Exploration of the Efficiency of SLM-Enabled Platforms for Everyday Tasks

Volodymyr Rusinov and Nikita Basenko
Faculty of Informatics and Computer Engineering, Igor Sikorsky Polytechnic Institute, Beresteiska Avenue 37, 03056 Kyiv, Ukraine
v.rusinov.io11f@kpi.ua, basenko.nikita@lll.kpi.ua

Keywords: AI, SLM, IoT, NLP, Model Evaluation Metrics.

Abstract: This study explores the potential of Small Language Models (SLMs) as an efficient and secure alternative to larger models like GPT-4 for various natural language processing (NLP) tasks. With growing concerns around data privacy and the resource-intensiveness of large models, SLMs present a promising solution for research and applications requiring fast, cost-effective, and locally deployable models. The research evaluates several SLMs across tasks such as translation, summarization, Named Entity Recognition (NER), text generation, classification, and retrieval-augmented generation (RAG), comparing their performance against larger counterparts. Models were assessed using a range of metrics specific to the intended task. Results show that smaller models perform well on complex tasks, often rivalling or even outperforming larger models like Phi-3.5. The study concludes that SLMs offer an optimal trade-off between performance and computational efficiency, particularly in environments where data security and resource constraints are critical. The findings highlight the growing viability of smaller models for a wide range of real-world applications.

1 INTRODUCTION

In the period of large language models (LLM) dominance in the market, some companies are thinking about the security of their data when developing their own solutions using open source LLM application programming interfaces(API). Data leaks, identity theft and other types of malicious activity may present a challenge when using these solutions, as company data may contain sensitive information such as personal documents, financial reports and so on. Given the specifics of the business area, companies are enticed to train their own custom LLMs or provide data to outside vendors for them to train the models, facing the same obstacles described earlier. [1] In addition, such systems require computational power that require outsourcing LLMs into Cloud, which cannot function without the Internet, as they send requests to APIs, saving the computing resources of the device.

It is also worth noting that sometimes the use of large models (such as GPT-4o, Claude Sonnet-3.6 or even Llama-70b) for simple tasks such as text classification or answers to household questions. This leads not only to an increase in the cost of such a system, but also to the execution time, which negatively affects the user experience directly.

2 OBJECTIVE AND TOOLS

This study aims to evaluate the efficacy of Small Language Models (SLMs) as viable alternatives to larger models for specific natural language processing tasks in resource-constrained environments. The primary challenge lies in applying such models to a range of tasks commonly employed by researchers to enhance productivity or automate routine processes. For instance, in fields like healthcare, where data security is critical, reliance on large corporations for data management may be inappropriate due to confidentiality concerns. It is therefore necessary to identify the core tasks that small language models (SLMs) must be capable of performing to effectively address the demands of contemporary research.

To comprehensively evaluate SLM performance, we have identified five core NLP tasks that represent common requirements in research and practical applications:

1) Translation - The conversion of text between major world languages, representing a fundamental yet well-defined linguistic task that serves as a baseline for model linguistic capabilities.

2) Summarization - The extraction and condensation of key information from longer texts, requiring semantic understanding and content prioritization abilities.
3) Named Entity Recognition (NER) - The identification and classification of named entities within text, representing a structured information extraction task that evaluates the model's ability to recognize semantic patterns.
4) Text Classification - The categorization of text into predefined or custom classes, requiring the model to understand context and apply flexible classification schemas.
5) Retrieval-Augmented Generation (RAG) - The integration of external context into generated responses, evaluating the model's capacity to process supplied information and incorporate it into outputs.

These tasks were selected to provide a comprehensive assessment of both fundamental and advanced linguistic capabilities required in contemporary research and practical applications.

The selection of models for evaluation was governed by the following criteria. The first one is the limitation of the number of parameters and the size of the model. In this research small models up to 2B parameters are observed, because such models are small enough to be able to run locally on any research hardware without affecting system performance, in addition chosen models should be up to 2.5GB. It is important to consider not only the average value but also specific metrics, such as IFEval (instruction-following evaluation, used for evaluation model's capabilities to follow instructions), MUSR (multistep soft reasoning, used for understanding large contexts and reasoning) and not necessarily but still it would be good BBH (big-bench Hard, used for general understanding of the world).

The evaluation incorporates multiple model families, including fine-tuned variations of Llama, Qwen [11] and its variations and also a bunch of different non NLP related models (different 3d, 2d visualizations which are not considered in this article). [13] And the SmolLM [1] family, whose capabilities prompted this investigation. Additionally, the Phi-3.5 model from Microsoft was incorporated to serve as a comparative benchmark against the other smaller models. Each model will be systematically evaluated across all six tasks using task-specific metrics that accurately reflect performance quality. The evaluation will be conducted within a controlled environment using identical prompts, configuration parameters, and system instructions to ensure comparative validity.

The experimental setup will utilize the distributed AI platform described in Section 3, which enables efficient deployment and testing of multiple models simultaneously while maintaining consistent evaluation conditions.

3 PLATFORM

For this study, a distributed AI platform was implemented using a network of Raspberry Pi devices, interconnected in a tree topology augmented with additional inter-level connections following a De Bruijn sequence. This configuration, combined with an MLOps (Machine Learning Operations) process, provides an efficient and cost-effective setup for deploying and testing SLMs for the purposes of this research. The goal is to utilize the advantages of the embedded systems and locally deployable nature of Raspberry Pi hardware to evaluate the performance of SLMs across a range of natural language processing (NLP) tasks, while ensuring efficient model training and deployment workflows. The platform is organized in a tree topology with additional DeBrujin connections, where one central node (managed by Cloud in this case) managed the coordination of tasks, while multiple Raspberry Pi devices handled SLM tasks. [6] This hierarchical structure allowed the system to scale easily, with additional nodes onboarding different models, demonstrated in this work (Fig. 1). This approach showcases the decentralized nature of Edge computing, with local deployment, which is beneficial for the systems that utilize SLMs.

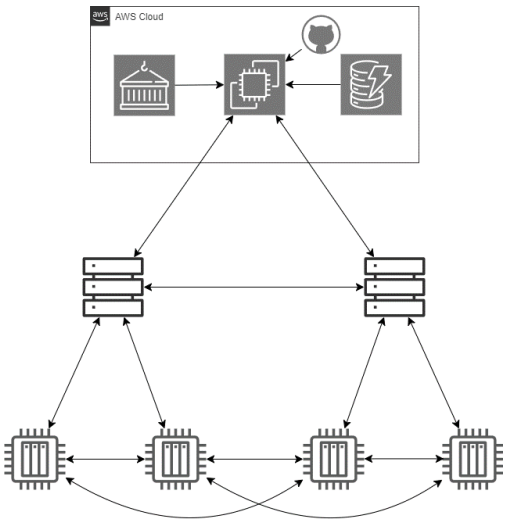

Figure 1: Example of an AI platform using the described approach.

In order to use this system optimally, MLOps is used in streamlining the machine learning lifecycle within this platform. The CI/CD pipeline was responsible for automating the workflow of data preparation, model training, validation, deployment, and monitoring. Specifically, these tools are used to automate the deployment of SLMs across the Raspberry Pi devices, ensuring that necessary updates are propagated across the network. In addition, this platform provides a framework for continuous integration and continuous delivery, allowing the new models to be iteratively improved based on real-time performance feedback and continuous training of SLMs. Each Raspberry Pi node was equipped with a lightweight containerized environment, which allowed for the isolated execution of different SLMs. This containerization ensured that each model could run independently on the devices without interfering with other processes, optimizing the overall performance and stability of the system. The nodes communicated through a message-passing interface, enabling them to share intermediate results, such as model predictions, across the distributed network in real time.

The distributed setup, coupled with the MLOps process, allowed for efficient parallel processing during model evaluation. Moreover, this process enabled continuous monitoring of the models' performance, with real-time metrics being collected and analyzed. This helped ensure that the models were functioning as expected and allowed for quick troubleshooting in the event of model drift or errors. In order to leverage the benefits of using such approach and to test new model parameters or sequences, LM Studio is used for the experiments. LM studio is integrated through the API with the user interface, furthermore, it allows for simple integration into continuous training process, which is beneficial for the platform's ability to rapidly reconfigure itself in various scenarios. The ability to quickly iterate on models and deploy them across the Raspberry Pi network made it possible to optimize their performance on specific tasks, all while maintaining a low resource footprint. For example, while one node handles a translation task, another could simultaneously work on summarization, and a third could be responsible for Named Entity Recognition (NER). This parallel processing capability significantly reduced the overall execution time for testing multiple models on various tasks, making the setup highly efficient and scalable for handling large volumes of data.

4 EXPERIMENTS

In order to test such models it is necessary to collect some data from these tasks. Additionally, all models were tested using the platform described in the previous paragraph, which is specifically designed for rapid reconfiguration and deployment of various models.

The same parameters (numbers – refer to screenshot 15.01) such as Seed value, temperature, context size were set for all models (see Table 1 for model specifications). The models were not provided with any prior interaction history, as the evaluation was conducted without the use of contextual cues or supplementary prompts. Additionally, all models were provided with the same system prompt, which is default for all models.

Table 1: Model overview.

Model name	Number of parameters	Model size	Quantization
google/gemma-2-2b-it	2b	1.52GB	q4
Qwen/Qwen2.5-1.5B-Instruct	1.54b	1GB	q4
meta-llama/Llama-3.2-1B-Instruct	1.24b	1.23GB	q4
Qwen/Qwen2.5-0.5B-Instruct	0.494b	0.65GB	q8
HuggingFaceTB/SmolLM2-1.7B-Instruct	1.71b	1GB	q4
HuggingFaceTB/SmolLM2-360M-Instruct	0.362b	0.7GB	q16
microsoft/Phi-3.5-mini-instruct	3.82b	2.23GB	q4

4.1 Model Comparison

In this section, we compare different models using a set of metrics, all specific to each task.

Translation. In this case, the reference translation of the text was compared with the translations generated by the models. For evaluation, COMET (Crosslingual Optimized Metric for Evaluation of Translation), a state-of-the-art tool for the automatic assessment of machine translation quality [2], was utilized. This metric is particularly appropriate, as it is grounded in a pre-trained XLM-RoBERTa language model, which facilitates the evaluation of both linguistic accuracy across multiple languages and the contextual coherence of the translated text. This metric was applied to the outputs generated by our models, yielding the following results (Fig. 2).

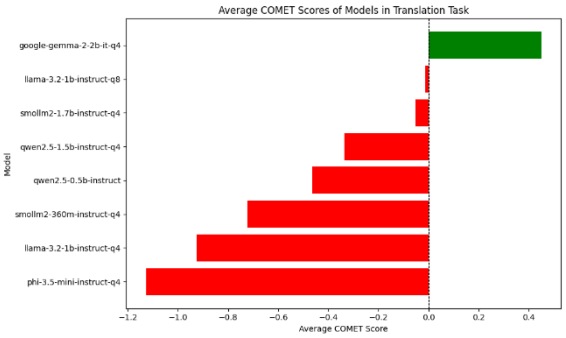

Figure 2: Comparison between different SLMs using COMET.

As shown on the graph (Fig. 2.), the model developed by Google outperforms the others, which can be attributed to the extensive multilingual training data used for the Gemma model family. This training approach enabled the model to achieve higher performance in understanding a wide range of languages, particularly those that are most commonly used. The worst values were obtained from the Phi-3.5 model because of its tendency to write verbose explanations, when it was not inquired to do so. In addition, this model tends to constantly write what words it translated and why. This affects both the evaluation and the user experience. The difference between variying degrees of quantization of Llama-3.2-1b models is evident, wtih int4 performing significantly worse than int8.

The largest model Phi-3.5 has a peculiarity to describe additional information, when it is not explicitly asked for.

Summarization In this traditional task, we will evaluate the models using the BERTScore metric, which, similar to COMET, leverages pre-trained language models such as BERT (Bidirectional Encoder Representations from Transformers) to assess the semantic similarity between the predicted text and the reference text. Because of the bidirectional representation of words and the transformation of texts into a vector space, BERTScore helps to better determine the degree of semantic similarity between texts. [3] Below are the results of the models in our tests:

When analyzing the test results (Fig. 3) it is noted that all the tested models are highly efficient for the explored tasks. It is worth mentioning that the SmolLM2 family models - both models (larger and medium sized), were able to perform significantly better than the largest model among all presented - Phi-3.5. It can also be concluded that for this task, smaller models such as SmolLM2-360m are more effective, as they make optimal use of the device's computing power—an advantage that is less prominent in the Phi-3.5 model.

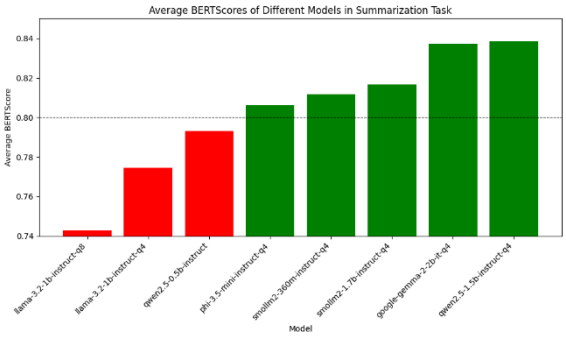

Figure 3: Comparison between different SLMs using BERT.

NER. In this instance, the models were evaluated based on their prediction accuracy using the following (1), which incorporates both the incompleteness of the model's response and a penalty for failure to adhere to the provided instructions:

$$Rbase=(Correct + 0.5 * Partial) / TotalTrueEntities. \quad (1)$$

In this equation, Correct - fully correct entities, Partial - partially correct entities. - TotalTrueEntities - total number of true entities.

Based on this (2), the models are evaluated:

$$Rfinal = if(ExtraSymbols = "✓", Rbase / 2, Rbase), \quad (2)$$

where ExtraSymbols indicates the presence or absence of additional irrelevant content in the model's response, and Rbase represents the base accuracy value calculated from (1). The final rating (Rfinal) is penalized by a factor of 2 when the model's output contains extraneous symbols or information not requested in the query. This penalty mechanism was implemented to quantitatively account for the models' adherence to the instruction constraints, as superfluous output can negatively impact both computational efficiency and user experience in practical applications. This approach allows for a more comprehensive evaluation of model performance beyond mere entity recognition accuracy, incorporating the quality and precision of the generated output as essential evaluation criteria. A notable result of the study is the unexpectedly high performance of the Qwen-2.5-0.5b and SmolLM2-360m models, which demonstrated exceptional performance on a task that is traditionally considered

to be more complex for models with a limited number of parameters. This result is of particular interest because it contradicts the commonly held assumption of a direct correlation between model size and the quality of its performance on complex linguistic tasks.

In the context of this study, the use of Large Language Models (LLMs) demonstrates significant advantages over traditional natural language processing tools such as Spacy. The key difference lies in the quality and structured nature of the output data. While LLMs generate logically organized and easily human interpretable output, Spacy-based solutions, despite their transformational architecture, often provide less coherent output consisting of a set of semantically similar tokens. This observation highlights the superiority of LLM in tasks requiring not only processing accuracy but also human-readable results.

Classification. This classification task stands out from the classic approach by employing custom user classes. User classes are named, defined and modified by the user. Accuracy metric is used as an expected model output, formatted either solely as a class or a structured output defining a class. Therefore, accuracy metric is the best fit for this task and it accurately displays the classification task.

$$Accuracy = (Number\ of\ Correct\ Predictions) / (Total\ Number\ of\ Predictions) \quad (3)$$

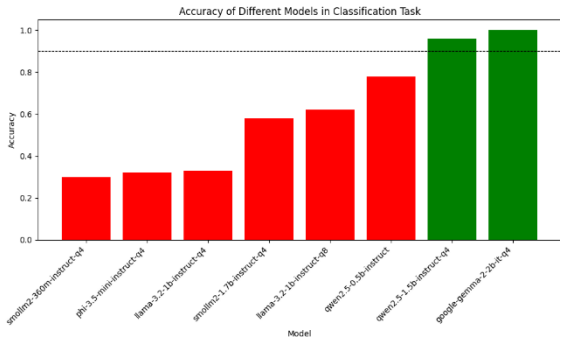

Figure 4: Comparison between different SLMs using Accuracy metric for classification task.

The task itself is complicated by the fact that the model, as previously mentioned in the earlier example, must guess a custom category based on the custom description. Figure 4 illustrates the performance comparison of different SLMs in this classification task using the Accuracy metric. These categories can be different and the text can be inexplicable. Gemma predicted all the test cases, which is not the case with the latest models. Phi-3.5 and Llama - like to hallucinate, giving a non-existent category or return the prompt itself that it was provided with. It is possible that by slightly modifying the prompt or system instructions, the performance of Qwen2.5-0.5b could be improved, potentially allowing it to achieve results comparable to its older counterpart or even outperform Gemma.

RAG. This task is evaluated using the RAGAS (Retrieval Augmented Generation Assessment) estimation method, which is based on model evaluation by means of a larger model. In a similar way, LLM models are trained by using a model-evaluator to evaluate the correctness of the model. The RAGAS approach in this experience evaluates any task but in this case it is applied to RAG tasks to observe how well the model responds. [4] The model estimator is openai-4o. The accuracy metric employed is implicitly consistent, assigning a value of 0 for incorrect responses and 1 for correct responses. The results of the evaluation can be observed below:

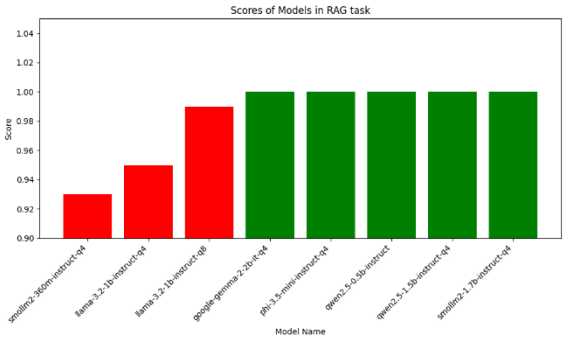

Figure 5: Comparison between different SLMs using RAGAS approach.

The analysis of the results (Fig. 5) demonstrates that all the models under study are capable of achieving a high level of performance. Special attention should be paid to the efficient performance of small-size models in particular Qwen2.5-0.5b and Llama-3.2-1b. Regarding the SmolLM-0.36b model, it is important to highlight its important characteristic: rather than providing short answers, it tends to reproduce a portion, or, in some cases, the entire input query within its own responses.

5 CONCLUSIONS

Based on a comprehensive analysis of the obtained results, conclusion can be reached that the Gemma-2-2b-instruct model has performed better than the other researched models, taking into account that it has the

largest parametric characteristics among all the studied models. However, when considering the efficiency of computational resource utilization, special attention should be paid to the Qwen-2.5 series model (in particular, the version with 0.5 billion parameters), which demonstrates an optimal ratio between performance and resource intensity, despite the fact that it does not reach the maximum performance in terms of absolute metrics in this study.

A notable aspect of the study is the performance of the Phi-3.5 model, which, despite its larger parametric characteristics and theoretical potential to perform more efficiently, did not demonstrate the expected superiority over smaller models. Contrary to initial assumptions that this model was expected to deliver qualitatively superior results due to its extended architecture, its actual performance was significantly below the leading position in the comparison table. This observation underscores the important conclusion that increasing the size of a model does not always correlate directly with improved performance in specific applications.

REFERENCES

[1] "SmolLM - blazingly fast and remarkably powerful," Feb. 2024, [Online]. Available: https://huggingface.co/blog/smollm.

[2] R. Rei, C. Stewart, A. C. Farinha, and A. Lavie, "COMET: A Neural Framework for MT Evaluation," arXiv.org, Sep. 18, 2020, [Online]. Available: https://arxiv.org/abs/2009.09025.

[3] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," arXiv.org, Apr. 21, 2019, [Online]. Available: https://arxiv.org/abs/1904.09675.

[4] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated Evaluation of Retrieval Augmented Generation," arXiv (Cornell University), Jan. 2023, [Online]. Available: https://doi.org/10.48550/arxiv.2309.15217.

[5] A. Grattafiori et al., "The Llama 3 herd of models," arXiv.org, Jul. 31, 2024, [Online]. Available: https://arxiv.org/abs/2407.21783.

[6] V. Rusinov, O. Honcharenko, A. Volokyta, H. Loutskii, O. Pustovit, and A. Kyrianov, "Methods of topological organization synthesis based on tree and dragonfly combinations," in Lecture notes on data engineering and communications technologies, 2023, pp. 472-485, [Online]. Available: https://doi.org/10.1007/978-3-031-36118-0_43.

[7] G. Team et al., "Gemma 2: Improving open language models at a practical size," arXiv.org, Jul. 31, 2024, [Online]. Available: https://arxiv.org/abs/2408.00118.

[8] A. Yang et al., "QWen2.5 Technical Report," arXiv.org, Dec. 19, 2024, [Online]. Available: https://arxiv.org/abs/2412.15115.

[9] "smollm/text/README.md at main · huggingface/smollm," GitHub, [Online]. Available: https://github.com/huggingface/smollm/blob/main/text/README.md.

[10] M. Abdin et al., "PHI-3 Technical Report: A highly capable language model locally on your phone," arXiv.org, Apr. 22, 2024, [Online]. Available: https://arxiv.org/abs/2404.14219.

[11] J. Bai et al., "Qwen Technical Report," arXiv.org, Sep. 28, 2023, [Online]. Available: https://arxiv.org/abs/2309.16609.

[12] Z. Wang et al., "Re-TASK: Revisiting LLM Tasks from Capability, Skill, and Knowledge Perspectives," arXiv.org, Aug. 13, 2024, [Online]. Available: https://arxiv.org/abs/2408.06904.

[13] J. Zhou et al., "Instruction-Following evaluation for large language models," arXiv.org, Nov. 14, 2023, [Online]. Available: https://arxiv.org/abs/2311.07911.