gemma-3-4, which serves as an example of a smaller (4B) but highly performant multilingual model. In addition, two GGUF-based quantization variants (Organization, 2023–2025) of the largest model were considered. Table 1 provides an overview of all models evaluated.

4 Evaluation Methodology

The quality evaluation methodology combined pairwise comparison as the evaluation framework with GPT-4o as judge (LLM-as-judge approach).

The pairwise comparison was selected as the evaluation framework to derive a global ranking from a series of direct comparisons between model pairs (Liu et al., 2024). Unlike approaches that assign absolute scores to individual responses - which require consistent interpretation of rating scales across diverse question types - pairwise evaluation focuses on relative quality within each comparison. This design naturally aligns with the Bradley–Terry model, which estimates model rankings based on the outcomes of pairwise contests.

GPT-4o from OpenAI was used as the judge model. For each evaluation instance, the model received the user query along with two candidate responses (from Model A and Model B). The judge's

Name	Model Identifier (Hugging Face)	Format	Params	Memory in GB
PLLuM-70	CYFRAGOVPL/Llama-PLLuM-70B-chat-250801	FP16	70B	131.4
PLLuM-70-q8	quantized version of PLLuM-70	Q8_0	70B	69.9
PLLuM-70-q4	quantized version of PLLuM-70	Q4_K_M	70B	40.8
PLLuM-12	CYFRAGOVPL/PLLuM-12B-nc-chat	FP16	12B	22.84
Bielik-11	speakleash/Bielik-11B-v2.6-Instruct	FP16	11B	20.8
Bielik-4.5	speakleash/Bielik-4.5B-v3.0-Instruct	FP16	4.6B	8.9
Gemma-3-4	google/gemma-3-4b-it	FP16	4B	8.6

Table 1: Specifications of the evaluated large language models. FP16 denotes 16-bit floating-point weights; Q8_0 denotes 8-bit integer-quantized weights; Q4_K_M denotes 4-bit mixed k-means–based integer quantization. Memory usage (last column) is reported by vLLM.

task was to decide which response was superior or to declare a tie when both were equivalent. The evaluation followed a hierarchy of criteria: factual correctness (highest priority), completeness of the response, adherence to instructions, and clarity and structure of the output.

Due to the stochastic nature of LLMs, a single comparison may not provide fully reliable results. To address this issue, the self-consistency technique was applied. Five independent comparisons were performed for each pair of responses to a given question, with temperature=0.7 to enable response diversity. The final winner was determined by majority voting allowing for ties. Position bias was mitigated by randomly swapping the positions of responses A and B with a probability of 0.5, using a fixed seed (seed=42) to ensure reproducibility. Verbosity bias was mitigated through explicit anti-bias instructions in the evaluation prompt. All API calls were made using the OpenAI Python client library (version 1.58.1) with max_tokens=1000.

After aggregation across all questions, summary statistics were calculated for each pair of models, including the total number of questions won by Model A, the total won by Model B, and the number of ties.

The Bradley–Terry model was applied (Bradley and Terry, 1952) to derive global rankings from pairwise comparison results. This probabilistic model estimates the relative strength of each model based on the observed results of pairwise comparisons, where the probability that model i defeats model j is given by:

$$P(i \text{ beats } j) = \frac{\pi_i}{\pi_i + \pi_j} \tag{1}$$

where π_i and π_j are the latent strength parameters (ratings) for models i and j, respectively.

Ratings π_i are estimated from pairwise comparison outcomes using the iterative MM (Minorization-Maximization) algorithm (Hunter, 2004). Let w_{ij} denote the number of times model i defeated model j (with ties counted as 0.5), and let n_{ij} denote the total number of comparisons between models i and j. The algorithm proceeds as follows:

1. Initialize all ratings to 1: $\pi_i^{(0)} = 1$ for all models i.
2. At iteration t, update each model's rating using:
$$\pi_i^{(t+1)} = \frac{\sum_{j \neq i} w_{ij}}{\sum_{j \neq i} \frac{n_{ij}}{\pi_i^{(t)} + \pi_j^{(t)}}} \tag{2}$$
3. Normalize ratings so that $\sum_i \pi_i = N$, where N is the number of models.
4. Repeat steps 2-3 until convergence or maximum iterations reached.

The algorithm converges when the maximum absolute change in any rating between iterations falls below $\epsilon = 10^{-6}$, or after a maximum of 100 iterations.

The Bradley-Terry model takes into account the strength of opponents, meaning that a model achieving victories mainly against weaker competitors will receive a lower rating than a model with similar win statistics but facing stronger competition. The normalized ratings have a mean of 1.0, with values above 1.0 indicating above-average performance and values below 1.0 indicating below-average performance.

The evaluation was conducted in batches. For a data set of 1,000 questions, this resulted in 10 independent evaluations, each producing a separate

Bradley-Terry ranking. For each model, 10 rating values were collected (one per batch) and the mean and standard deviation were calculated.

5 Results

5.1 Bradley-Terry Model Rankings

Table 2 presents the final Bradley-Terry rankings derived from pairwise comparisons between all model pairs. The ranking represents the mean rating for 10 batches of 100 questions each, with standard deviations indicating the stability of the ranking.

Rank	Model	Rating
1	PLLuM-70	1.292 ± 0.019
2	PLLuM-12	1.273 ± 0.030
3	PLLuM-70-q8	1.207 ± 0.026
4	Bielik-11	1.061 ± 0.014
5	Bielik-4.5	0.800 ± 0.018
6	PLLuM-70B-q4	0.734 ± 0.044
7	Gemma-3-4	0.632 ± 0.034

Table 2: Global ranking of language models derived using the Bradley-Terry model from pairwise comparisons of RAG-generated answers. Reported values correspond to the mean quality rating $\pm$ standard deviation across 10 independent batches of 100 questions each. Higher ratings indicate stronger overall preference in pairwise evaluations.

PLLuM-70 achieved the highest rating (1.292 ± 0.019), followed closely by PLLuM-12 (1.273 ± 0.030) and PLLuM-70-q8 (1.207 ± 0.026). Bielik models occupy middle positions, with Bielik-11 (1.061 ± 0.014) substantially outperforming Bielik-4.5 (0.800 ± 0.018). Gemma-3-4 received the lowest rating (0.632 ± 0.034).

Notably, the gap between the top-ranked PLLuM-70 and second-ranked PLLuM-12 is minimal (0.019), despite PLLuM-70 having approximately six times more parameters. Standard deviations range from 0.014 (Bielik-11) to 0.044 (PLLuM-70-q4), indicating varying degrees of consistency across question batches.

5.2 Pairwise Win Probabilities

Table 3 presents the win probability matrix, where each entry indicates the probability that the row model produces a superior response compared to the column model.

The top three models show near-even head-to-head matchups: PLLuM-70 versus PLLuM-12 (0.50), PLLuM-12 versus PLLuM-70-q8 (0.51), and PLLuM-70 versus PLLuM-70-q8 (0.52). The 8-bit quantized PLLuM-70-q8 maintains competitive win rates against top models (0.48–0.49), demonstrating preservation of quality. In contrast, 4-bit quantization introduces substantial degradation: PLLuM-70-q4 achieves only 0.36-0.38 against PLLuM-70/PLLuM-12, comparable to the bottom-tier models.

5.3 Energy Consumption

Using the same dataset as in the quality analysis, we measured the energy consumption of the analyzed models. A total of 100 prompts were sent to each model sequentially, and power consumption was recorded using the power telemetry provided by the server's power supply unit. In addition, we measured the number of tokens generated for each question. The results are presented in Table 4, which reports the energy consumption per token and per question, as well as the average response length. All experiments were performed on an NVIDIA H100 GPU (96GB) using vLLM server version 0.10.2 (Kwon et al., 2023). All models, except PLLuM-70, were deployed on a single GPU. Due to its memory requirements exceeding 131.4 GB, PLLuM-70 was deployed using two GPUs with model parallelism.

Three significant observations emerge from the experiments. First, the average response length varies considerably between models, ranging from 83 to 670 tokens, despite using identical prompts with the temperature set to zero. This variation has a direct impact on power consumption, as longer autoregressive sequences require proportionally more computational resources per token.

Secondly, the quantized models exhibit noticeably poorer performance. Although they require substantially less GPU memory (approximately 2× for Q8_0 and 4× for Q4_K_M), their energy consumption is 3.34× and 6.16× higher, respectively. Although low-bit quantization (4-bit and 8-bit) reduces memory footprint, our experiments demonstrate that on GPUs such as the NVIDIA H100, these models often achieve slower inference compared to FP16 models (Lin et al., 2025). Although the H100 can accelerate INT8/INT4, used GGUF quantizations may not fully utilize low-bit Tensor Cores due to runtime dequantization to FP16 and limited kernel optimization in vLLM.

Third, Gemma 3-4 exhibits a notably high energy consumption relative to its size. In general,

Model	PLLuM-70	PLLuM-12	PLLuM-70-q8	Bielik-11	Bielik-4.5	PLLuM-70-q4	Gemma-3-4
PLLuM-70	–	0.50	0.52	0.55	0.62	0.64	0.67
PLLuM-12	0.50	–	0.51	0.55	0.61	0.63	0.67
PLLuM-70-q8	0.48	0.49	–	0.53	0.60	0.62	0.66
Bielik-11	0.45	0.45	0.47	–	0.57	0.59	0.63
Bielik-4.5	0.38	0.39	0.40	0.43	–	0.52	0.56
PLLuM-70-q4	0.36	0.37	0.38	0.41	0.48	–	0.54
Gemma-3-4	0.33	0.33	0.34	0.37	0.44	0.46	–

Table 3: Pairwise win probability matrix for all evaluated models in the RAG setting. Each cell (i, j) reports the empirical probability that the model in row i produces a response judged superior to the model in column j by GPT-4o, after aggregation across all questions and tie handling. Values close to 0.5 indicate near-parity between models, while larger deviations from 0.5 reflect systematic performance differences in answer quality.

Model	**Energy per token [J/t]**	**Energy per quest. [J/q]**	**Aver. resp. length**
PLLuM-70	84.6	21 580	255
PLLuM-70-q8	282.2	63 052	223
PLLuM-70-q4	521.0	43 080	83
PLLuM-12	15.0	3 520	235
Bielik-11	15.3	10 235	670
Bielik-4.5	10.8	4 166	386
Gemma-3-4	24.8	4 299	173

Table 4: Energy consumption of the evaluated LLMs in the RAG pipeline, reported per token and per question, along with the average response length. Measurements were obtained using PSU power telemetry on an NVIDIA H100 GPU with vLLM version 0.10.2.

the energy required per token is roughly proportional to the size of the model. When we normalize the energy per token by the model size, we obtain values of 1.2 for PLLuM-70, 1.25 for PLLuM-12, and as high as 6.2 for Gemma-3-4B. Although Gemma-3-4B has fewer parameters than PLLuM-12 (based on Mistral-NeMo-Base-12B), inference performance in vLLM is mainly determined by memory access patterns and kernel efficiency rather than parameter count. The PagedAttention mechanism of vLLM (Kwon et al., 2023) and model-specific fused kernels favor architectures with mature attention and normalization implementations. Consequently, PLLuM-12, which benefits from highly optimized kernels in vLLM, achieves higher FP16 throughput than the smaller but less optimized Gemma-3-4B.

5.4 Efficiency vs. Energy

Having analyzed both the quality of the LLMs (Table 2) and their energy consumption (Table 4) in the RAG pipeline, we visualized the relationship in a 2D scatter plot (Figure 2). The y-axis represents the Bradley-Terry model ratings, while the x-axis shows the average energy consumption per RAG question. This visualization highlights the trade-off between model quality and energy efficiency, enabling a comparative assessment of which models achieve high performance while minimizing energy usage. The resulting conclusion is that PLLuM-12 is the preferred model, as it achieves the lowest energy consumption while performing only slightly worse than PLLuM-70 (1.273 versus 1.292).

6 Discussion

6.1 Diminishing Returns with Model Scale

The near-parity observed between PLLuM-70 and PLLuM-12 (win probability of 0.50) constitutes one of our most notable findings. Although PLLuM-70 contains approximately 6x times more parameters than PLLuM-12 (70B vs. 12B), their performance differs only marginally, as evidenced by a rating gap of merely 0.019 and fully overlapping confidence intervals. This result challenges prevailing assumptions regarding the benefits of model scaling and indicates that raw parameter count may not be the principal determinant of performance in Polish RAG-based question-answering tasks.

This observation aligns with recent findings in recent work on scaling laws (Kaplan et al., 2020), where performance gains from scale follow dimin-

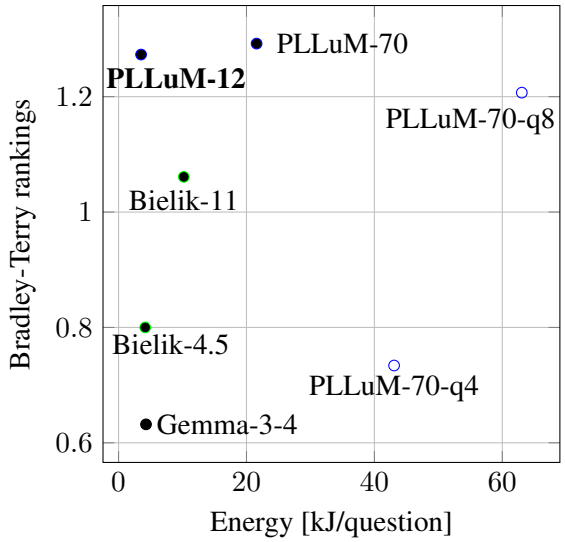

Figure 2: Trade-off between model quality and energy efficiency in the RAG pipeline. The x-axis shows the average energy consumption per RAG question, while the y-axis represents Bradley–Terry model ratings.

ishing returns beyond certain thresholds. For RAG applications specifically, the ability to effectively leverage retrieved context may depend more on architectural design - such as attention mechanisms optimized for long contexts or positional encodings that preserve document boundaries - than on raw model capacity. The retrieved passages already contain the factual information needed to answer questions; the model's role is primarily to extract, synthesize, and reformulate this information rather than to rely on parametric knowledge.

From a practical deployment perspective, this finding has significant implications. PLLuM-12 offers comparable performance at substantially lower computational costs: memory requirements drop by 82.6% (from 131.4 GB to 22.8 GB), and energy consumption decreases by 83.8% (from 21.6 kJ to 3.5 kJ per question), while maintaining 98.5% of the quality (rating 1.273 vs 1.292). For organizations deploying Polish RAG systems, this represents a clear Pareto improvement: minimal quality sacrifice for dramatic resource savings.

6.2 Quantization: Trade-offs and Thresholds

Results reveal a clear threshold effect in quantization. The 8-bit quantized PLLuM-70-q8 retains 93.4% of PLLuM-70's quality (rating 1.207 vs 1.292) and achieves near-parity in head-to-head matchups (48% win rate), while reducing memory by 46.8% (69.9 GB vs 131.4 GB). This demonstrates that Q8_0 quantization preserves the model's ability to process and synthesize retrieved information without substantial degradation.

In stark contrast, 4-bit quantization crosses a quality threshold. PLLuM-70-q4 achieves win probabilities of only 0.36-0.38 against top-performing models and exhibits the highest variance across batches (std = 0.044), more than three times that of the most stable model. More concerningly, PLLuM-70-q4 maintains only a marginal advantage (0.54 win probability) over the much smaller Gemma-3-4 model, suggesting that aggressive 4-bit quantization negates the benefits of the larger parameter count.

Counter-intuitively, quantized models consume substantially more energy per token on H100 hardware: 3.34× (Q8_0) and 6.16× (Q4_K_M) compared to FP16.

6.3 Performance and Consistency as Independent Attributes

An unexpected pattern emerged in the relationship between model stability and overall performance. Bielik-11 exhibits the lowest variance across batches (std = 0.014) despite occupying a mid-tier position in the ranking (rating 1.061). Conversely, the top-performing PLLuM-70 shows moderate variability (std = 0.019), while PLLuM-70-q4, displays the highest variance (std = 0.044).

This lack of correlation suggests that stability and overall quality may be governed by distinct underlying model characteristics. The consistency exhibited by Bielik-11 may stem from architectural features such as more uniform attention distributions or training dynamics that promote smoother gradient propagation. These properties support robust generalization across diverse question types, even if the model's absolute performance is lower.

The observed pattern may also reflect differences in how models handle more challenging inputs. PLLuM models may engage in more complex reasoning or synthesis when confronted with difficult questions, which could lead to higher output variance but superior average performance. In contrast, Bielik models may rely on more conservative decoding or representation strategies, producing stable yet occasionally suboptimal responses.

6.4 Beyond RAG: Implications for Long-Context Tasks

While this evaluation uses RAG as a testbed, the findings extend to any scenario requiring long-context processing. Our results suggest that 12B

models can effectively synthesize information across multiple documents, with our 10-passage setup representing approximately 2000-4000 tokens of context, without requiring 70B-scale capacity.

This has direct implications for multi-document summarization, legal document analysis, regulatory compliance review, and technical documentation processing—all tasks that involve identifying relevant information across extended contexts and presenting coherent syntheses. Similarly, code understanding tasks that require processing multiple files simultaneously mirror the multi-passage synthesis in our experiments.

The key insight applies broadly: once relevant information is present in context (whether via retrieval, direct input, or tools), the model's primary role shifts from recall to synthesis. Our findings suggest that this synthesis capability scales differently than general knowledge or complex reasoning. Mid-size models achieve near-parity with larger variants when required information is explicitly provided, with performance gaps appearing primarily in scenarios demanding extensive parametric knowledge or multi-step reasoning beyond the provided context. This distinction has practical implications for model selection across diverse long-context applications beyond retrieval-augmented generation.

7 Conclusion

With the growing adoption of large language models in production systems, organizations face a critical challenge: balancing response quality with operational efficiency. This study evaluates the energy-quality trade-off for seven language models (4B-70B parameters) in a Wikipedia-based RAG pipeline, combining quality assessment through pairwise comparison with empirical energy measurement.

Our evaluation yields three principal contributions. First, we provide the first systematic assessment of Polish language models (PLLuM and Bielik families) in RAG scenarios, demonstrating that mid-size models (12B parameters) achieve near-parity with flagship 70B variants while consuming 83.8% less energy. Second, we reveal a quantization paradox: while 8-bit precision preserves quality, both 8-bit and 4-bit quantization increase per-token energy consumption by 3.3× to 6.2× on H100 GPUs, contradicting assumptions about compression benefits. Third, we contribute a replicable evaluation methodology combining pairwise comparison with bias mitigation, Bradley-Terry aggregation, and synchronized energy measurement, enabling holistic cost-aware model assessment.

These findings carry immediate practical implications for organizations deploying Polish RAG systems. For quality-critical applications with flexible budgets, PLLuM-12 represents the optimal choice, delivering 98.5% of flagship model quality at one-sixth the operational cost. For memory-constrained environments, 8-bit quantization offers acceptable quality preservation (93.4% retention) with halved memory requirements, though practitioners should note increased energy consumption on current hardware. Conversely, 4-bit quantization should be avoided for production deployments, as severe quality degradation outweighs memory savings. More broadly, our results challenge the assumption that larger models are necessary for knowledge-intensive tasks when external information is provided through retrieval.

Several important directions warrant future investigation. First, extending evaluation beyond Wikipedia to domain-specific corpora such as legal documents, medical records, or technical documentation would test whether these trade-offs generalize across different knowledge types. Second, reducing evaluation costs represents a critical challenge. Our pairwise comparison approach required substantial API expenses, limiting scalability to larger model sets or frequent benchmarking cycles. Developing cost-efficient methods such as learned judging models or strategic sampling would enable more comprehensive quality assessment. Third, investigating alternative quantization frameworks and inference engines could identify configurations that realize theoretical energy benefits without current overhead. Finally, establishing a standardized benchmark for Polish language models incorporating both quality metrics and resource consumption would provide the community with a shared reference for evaluating progress and guiding deployment decisions. As language technologies expand globally, such comprehensive evaluation frameworks become essential for ensuring advanced NLP capabilities remain accessible across diverse linguistic communities and resource environments.

Limitations

The evaluation used employs a single RAG pipeline configuration based on Polish Wikipedia. The retrieval system uses BAAI/bge-m3 for embedding and BAAI/bge-reranker-v2-m3 for reranking, with fixed hyperparameters (top-100 retrieval, top-10 reranking). Performance may vary substantially with alternative retrieval strategies. Sparse retrievers may favor different model characteristics than dense retrievers, while late-interaction models could alter the quality-energy trade-off by reducing the number of passages requiring full LLM processing. The fixed setting of top-k=10 represents a single point in the quality trade-off space for recovery; adaptive retrieval strategies that adjust passage counts based on the complexity of the question could yield different optimal model choices. Furthermore, Wikipedia articles provide well-structured, encyclopedic text; retrieval from noisy web sources, conversational data, or domain-specific corpora (legal documents, medical records) may change relative model performance. Our findings should therefore be interpreted as specific to Wikipedia-based RAG with dense retrieval, not as universal claims about all RAG configurations.

The energy measurements were conducted exclusively on NVIDIA H100 (96GB HBM3) GPUs using vLLM version 0.10.2. This introduces several potential sources of non–generalizability. First, hardware-specific optimizations mean that different GPU architectures exhibit different efficiency characteristics. Previous-generation NVIDIA GPUs (A100 with 3rd-gen Tensor Cores, A40 for inference workloads) lack H100's fourth-generation Tensor Cores and FP8 support, potentially showing different throughput profiles for FP16 models. Alternative accelerators (Google TPUs, AWS Trainium, AMD MI300) employ fundamentally different memory hierarchies and instruction sets, potentially reversing our findings about quantization efficiency. Second, software framework dependencies substantially affect performance. vLLM's PagedAttention and kernel implementations favor LLaMA/Mistral architectures, as evidenced by Gemma-3-4's poor efficiency. Alternative frameworks - TensorRT-LLM with its optimized fusion patterns, llama.cpp with its CPU-targeted quantization kernels, or Hugging Face Transformers with its flexibility but lower peak throughput—would produce different absolute energy values and potentially different model rankings. Third, quantization format matters. We evaluated the Q8_0 and Q4_K_M formats from the GGUF quantization family. Alternative schemes - GPTQ, AWQ, SmoothQuant, or activation-sensitive quantization - may exhibit different quality-energy trade-offs. Our conclusions about 4-bit quantization degradation apply specifically to the Q4_K_M scheme and may not be generalized to other 4-bit approaches. However, the relative rankings and trade-offs (e.g., PLLuM-12 vs. PLLuM-70) likely exhibit more robustness, as they reflect fundamental model characteristics rather than implementation details.

The pairwise comparison approach using GPT-4o as a judge introduces both financial and methodological limitations. Evaluating seven models on 1,000 questions with five-fold self-consistency required 105,000 GPT-4o API calls (21 model pairs × 1,000 questions × 5 judgments), incurring substantial costs. This expense limits scalability: evaluating 15 models would require 525,000 calls.

Additionally, our evaluation treats response quality holistically through pairwise comparison but does not explicitly control for user preferences regarding response length. The models exhibited substantial variation in output verbosity (83–670 tokens) despite identical prompts and temperature settings. Although GPT-4o was instructed not to favor longer responses unless they improved correctness or completeness, some users may prefer concise answers (minimizing reading time), while others prefer comprehensive explanations (maximizing information). Our quality rankings conflate these preferences into a single score.

Acknowledgements

GPT-4o was used solely for experimental evaluation as described in the methodology. The work was financed by CLARIN-PL: Common Language Resources and Technology Infrastructure (POIR.04.02.00-00C002/19, 2024/WK/01, FENG.02.04-IP.040004/24).

References

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. M3-embedding: Multi-linguality, multi-functionality,

multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024b. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Jae-Won Chung, Jeff J. Ma, Ruofan Wu, Jiachen Liu, Oh Jun Kweon, Yuxuan Xia, Zhiyu Wu, and Mosharaf Chowdhury. 2025. The ml.energy benchmark: Toward automated inference energy measurement and optimization. *Preprint*, arXiv:2505.06371.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Jared Fernandez, Clara Na, Vashisth Tiwari, Yonatan Bisk, Sasha Luccioni, and Emma Strubell. 2025. Energy considerations of large language model inference and efficiency optimizations. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32556–32569, Vienna, Austria. Association for Computational Linguistics.

David R. Hunter. 2004. Mm algorithms for generalized bradley-terry models. *The Annals of Statistics*, 32(1):384–406.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Jan Kocoń and et al. 2025. Pllum: A family of polish large language models. *Preprint*, arXiv:2511.03823.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making large language models a better foundation for dense retrieval. *Preprint*, arXiv:2312.15503.

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. 2025. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. *Preprint*, arXiv:2405.04532.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan vulic, anna korhonen, and Nigel Collier. 2024. Aligning with human judgement: The role of pairwise preference in large language model evaluators.

Krzysztof Ociepa, Łukasz Flis, Remigiusz Kinas, Krzysztof Wróbel, and Adrian Gwoździej. 2025a. Bielik v3 small: Technical report. *Preprint*, arXiv:2505.02550.

Krzysztof Ociepa, Łukasz Flis, Krzysztof Wróbel, Adrian Gwoździej, and Remigiusz Kinas. 2025b. Bielik 11b v2 technical report. *Preprint*, arXiv:2505.02410.

GGML Organization. 2023–2025. llama.cpp: Port of transformer llms for efficient inference. `https://github.com/ggml-org/llama.cpp`. Accessed: 2025-12-22.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Pratyush Patel, Esha Choukse, Chaojie Zhang, Íñigo Goiri, Brijesh Warrier, Nithish Mahalingam, and Ricardo Bianchini. 2024. Characterizing power management opportunities for llms in the cloud. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, ASPLOS '24, page 207–222, New York, NY, USA. Association for Computing Machinery.

Piotr Rybak, Piotr Przybyła, and Maciej Ogrodniczuk. 2024. PolQA: Polish question answering dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12846–12855, Torino, Italia. ELRA and ICCL.

Konstantinos Vrettos and Michail E. Klontzas. 2025. Accurate and energy efficient: Local retrieval-augmented generation models outperform commercial large language models in medical tasks. *Preprint*, arXiv:2506.20009.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models.

Konrad Wojtasik, Adrian Berdowski, Inez Okulska, and Maciej Piasecki. 2025. Polichat: Retrieval augmented generation on university documents and regulations. In *Computational Science – ICCS 2025 Workshops: 25th International Conference, Singapore, Singapore, July 7–9, 2025, Proceedings, Part V*, page 273–288, Berlin, Heidelberg. Springer-Verlag.

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. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

A Pairwise Comparison Algorithm

This appendix provides detailed documentation of the pairwise comparison procedure used to evaluate model outputs.

A.1 Evaluation Procedure

For each question q and model pair (M_i, M_j), we performed $k = 5$ independent comparisons using GPT-4o as the judge model. Algorithm 1 presents the complete evaluation procedure.

The CompareResp function sends a structured prompt to GPT-4o (temperature=0.0) containing the user question and both responses. The model returns a JSON object specifying the winner: "A", "B", or "Tie".

A.2 Evaluation Criteria

The judge model evaluates responses using hierarchical criteria (ordered by priority):

Correctness (Highest Priority) Factual accuracy, logical validity, and absence of hallucinations. Citations must appropriately support claims.

Algorithm 1 Pairwise Comparison with Position Swapping

Require: Question q, responses r_i and r_j from models M_i and M_j
Require: Number of comparisons $k = 5$, random seed s
Ensure: Aggregated comparison result: votes_i, votes_j, $\text{votes}_{\text{tie}}$

Initialize: $\text{votes}_i \leftarrow 0$, $\text{votes}_j \leftarrow 0$, $\text{votes}_{\text{tie}} \leftarrow 0$
Set random seed: random.seed(s)
for $c = 1$ to k **do**
    $p \sim \text{Uniform}(0, 1)$ ▷ Random position assignment
    if $p < 0.5$ **then**
        winner $\leftarrow \text{CompareResp}(q, r_i, r_j)$
        if winner = "A" **then**
            $\text{votes}_i \leftarrow \text{votes}_i + 1$
        else if winner = "B" **then**
            $\text{votes}_j \leftarrow \text{votes}_j + 1$
        else
            $\text{votes}_{\text{tie}} \leftarrow \text{votes}_{\text{tie}} + 1$
        end if
    else
        winner $\leftarrow \text{CompareResp}(q, r_j, r_i)$
        if winner = "A" **then**
            $\text{votes}_j \leftarrow \text{votes}_j + 1$
        else if winner = "B" **then**
            $\text{votes}_i \leftarrow \text{votes}_i + 1$
        else
            $\text{votes}_{\text{tie}} \leftarrow \text{votes}_{\text{tie}} + 1$
        end if
    end if
end for
return votes_i, votes_j, $\text{votes}_{\text{tie}}$

Completeness Full coverage of all question components. Important details must not be omitted. Relevant sources should be referenced.

Instruction Adherence Strict adherence to user instructions. Responses must stay on-topic without unnecessary content.

Clarity & Structure. Understandability, organization, and appropriate conciseness.

Responses win if superior on higher-priority criteria. Ties occur only when responses are essentially equal in correctness and completeness.

A.3 Bias Mitigation

Position Randomization. Each comparison randomly assigns responses to positions A and B with $p = 0.5$, mitigating position bias.

Self-Consistency. $k = 5$ independent comparisons per question with majority voting reduce judgment inconsistency.

Anti-Bias Instructions. Explicit prompt instructions prohibit favoring: longer responses, confident tone, writing style, or excessive citations unless they improve correctness or completeness.

Reproducibility. Fixed random seed ($s = 42$) ensures deterministic position assignments.

B Evaluation Dataset

This appendix provides detailed information about the PolQA dataset (Rybak et al., 2024) used for model evaluation.

B.1 Dataset Overview

PolQA (Polish Question Answering) is a question-answering dataset based on Polish Wikipedia. The dataset was specifically designed to evaluate retrieval-augmented generation (RAG) systems in the Polish language. Questions cover diverse domains including history, science, culture, sports, and general knowledge, reflecting the broad scope of encyclopedic content.

For this evaluation, we used the test split containing 1,000 questions. Each question was processed through a RAG pipeline that retrieved 10 relevant document passages from Polish Wikipedia. Model responses were generated based on these retrieved contexts, and the evaluation focused on how accurate and complete the answers were.

B.2 Dataset Statistics

Table 5 summarizes key statistics of the evaluation dataset.

Characteristic	Value
Total questions	1,000
Retrieved documents per question	10
Total document passages	10,000
Source corpus	Polish Wikipedia
Question Length (characters)	
Average	66.6
Minimum	19
Maximum	208

Table 5: Statistics of the PolQA evaluation dataset. Retrieval scores indicate semantic similarity between questions and retrieved passages.

B.3 Question Characteristics

Questions exhibit substantial variation in length and complexity. The shortest question (19 characters) is "Jaki kolor ma neon?" ("What color is neon?"), while the longest (208 characters) asks about the historical development of African American spirituals and their relationship to blues music. This diversity ensures that model rankings reflect capability across both simple factual queries and complex, multi-clause questions requiring deeper understanding.

C Judge Prompt Template

The evaluation prompt sent to GPT-4o for each comparison:

```
The following task is to compare two responses
to the same user question. The responses will be
written in Polish. Your goal is to decide which
response is better overall, or whether they are
tied. You MUST follow the evaluation procedure
exactly.

IMPORTANT: The responses may contain
citations/references to source documents (e.g.,
[1], [2], "According to Document X", etc.).
These citations are part of the response and
should be considered when evaluating correctness
and completeness.

EVALUATION PROCEDURE (INTERNAL)
1. Read the user question carefully and identify
   all explicit and implicit requirements.
2. Evaluate Response A and Response B separately
   against each criterion.
3. Compare Response A and Response B criterion
   by criterion.
4. Weigh the criteria in the given priority
   order.
5. Make a final decision.

You must reason step by step internally.
You must not reveal your reasoning or analysis.
```

```
EVALUATION CRITERIA (ordered by priority)
1. Correctness
   - Are all factual statements accurate?
   - Is the reasoning logically valid?
   - Are there hallucinations, incorrect claims,
     or unjustified assumptions?
   - If citations are present: Are they used
     appropriately to support claims?

2. Completeness
   - Does the response fully address all parts
     of the question?
   - Are important steps, details, or
     explanations missing?
   - If citations are present: Does the response
     reference relevant sources?

3. Instruction Adherence & Relevance
   - Does the response strictly follow the
     user's instructions?
   - Does it stay on-topic and avoid
     unnecessary content?

4. Clarity & Structure
   - Is the response easy to understand?
   - Is it well-structured and appropriately
     concise?

ANTI-BIAS RULES
- Do not favor longer responses unless they
  are clearly more correct or complete.
- Do not favor more confident or assertive
  tone.
- Do not favor writing style alone.
- Do not penalize minor language imperfections
  in Polish unless they affect understanding.
- Do not favor responses with more citations
  unless they provide better correctness or
  completeness.
- Assume both responses are written in good
  faith.

Choose "Tie" only if:
- Both responses are essentially equal in
  correctness and completeness, AND
- Any differences are minor or stylistic, AND
- You cannot confidently prefer one over the
  other.

User question:
{question}

Response A:
{response_a}

Response B:
{response_b}

Return ONLY the following JSON object and
nothing else:
{
  "winner": "A" | "B" | "Tie"
}
```

The prompt incorporates: (1) explicit criteria hierarchy, (2) step-by-step reasoning instructions, (3) anti-bias rules, and (4) structured JSON output for reliable parsing