Diminishing Returns: LLM Accuracy vs. Computational Cost on Constrained Hardware

Gustavo Fernandes Luz[1], **Tiago Buarque Assunção de Carvalho**[1]

[1]Universidade Federal do Agreste de Pernambuco (UFAPE)
Pernambuco – PE – Brazil

{gustavo.luz,tiago.buarque}@ufape.edu.br

Abstract. *Large Language Models (LLMs) have become increasingly vital across diverse fields. This study investigates the trade-off between accuracy and computational cost in language models when operating under hardware constraints. While mainstream research relies heavily on parametric scaling, our evaluation of thirteen models on the MMLU benchmark reveals diminishing marginal returns. Specifically, increasing parameter counts yields progressively smaller accuracy gains while drastically increasing execution time. We conclude that continuous structural scaling is inefficient for resource-constrained environments.*

1. Introduction

The evolution of Large Language Models (LLMs), predominantly grounded in the Transformer architecture [Vaswani et al. 2017], has been strongly driven by the so-called scaling laws, according to which the systematic increase of the number of parameters and the volume of training data results in consistent performance improvements [Kaplan et al. 2020]. However, recent literature shows that this indiscriminate structural scaling frequently culminates in severe computational inefficiencies [Hoffmann et al. 2022]. The substantial growth in model size demands highly specialized infrastructures, significantly raising energy consumption and inference latency [Desislavov et al. 2023], factors that render their practical adoption infeasible in environments subject to hardware constraints.

In this scenario, Small Language Models (SLMs) emerge as a viable alternative, providing greater computational efficiency, feasibility of local execution, and predictability of operational costs. Supporting the viability of smaller models, [Filho and Fernandes 2025] demonstrate that locally deployed SLMs can successfully execute complex tasks, such as paper screening. Furthermore, [Oliveira et al. 2025] reinforce that adopting open models for task automation can be achieved by decoupling planning from execution, thereby ensuring data privacy and maintaining low operational costs.

While a smaller parameter count restricts a model's knowledge and increases its susceptibility to hallucinations, contemporary research demonstrates that these constraints can be mitigated through context enrichment strategies [Ganesh et al. 2023, Liu et al. 2024]. Integrating Retrieval-Augmented Generation (RAG) addresses the structural limitations of SLMs by grounding response generation in external, up-to-date knowledge bases. [Agustiani et al. 2026].

Despite the empirical success of context enrichment strategies, mainstream research predominantly relies on parametric scaling to advance model accuracy, frequently neglecting its computational overhead [Prucs et al. 2025]. Current literature lacks methodologically standardized frameworks to quantify the trade-offs between model scale, marginal performance gains, and hardware constraints—such as the necessity for hybrid CPU-GPU execution due to VRAM limitations.

To address these gaps, this work empirically quantifies the trade-off between accuracy and computational cost in small to medium-scale language models operating under severe hardware constraints. To this end, a standardized evaluation of 13 models (ranging from 1 to 30 billion parameters) was conducted using the MMLU benchmark in an environment limited to 6 GB of VRAM, forcing hybrid (CPU-GPU) execution for larger architectures.

The remainder of this paper is structured as follows: Section 2 presents a systematic literature review on context enrichment strategies; Section 3 describes the experimental design and hardware configuration; Section 4 discusses the empirical results; and Section 5 concludes the paper with directions for future work.

2. Systematic Literature Review

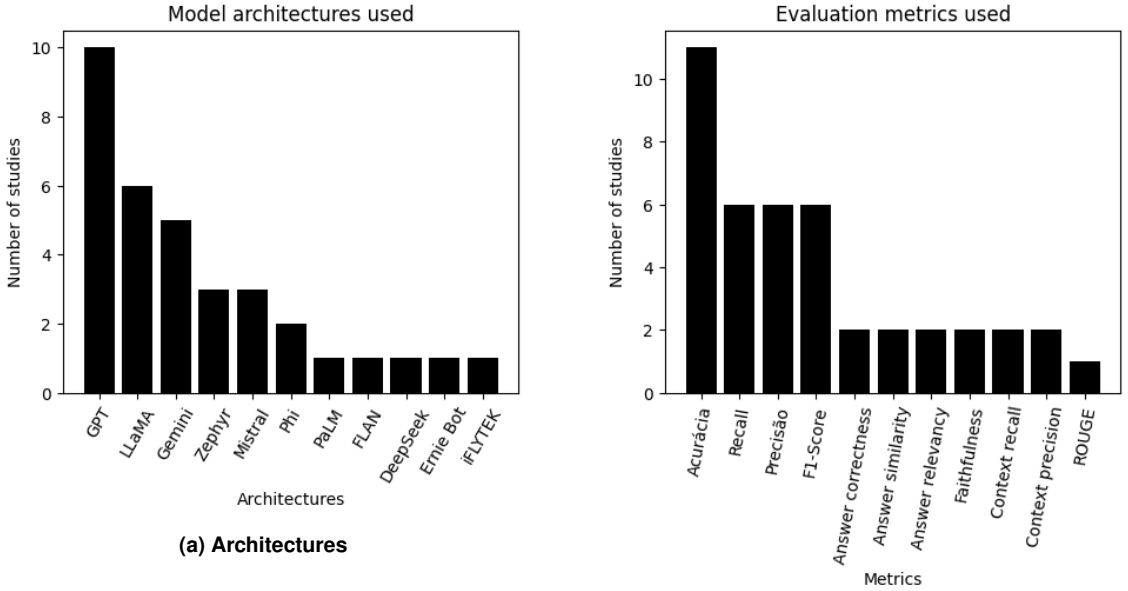
To systematically map existing strategies for optimizing language model performance, this section presents a Systematic Literature Review (SLR) of the current literature. We designed this review to investigate current approaches to optimizing accuracy and mitigating hallucinations within resource-constrained environments. Specifically, we examine how effectively context enrichment strategies substitute for brute-force parametric scaling.

The SLR was conducted following protocol, aiming to ensure methodological rigor, reproducibility, and bias reduction. The following scientific databases were consulted: IEEE Xplore; ACM Digital Library; and Web of Science. The search strategy combined terms related to language models, hallucination, accuracy, Web Scraping, and Retrieval-Augmented Generation (RAG), using Boolean operators to maximize the coverage of relevant studies[1]. The inclusion criteria prioritized studies that: used Web Scraping for context enrichment in LLMs or SLMs; employed information retrieval techniques, such as RAG, integrated with language models; and presented empirical performance evaluation. We excluded studies that did not focus on language models, omitted web scraping or RAG, or were published as literature reviews or book chapters, alongside works without full-text access. The initial corpus of 1,711 works was reduced to 1,702 after removing duplicates, and then to 147 after abstract screening. Evaluating the introductions and conclusions narrowed the selection to 42 articles; a full-text analysis against our quality criteria finalized the selection at 20 papers.[2].

[1](”AI” OR ”artificial intelligence” OR ”hallucination in AI models”) AND (”accuracy” OR ”approach” OR ”technique” OR ”method” OR ”guide” OR ”models” OR ”open issues” OR ”load mapping” OR ”open research” OR ”open problem” OR ”SLM” OR ”small language model” OR ”LLM” OR ”large language model” OR ”Retrieval-Augmented Generation” OR ”RAG”) AND (”web scraping” OR ”web crawling” OR ”data crawling”)

[2]Of the 20 selected articles, 14 are from IEEE ([Kumar et al. 2024, Saad and Qawaqneh 2024, Rathod et al. 2024, Pushpalatha and Aravindan 2025, Jervas et al. 2025, Sampaio et al. 2024,

Out of the 20 analyzed studies, 8 incorporate Small Language Models (SLMs), whereas the rest make exclusive use of Large Language Models (LLMs). The literature heavily features models based on GPT and LLaMA architectures. In terms of frequency, GPT led with 10 deployments, followed by LLaMA (6), Gemini (5), Zephyr (3), Mistral (3), Phi (2), and the remaining models (PaLM, FLAN, DeepSeek, Ernie Bot, and iFLYTEK) appeared in only one study each, as illustrated in Figure 1a. Accuracy remains the predominant evaluation metric (11 studies), often paired with precision, recall, and F1-score (6 studies) (Figure 1b). Crucially, existing comparative analyses seldom leverage formal statistical testing to validate their findings—a deficiency that undermines the reliability of reported variances and establishes the direct imperative for our experimental proposal.



(a) Architectures

(b) Metrics

Figure 1. (a) Model architectures adopted in the selected studies. (b) Performance metrics utilized in the reviewed studies.

Web scraping techniques are employed across all selected studies as the primary mechanism for external data collection. In the majority of cases, the extracted data is specifically utilized for context enrichment during the inference phase. The integration of RAG was observed in more than half of the reviewed studies, whereby dynamically retrieved documents are appended to the input context to ensure responses are grounded in real-time or domain-specific information. The documented consensus indicates that the synergy between web scraping and RAG is strongly associated with enhanced accuracy, mitigated hallucinations, superior response veracity, and specialized domain alignment. These insights reinforce that context enrichment successfully offsets the bounds of parametric knowledge in smaller architectures, anchoring the subsequent gap analysis and our proposed experiment.

Senapati et al. 2024, Chowdhury et al. 2025, Gupta et al. 2024, Porlou et al. 2024, Kanataria et al. 2024, Hachicha et al. 2024, Sriram et al. 2024, Kumbar et al. 2024]), 4 from Web of Science ([Tan et al. 2025, Salim et al. 2025, Jamil et al. 2025, Sufi and Alsulami 2025]), and 2 from ACM ([Agrawal et al. 2025, Huang et al. 2024]).

3. Experimental Methodology

Despite the success of context enrichment via web scraping and RAG, mainstream research still treats parametric scaling as the default path to higher accuracy, routinely overlooking computational overhead. Crucially, the literature lacks standardized methodologies, formal statistical validation, and evaluations under strict VRAM constraints. This leaves the critical trade-offs between model accuracy, resource costs, and scalability largely unquantified.

Addressing these research gaps, this study contributes a statistical analysis of the structural trade-off between accuracy and computational cost in small and medium-scale language models. Our empirical investigation quantifies the correlations between model size and performance, evaluates statistically significant differences across model categories, and examines the diminishing marginal returns of parametric scaling under real-world, hardware-constrained conditions (6 GB VRAM). These findings provide concrete evidence to inform the choice between structural scaling and contextual enrichment in resource-constrained scenarios.

This section describes the experimental setup adopted to quantitatively investigate the relationship between the number of parameters, accuracy, and computational cost in language models, considering real infrastructure constraints. The experiments were conducted on a machine with the following specifications: AMD Ryzen 5 5600 Processor; NVIDIA RTX 2060 GPU with 6 GB of VRAM; and 32 GB DDR4 RAM. The 6 GB VRAM limitation imposes a practical constraint: models with more than 7 billion parameters cannot be fully executed on the GPU, requiring partial offloading of layers to RAM and their execution by the CPU. This hybrid CPU-GPU environment directly conditions the execution time results reported in the following sections, being fundamental for interpreting the observed computational cost. All code used in this study is publicly available on GitHub [3].

3.1. Evaluated Models

Thirteen language models with sizes ranging from 1 to 30 billion parameters were evaluated, as listed in Table 1. The selection included dense models and Mixture of Experts (MoE) architectures which, although the total number of parameters is large, only a fraction is activated per token during inference. Models with explicit reasoning mechanisms were excluded due to the significant increase in inference time, which exceeded the temporal scope defined for the experiment.

3.2. Benchmark and Dataset

Model performance was evaluated using the Massive Multitask Language Understanding (MMLU) benchmark, widely adopted for assessing cognitive capabilities and generalization across multiple knowledge domains [Zhao et al. 2024]. Only the publicly available test split was used, totaling 14,042 multiple-choice questions distributed across various areas, including exact sciences, humanities, medicine, law, social sciences, among others. Accuracy was defined as the proportion of correct answers relative to the total number of questions. The dataset used in this study is publicly available on Hugging Face [4].

[3] Available at: `https://github.com/gustavo-ufape/slm-accuracy-cost-tradeoff`

[4] Available at: `https://huggingface.co/datasets/cais/mmlu`

3.3. Experimental Procedure

The models were executed locally through the LM Studio 0.4.1 tool, using its API for automated submission of requests via Python scripts. To ensure evaluation standardization, a deterministic prompt template was used, instructing the model to respond exclusively with the correct alternative (A, B, C, or D), without additional explanations. This standardization minimizes variability associated with text generation style. For each model, the total accuracy on the MMLU and the total execution time for processing the 14,042 questions were collected. The execution time was measured in hours and considered as a proxy for computational cost under the adopted hardware configuration.

It is important to emphasize that context-enrichment techniques, such as RAG and web scraping, were intentionally excluded from this initial stage of the study. The primary objective of this experiment was to establish a clean and controlled baseline for evaluating the structural capabilities and intrinsic knowledge of the selected language models under hardware constraints. This approach enables an accurate assessment of the trade-off between parametric scale and computational overhead. Accordingly, the integration and comparative evaluation of these models using both RAG and dynamic web scraping strategies are reserved for subsequent phases of this research.

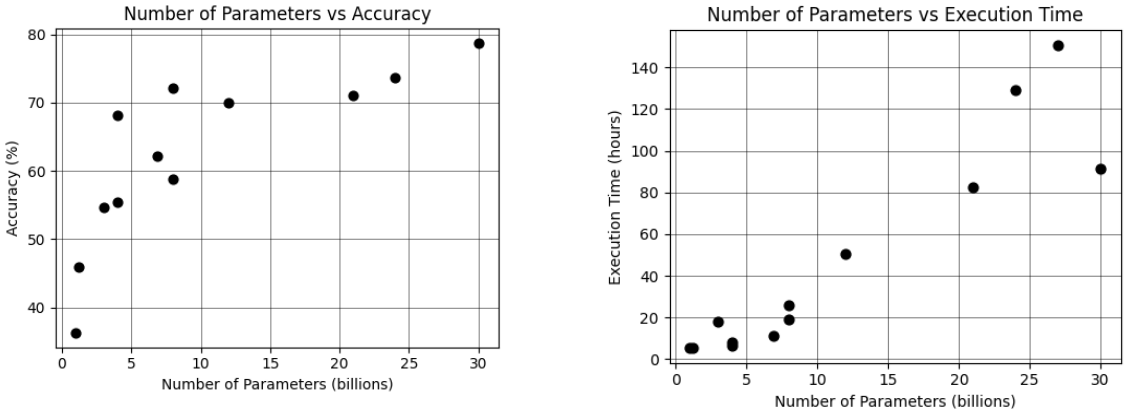
Table 1. Accuracy and execution time per model.

Model	Param. (B)	Accuracy (%)	Time (hours)	MoE	VRAM
Gemma 3 1B	1	36.276	5.48	No	782 MB
Lfm2 1.2B	1.2	45.926	5.62	No	1.25 GB
Ministral 3 3B	3	54.649	18.13	No	3 GB
Gemma 3 4B	4	55.448	8.13	No	3.6 GB
Qwen3 VL 4B	4	68.167	6.58	No	3.4 GB
Gemma 3n E4B	6.9	62.205	11.37	No	4.3 GB
Granite 3.2 8B	8	58.731	19.05	No	5.1 GB
Qwen3 VL 8B	8	72.148	26.00	No	6.3 GB
Gemma 3 12B	12	69.919	50.54	No	8.7 GB
Ernie 4.5 21B A3B	21	71.101	82.35	**Yes**	11.9 GB
Devstral Small 2507	24	73.651	129.20	No	12.3 GB
Gemma 3 27B	27	73.025	150.50	No	12.6 GB
Qwen3 VL 30B	30	78.706	91.60	**Yes**	12.1 GB

4. Results

4.1. Overall Model Performance

Table 1 summarizes the empirical results for the 13 evaluated models, detailing their parameter count, MMLU accuracy, total execution time, Mixture-of-Experts (MoE) architecture utilization, and VRAM consumption. A general trend of increasing accuracy with the growth of the number of parameters is observed, accompanied by a substantial increase in processing time, patterns that are statistically analyzed in the following subsections. Models with more than 7 billion parameters, by requiring hybrid CPU-GPU execution, exhibited execution times starting from 19 hours, while smaller models completed their execution in up to 12 hours, as shown in Figure 2b. This reflects the impact of



(a) Parameters vs accuracy.

(b) Parameters vs time.

Figure 2. (a) Graph comparing the number of parameters and the accuracy of each model. (b) Graph comparing the number of parameters and the execution time of each model.

the hardware constraints described in the methodology. These descriptive results provide an overview of the behavior of the evaluated models, serving as the basis for the statistical analyses presented in the following subsections.

4.2. Relationship between Number of Parameters and Accuracy

In order to investigate the relationship between model size and performance, the correlation between the number of parameters and the accuracy obtained on the MMLU benchmark was analyzed. Initially, Pearson's correlation coefficient was calculated, which is appropriate for evaluating linear relationships between continuous variables. The obtained value was $r = 0.778$, indicating a strong positive linear association between the number of model parameters and the achieved accuracy.

Complementarily, Spearman's correlation coefficient was calculated, which evaluates monotonic relationships independently of the functional form of the relationship between variables. The obtained value was $\rho = 0.937$, indicating an even stronger monotonic correlation between model size and performance. This result suggests that, when ordering models by the number of parameters, a consistent trend of increasing accuracy is observed.

Figure 2a presents the scatter plot between the number of parameters and accuracy, allowing visual observation of this trend. It is observed that, although there is a general trend of increasing accuracy with the growth of the number of parameters, there are variations among models with similar sizes, possibly associated with architectural differences, training strategies, or datasets used during pre-training. These results indicate that the increase in the number of parameters is strongly associated with performance gains, corroborating evidence previously reported in the literature on scaling laws in language models [Kaplan et al. 2020]. Although larger models generally perform better on the evaluated benchmark, the relationship is not strictly linear; consequently, a rigorous analysis of computational cost is required to fully characterize the trade-off between performance and practical viability.

4.3. Relationship between Number of Parameters and Computational Cost

Beyond the relationship between model size and accuracy, the impact of the number of parameters on the computational cost required to execute the benchmark was also analyzed. For this analysis, total execution time was used as a proxy for computational cost under the adopted hardware configuration. The correlation coefficients indicate a strong association between the number of parameters and processing time. Pearson's coefficient was $r = 0.933$, suggesting a high positive linear relationship between model size and the time required to process the set of questions. Complementarily, Spearman's coefficient presented a value of $\rho = 0.948$, indicating a strong monotonic relationship between the variables. These results indicate that, as the number of parameters increases, execution time grows consistently. Figure 2b presents the scatter plot between the number of parameters and the processing time observed during the experiment.

It is observed that models with a larger number of parameters demand substantially longer times to process the same set of questions. This behavior is intensified in models with more than 7 billion parameters, which could not be fully executed within the VRAM of the GPU used. In these cases, it was necessary to use partial offloading to RAM, causing some of the model's layers to be executed on the CPU. This hybrid CPU-GPU processing introduces additional latencies, contributing to the observed increase in execution time.

These results demonstrate that increasing the number of parameters, although associated with performance gains, implies significant growth in computational cost, reinforcing the need to jointly analyze accuracy and computational efficiency when evaluating language models. While the relationship between parameter count and processing time scales linearly, the same does not hold true for accuracy. Accuracy gains are marginal compared to the escalating computational cost.

4.4. Statistical Comparison between Smaller and Larger Models

In order to assess whether the performance difference between small-scale and larger language models is statistically significant, a Student's t-test for independent samples was conducted. The evaluated models were divided into two groups, based on the hardware limitation described in the experimental methodology:

- **Group 1 (Smaller models):** models with up to 7 billion parameters, capable of full execution on the available GPU;
- **Group 2 (Larger models):** models with more than 7 billion parameters, which required partial offloading to RAM and hybrid CPU–GPU execution.

The variable of interest considered in the analysis was the total accuracy obtained on the MMLU benchmark. The statistical hypotheses were defined as follows:

- H_0: there is no statistically significant difference between the mean accuracies of the two groups;
- H_1: there is a statistically significant difference between the mean accuracies of the two groups.

Considering the small sample sizes and the absence of guarantees regarding the homogeneity of variances between the groups, Welch's correction of the t-test was

adopted [Welch 1947], which does not assume equal variances between samples. The results indicated that the mean accuracy of smaller models ($\leq$ 7B parameters) was 53.78%, while models with more than 7 billion parameters presented a mean of 71.61%. Welch's t-test confirmed the performance disparity between the groups ($t = -3.32$; $p < 0.05$), resulting in the rejection of the null hypothesis (H_0). This finding statistically confirms that larger models maintain higher accuracy.

4.5. Analysis of Marginal Accuracy Gains

In order to analyze how accuracy evolves as a function of the increase in the number of parameters, a descriptive analysis of marginal performance gains between consecutive models ordered by size was conducted. Initially, the models were organized according to the number of parameters, and the accuracy difference between subsequent models was calculated, representing the raw marginal gain.

Since the parameter increments between models are not uniform, the accuracy gains were normalized by the parametric increase. The accuracy gain per billion parameters metric is expressed by Equation 1:

$$\text{Gain per Billion (\%/B)} = \frac{\Delta\text{Accuracy}}{\Delta\text{Parameters}} = \frac{\text{Accuracy}_i - \text{Accuracy}_{i-1}}{\text{Parameters}_i - \text{Parameters}_{i-1}} \quad (1)$$

where i represents the current model in the ordered sequence and $i - 1$ represents the immediately smaller model. For models with the same number of parameters, the average accuracy was used, as occurs with models of 4 and 8 billion parameters, thus transforming into a model of 4 billion parameters with an accuracy of 61.808% and a model of 8 billion parameters with an accuracy of 65.44%.

Figure 3 (and Table 2) illustrates the relationship between the normalized marginal accuracy gain and the number of parameters. From the analysis, it is observed that the largest accuracy gains occur in smaller models, reaching peaks of up to 48% gain per billion parameters (bp) added (from 1 to 1.2 bp). As the parametric scale grows, accuracy increments become progressively smaller. This behavior characterizes a clear pattern of diminishing marginal returns, where the additional benefit of structural scaling diminishes significantly. These results demonstrate that expanding the size of large models requires a disproportionate computational cost relative to the modest accuracy gains achieved in environments with hardware constraints.

Table 2. Normalized marginal accuracy gains.

Parameters (B)	**1.2**	**3**	**4**	**6.9**	**8**	**12**	**21**	**24**	**27**	**30**
Accuracy (%)	48.25	4.84	7.15	0.13	2.94	1.11	0.13	0.84	-0.2	1.89

5. Conclusion

The results reveal a critical trade-off between accuracy and computational cost: although Welch's t-test confirms that models with more than 7 billion parameters exhibit significantly higher mean accuracy than smaller ones (71.61% vs. 53.78%; $p = 0.0117$), this

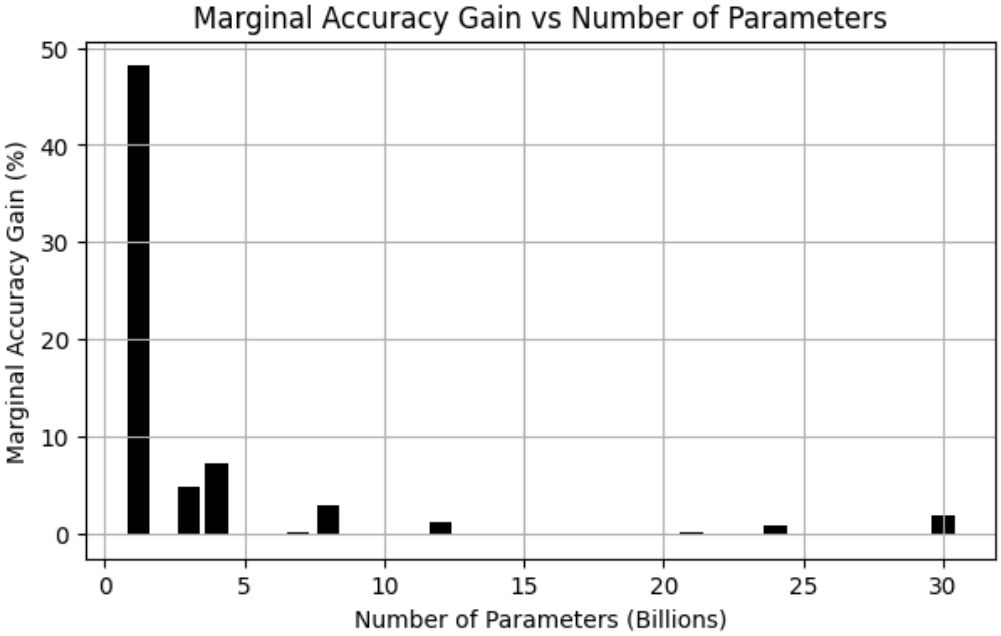


Figure 3. Normalized marginal accuracy gain as a function of the number of parameters.

gain is accompanied by a steep increase in execution time, intensified by the hybrid CPU–GPU processing required by these models under the adopted hardware constraints. Therefore, continuous structural scaling demonstrates limited practical viability in environments with infrastructure constraints.

This scenario is aggravated by the pattern of diminishing marginal returns identified in the analysis of Section 4.5: the largest accuracy gains are concentrated in smaller models, while additional parametric increments produce progressively smaller performance improvements. In other words, the computational cost grows at a much faster rate than accuracy, making structural scaling a cost-benefit unfavorable strategy as models become larger.

By cross-referencing these empirical findings with those from the Systematic Literature Review, it is established that isolated structural scaling is not necessarily the most efficient strategy for improving models in computationally constrained scenarios. The SLR demonstrated that context enrichment techniques (Web Scraping and RAG) are consistently associated with increased accuracy, reduced hallucinations, and greater domain specialization, without the burden of parametric increase. Approaches that integrate SLMs with these mechanisms therefore emerge as alternatives with a substantially more favorable cost-benefit ratio.

Consequently, for real-world applications that require local execution, strict data control, or operation on limited infrastructures, SLMs reinforce their strategic relevance. The findings of this study indicate that, instead of focusing exclusively on the continuous increase of parameters, it becomes essential to investigate hybrid architectures capable of combining the efficiency of smaller models with the dynamic retrieval of updated external information.

Among the limitations of this work, there was the hardware constraint available for the experiment, limiting models to up to 30 billion parameters, in addition to using a single benchmark and not employing context enrichment techniques identified in the literature review. Consequently, for future developments, the empirical investigation of the integration of SLMs with RAG strategies, dynamic data collection from Internet websites, and the expansion of the evaluation to multiple benchmarks is proposed.

Finally, the authors acknowledge the use of Google's Gemini model strictly for language revision, stylistic editing, and translation assistance during the preparation of this manuscript. All conceptualization, experimental design, empirical analysis, and interpretation of the results were conducted entirely by the authors, who maintain full responsibility for the content and conclusions presented.

References

Agrawal, R., Asrani, M., Youssef, H., and Narayan, A. (2025). Scmrag: Self-corrective multihop retrieval augmented generation system for llm agents. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Detroit, MI, USA. International Foundation for Autonomous Agents and Multiagent Systems.

Agustiani, A. D., Putri, S. M., Hutahaean, J., Sholahuddin, M. R., Alifi, M. R., and Hodijah, A. (2026). Evaluating rag performance on small language models for low-resource devices through chunking and retrieval methods. *Jurnal Online Informatika*, 11(1):45–56.

Chowdhury, M. D. T. B. Z., Hossain, M., and Islam, M. R. (2025). Durghotona gpt: A web scraping and large language model based framework to generate road accident dataset automatically in bangladesh. *arXiv preprint*, abs/2504.21025.

Desislavov, R., Martínez-Plumed, F., and Hernández-Orallo, J. (2023). Trends in ai inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems*, 38:100857.

Filho, J. F. and Fernandes, S. (2025). Do slm agents screen papers as well as llms? In *Anais do XXII Encontro Nacional de Inteligência Artificial e Computacional*, pages 592–603, Porto Alegre, RS, Brasil. SBC.

Ganesh, P., Goel, K., and McCallum, A. (2023). Small language models are also few-shot learners. *arXiv preprint arXiv:2306.07930*.

Gupta, D., Suganya, R., Bhoji, S., Geetha, S., Arora, M., and Manjula, D. (2024). Enhanced research on legal data extraction and document analysis tool. In *Proceedings of the 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, pages 756–763. IEEE.

Hachicha, M., Omezine, M. A., Zghal, M. K., Riahi, M. H., and Ncib, L. (2024). Recommendation module based on web scraping and llm models with the rag technique. In *Proceedings of the Artificial Intelligence Revolutions (AIR)*. IEEE.

Hoffmann, J. et al. (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Huang, Z., Tang, J., Karir, M., Liu, M., and Sarabi, A. (2024). Analyzing corporate privacy policies using ai chatbots. In *Proceedings of the 2024 ACM on Internet Measurement Conference (IMC '24)*, pages 505–515, Madrid, Spain. Association for Computing Machinery.

Jamil, H., Noorosmawie, A. B., Rabu, H. W., and Razak, L. A. (2025). From archives to ai: Residential property data across three decades in brunei darussalam. *Data in Brief*, 60:111505.

Jervas, S., Jacob, C., Sundar, S., and Varma, D. P. (2025). Context-driven chatbot development: Leveraging zephyr-7b with rag for improved response accuracy. In *Proceedings of the Emerging Technologies for Intelligent Systems Conference*. Conference paper, Feb. 2025.

Kanataria, N., Goel, P., Patel, K. P., Patel, K., Patel, H. N., and Garg, D. (2024). Rag-enhanced large language model for intelligent assistance from web-scraped data. In *Proceedings of the 9th International Conference on Communication and Electronics Systems (ICCES)*. IEEE.

Kaplan, J., McCandlish, S., Henighan, T., et al. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Kumar, K. S., Srihari, P., and Raman, C. J. (2024). Ai for career growth: Advanced resume analysis and linkedin scraping for personalized job recommendations. In *Proceedings of the 2nd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS-2024)*, pages 1287–1293, Chennai, India. IEEE.

Kumbar, P., Sharma, S., Batra, I., Malik, A., and Ashfaq, F. (2024). Storyspinner: Ai based technique to narrate the web links. In *Proceedings of the International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*. IEEE.

Liu, H., Meng, Y., Zhong, Y., Zhao, C., and Han, J. (2024). Small language models: Survey, measurements, and insights. *arXiv preprint arXiv:2401.03720*.

Oliveira, J., Rufino, N., Levcovitz, S., Utino, M., Lobato, F., Turine, M., Rezende, S., Junior, A. J., and Marcacini, R. (2025). Agents4gov: Privacy-preserving web agents using open llms for public sector tasks. In *Anais do XXII Encontro Nacional de Inteligência Artificial e Computacional*, pages 580–591, Porto Alegre, RS, Brasil. SBC.

Porlou, C., Makrynioti, M., Alexiadis, A., Stavropoulos, G., Pantelis, G., Votis, K., and Tzovaras, D. (2024). Optimizing an llm prompt for accurate data extraction from firearm-related listings in dark web marketplaces. In *Proceedings of the IEEE International Conference on Big Data*. IEEE.

Prucs, Á., Csutora, M., Antal, M., and Marosi, M. (2025). Compute-accuracy pareto frontiers for open-source reasoning large language models. *arXiv preprint*.

Pushpalatha, M. and Aravindan, M. S. (2025). Comparative analysis of web scraping methodologies using generative ai. In *Proceedings of the 6th IEEE International Conference on Recent Advances in Information Technology (RAIT)*. IEEE.

Rathod, M., Yadav, A., Phutane, S., and Bhise, S. (2024). Comment compass: An approach to analyze youtube comments through web scraping, sentiment analysis, and

generative ai for actionable insights. In *Proceedings of the 2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT 2024)*. IEEE.

Saad, M. and Qawaqneh, Z. (2024). Closed domain question-answering techniques in an institutional chatbot. In *Proceedings of the 2024 4th International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME 2024)*, Male, Maldives.

Salim, M. S., Hossain, S. I., Jalal, T., Bose, D. K., and Basher, M. J. I. (2025). Llm based qa chatbot builder: A generative ai-based chatbot builder for question answering. *SoftwareX*, 29:102029.

Sampaio, T., Oliveira, P. F., and Matos, P. J. (2024). Development of a chatbot to support an university institutional website. In *Proceedings of the 2024 International Conference on Engineering and Emerging Technologies (ICEET)*.

Senapati, A., Ananya, J., Jha, A., Mahishi, A., and Srinivas, K. S. (2024). Driving innovation: Creating a dataset for automotive ad campaign analysis. In *Proceedings of the 2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT)*, pages 273–278. IEEE.

Sriram, G., Suji, A. R., Nivitha, K., Rahul, S., Nallusamy, P., and Arun, A. (2024). Scrapetalk: Chatbot conversation with web scraped insights. In *Proceedings of the International Conference on Integration of Emerging Technologies for the Digital World (ICIETDW)*. IEEE.

Sufi, F. and Alsulami, M. (2025). Disaster in the headlines: Quantifying narrative variation in global news using topic modeling and statistical inference. *Mathematics*, 13(13):2049.

Tan, D., Huang, Y., Liu, M., Li, Z., Wu, X., and Huang, C. (2025). Identification of online health information using large pretrained language models: Mixed methods study. *Journal of Medical Internet Research*, 27:e70733.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Welch, B. L. (1947). The generalization of 'student's' problem when several different population varlances are involved. *Biometrika*, 34(1-2):28–35.

Zhao, Q., Huang, Y., Lv, T., Cui, L., Wei, F., et al. (2024). Mmlu-cf: A contamination-free multi-task language understanding benchmark. *arXiv preprint arXiv:2412.15194*.