Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 111–119
DOI: 10.15439/2025F6165
ISSN 2300-5963 ACSIS, Vol. 43

Comparison of Large Language Models Supporting the Polish Language in Terms of Faithfulness in Retrieval-Augmented Generation Applications

Marcin Blachnik, Jakub Chmielewski
0000-0003-3336-4962, –
Silesian University of Technology
ul. Akademicka 2A, 44-100 Gliwice, Poland
Email: marcin.blachnik@polsl.pl

***Abstract*—This article presents an evaluation of Large Language Models with support for the Polish language, focusing on their ability to accurately extract detailed information embedded within input text called Faithfulness. This scenario reflects a typical use case in Retrieval-Augmented Generation systems, where precise factual recall is critical. For this purpose, a modified needle-in-a-haystack test was conducted, in which all queries targeted numerical values concealed within extended textual contexts. The evaluation was based on recent reports from Poland's Central Statistical Office (GUS), ensuring that the content was not included in the training data of the evaluated models.**

The results demonstrate that the best-performing model was NeuralDaredevil-8B-Abliterated, followed by PLLuM-12B-instruct and Bielik-11B-v2.3-Instruct. Notably, the error rate of NeuralDaredevil-8B-Abliterated was approximately half that of the second- and third-ranking models, marking a significant performance gap. The article also explores potential explanation for these discrepancy.

I. Introduction

ONE of the prominent and increasingly common applications of large language models (LLMs) is the integration with external information retrieval systems through a method known as Retrieval-Augmented Generation (RAG). RAG bridges the capabilities of LLMs with classical retrieval-based techniques, forming a hybrid system that combines the flexibility of generative models with the precision of structured data access. In this approach, a user query submitted to the RAG system is first processed by an information retrieval component, which selects the most relevant source documents from a predefined knowledge base. These retrieved documents are then appended to the original query and collectively passed to the language model. As a result, the model's response is informed not only by the user's input but also by the curated external content, effectively enriching the context provided to the LLM.

This architecture addresses several key challenges associated with the practical deployment of LLMs in business and industrial settings. One significant advantage of RAG is its ability to provide access to up-to-date and domain-specific information without requiring time-consuming and resource-intensive model retraining. Furthermore, RAG supports fine-grained permission control, allowing systems to retrieve only those documents that a user is authorized to access—thus aligning with enterprise data governance requirements. Perhaps most critically, RAG contributes to reducing the incidence of hallucinations—a common issue in generative models wherein outputs are plausible-sounding but factually incorrect. By grounding responses in retrieved, authoritative documents, RAG enhances the factual accuracy and reliability of generated outputs, making it a valuable framework for knowledge-intensive tasks across domains.

A significant challenge in the deployment of large language models (LLMs) lies in the limited availability of models that support the Polish language. This issue is particularly acute in business and institutional contexts, where data privacy and security requirements often necessitate on-premise deployment of LLMs rather than reliance on commercial cloud-based solutions. Despite the growing ecosystem of open-weight LLMs developed by leading research consortia—such as LLAMA, MISTRAL or QWEN—these models typically lack adequate support for Polish, making them less suitable for direct application in Polish-language tasks without additional fine-tuning or adaptation.

Although the Hugging Face platform offers a diverse collection of LLMs fine-tuned for the Polish language, preliminary investigations reveal that only a small subset of these models demonstrate the reliability and performance required for real-world applications. Many available models exhibit functional limitations, including issues such as repetitive text generation or outright failure to produce coherent outputs. These limitations underscore the need for a systematic evaluation of Polish-capable LLMs, particularly those with open weights suitable for secure, local deployment.

To address this gap, a curated selection of ten models was identified for detailed evaluation. These include four models with a context window of 4096 tokens: Bielik-11B-v2.3-Instruct, Bielik-7B-v0.1, trurl-2-13b, and gpt-3.5-turbo-instruct; five models with an extended context window of 8192 tokens: NeuralDaredevil-8B-Abliterated, Llama-3-8B-Omnibus-1-PL-v01-INSTRUCT, Kruk-7B-SP-001, Starling-

Topical area: Advanced Artificial Intelligence in Applications

LM-7B-alpha, and OpenChat-3.5-0106-Gemma; and one model, PLLuM-12B-instruct, supporting a 128k token context window. This selection serves as the foundation for comparative analysis in terms of stability and accuracy of identifying the detailed knowledge expressed in Polish-language.

Various methods for evaluating large language models (LLMs) have been proposed in recent literature [1]. However, when it comes to Retrieval-Augmented Generation (RAG) applications, the evaluation task often becomes more nuanced. In such scenarios, the primary objective is to assess the model's ability to extract specific and precise information embedded in a larger body of text—particularly when the required detail may be a minor element, potentially buried within an extensive document. To verify performance under these conditions, a modified version of the "needle in a haystack" test [2] was employed in this study. That allows for verification of the quality of the detailed knowledge extraction mechanism implemented in the LLMs measuring the Faithfulness of the model.

The article is structured as follows. The next section provides a short overview of LLM's evaluation techniques, then in section III the modified "needle in a haystack" test is described. Next in section IV, the details on the performed experiments are provided, and in the following section, the obtained results are discussed. The last section discusses the obtained results, identifying possible sources of differences between these models.

II. Large Language Models Evaluation Methods

Evaluating the performance of Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) systems is essential for ensuring their reliability, robustness, and alignment with real-world applications [3], [4]. Typically for LLM evaluations metrics such as BLEU, ROUGE, METEOR, and LEPOR are used [5]. They evaluate the n-gram overlap between generated and reference texts. These metrics are primarily used in tasks like machine translation and summarization. However, they may fall short in capturing semantic correctness and contextual relevance. In terms of semantic evaluation the common approach is based on embedding-based metrics leverage vector representations (e.g., Sentence-BERT) [6] to quantify the semantic similarity between outputs and reference texts. Additionally, LLM-as-a-Judge methodologies employ a language model to assess generation quality based on coherence, relevance, and factual correctness, offering a more nuanced and human-like evaluation [7]. Another approach for LLM's evaluatino is based on human evaluation. Despite being resource-intensive, human evaluation remains the gold standard. Annotators rate generated outputs based on fluency, factual accuracy, and relevance, although results may be subject to variability across raters. There are also several automated evaluation frameworks among which DeepEval [8] and ARES [9] have streamlined the model evaluation process. The DeepEval integrates various metrics and supports LLM-as-a-Judge assessments, while ARES reduces the dependency on human annotations by training lightweight evaluators for tasks such as context relevance and answer faithfulness.

In the context of RAG, the evaluation task becomes more complex due to the multi-component architecture of the system. A typical RAG pipeline includes a document corpus, an information retrieval module, and a large language model (LLM) that processes the retrieved document chunks to generate a final response. Each of these components can be evaluated individually, as well as in terms of their overall contribution to the system's performance.

The retrieval module is commonly assessed using several key metrics. Relevance measures how well the retrieved documents correspond to the user query. Comprehensiveness evaluates the diversity and coverage of retrieved content, ensuring that different aspects of the query are captured. Correctness refers to the accuracy of retrieved documents compared to all possible relevant candidates. Context Relevance assesses whether the retrieved context is sufficient to support a correct and complete response to the query. Fore more details see [10].

The LLM component can also be evaluated using distinct criteria. Faithfulness captures the degree to which the generated response accurately reflects the information found in the retrieved documents. Relevance, in this context, refers to the alignment of the generated response with the user's query intent. Key Point Recall measures how well the response incorporates essential information from the retrieved content. Response Completeness evaluates whether the answer fully addresses the user's query, while Response Conciseness assesses the amount of extraneous or irrelevant content present in the response. (see [11])

In this work, we focus only on the Faithfulness measure of the models supporting the Polish language.

III. The Modiffied "needle in a haystack" test

The classical formulation of the needle in a haystack test involves embedding a sentence containing a specific piece of factual information into a longer textual passage, followed by querying the model with a prompt that indirectly or directly references this content. While this method provides a foundational framework for assessing factual recall in language models, it exhibits several limitations. In particular, the artificial insertion of a standalone sentence often lacks semantic coherence with the surrounding context. This disjunction can disrupt the natural flow of the passage and interfere with the attention mechanisms of transformer-based architectures, potentially introducing artifacts that obscure the model's true retrieval capabilities. Consequently, such tests may not reliably reflect performance under realistic usage scenarios.

To overcome these shortcomings, the evaluation methodology was refined by embedding the target information in a more contextually integrated manner. Instead of inserting isolated factual statements, passages were selected or constructed to ensure that critical details were naturally embedded within a coherent narrative structure. Queries were then formulated to require comprehension, synthesis, and accurate retrieval of these embedded facts. In particular, we focused on numerical

facts that can be easily and precisely evaluated, allowing for direct measurement of the quality. In particular, we measured the relative error of the returned numerical value. This approach better reflects real-world RAG conditions, where relevant information is interwoven with broader context, and ensures a more robust and valid assessment of model behavior.

Beyond these refinements, the needle in a haystack framework was extended to investigate two additional factors critical to LLM evaluation: context length and needle position. The first factor concerns the hypothesis that shorter textual contexts simplify retrieval by reducing the model's search space, whereas longer contexts pose greater challenges due to increased sequence length and potential dilution of attention. The second factor addresses the position of the target information within the passage. Intuitively, content located at the beginning or end of a prompt may be more salient and thus more easily retrieved, whereas information embedded in the middle of the context may be less accessible. Evaluating model performance across these dimensions provides deeper insight into their retrieval fidelity and helps characterize their suitability for deployment in practical, high-recall RAG systems.

IV. The Experiment Setup

For the purpose of model evaluation, a test corpus was constructed using recently released reports from Poland's Central Statistical Office (GUS). The use of this data ensured that the language models under evaluation had not encountered the content during their pretraining, thereby minimizing the risk of data leakage and enabling a more rigorous assessment of generalization and retrieval capabilities. Notably, the information queried from the models consisted primarily of numerical values, which served as the "needles" in the evaluation framework. The use of numeric data allowed for an objective and precise evaluation of retrieval accuracy, as the model outputs could be directly compared against ground truth values.

A detailed list of the source documents used in the study, along with the corresponding needle information, is provided in Table I.

The query used in the evaluation is presented in Table II

The evaluation method for scoring the quality of the model is shown in Figure 1. It searches for a part of the text containing numbers and then calculates the relative error between the extracted value and the true value. If the answer doesn't contain numerical values, the returned error is 1.

An additional critical factor examined in this study was the influence of input text length on retrieval performance. The evaluation encompassed models with three different context window sizes: 4096 tokens, 8192 tokens, and one model supporting an extended context length of 128k tokens. Accordingly, two main families of experiments were conducted: one for models limited to a 4096-token context, and another for those capable of handling 8192-token contexts. To ensure consistency in the comparative analysis, the 128k-token model was evaluated using inputs constrained to 8192 tokens.

Table I: Documents, needles and the query used in the experiments. The documents were obtained from GUS website.

id	**Document title**	**Needle**	**Query**
1	Efektywność wykorzystania energii w latach 2012–2022	Według scenariusza rekomendowanego do 2050 r. ponad 66% budynków zostanie doprowadzonych do standardu pasywnego	Ile procent budynków zostanie doprowadzonych do standardu pasywnego do 2050 r.?
2	Efektywność wykorzystania energii w latach 2012–2022	W podziale na sektory wskaźnik ODEX brutto wykazywał poprawę efektywności energetycznej w przemyśle (o 57,2% w porównaniu do 2000 r.)	Jaką poprawę efektywności energetycznej w przemyśle wykazał wskaźnik ODEX brutto w porównaniu do 2000 r.?
3	Powierzchnia i ludność w przekroju terytorialnym w 2024 r.	Tereny wiejskie obejmujące gminy wiejskie i obszary wiejskie w gminach miejsko -wiejskich zajmują łącznie powierzchnię 29 012 600 ha, co stanowi 92, 42% obszaru Polski.	Jaki procent obszaru Polski zajmują łącznie tereny wiejskie?
4	Uczenie się osób dorosłych 2022	Na wsi udział uczniów szkół o profilu zawodowym wyniósł 23,8%	Jaki był udział szkół o profilu zawodowym na wsi?
5	Uczenie się osób dorosłych 2022	Znajomość więcej niż jednego języka obcego deklarowało 27,0% badanych.	Jaki procent badanych deklarowało znajomość więcej niż jednego języka obcego?
6	Wybrane wskaźniki przedsiębiorczości w latach 2018–2022	W 2022 r. podmioty małe (o liczbie pracujących od 10 do 49 osób) stanowiły 53% przedsiębiorstw szybkiego wzrostu oraz 79% szybkiego spadku	Jaki procent przedsiębiorstw szybkiego spadku stanowiły podmioty małe w 2022r.?
7	Wybrane wskaźniki przedsiębiorczości w latach 2018–2022	W 2022 r. wśród przedsiębiorstw stabilnych 64% stanowiły jednostki małe, a 29% podmioty średnie.	Jaki procent przedsiębiorstw stabilnych stanowiły podmioty średnie w 2022r.?
8	Powszechny Spis Rolny 2020 Charakterystyka gospodarstw domowych rolników na podstawie połączonych danych z PSR 2020 i NSP 2021	W okresie dziesięciolecia 2010–2020 wzrosła liczba gospodarstw najmniejszych o powierzchni do 1 ha UR włącznie (o 1,6%)	O jaki procent wzrosła liczba gospodarstw o powierzchni do 1 ha UR w okresie dziesięciolecia 2010- 2020?
9	Powszechny Spis Rolny 2020 Charakterystyka gospodarstw domowych rolników na podstawie połączonych danych z PSR 2020 i NSP 2021	Blisko połowa (48,6%) ludności wiejskiej tworzącej gospodarstwa domowe z użytkownikiem posiada wykształcenie zasadnicze zawodowe lub podstawowe.	Jaki procent ludności wiejskiej tworzącej gospodarstwa domowe z użytkownikiem posiada wykształcenie zasadnicze zawodowe lub podstawowe?
10	Koniunktura w przetwórstwie przemysłowym, budownictwie, handlu i usługach 2000-2024	W porównaniu z lipcem ub.r. wzrosło znaczenie barier niedoboru wykwalifikowanych pracowników (z 17,1% do 21,9%)	Do ilu procent wzrosło znaczenie barier niedoboru wykwalifikowanych pracowników w porównaniu z lipcem ub. r.?

Table II: Prompt used for model evaluation. The *query* and *needle* are provided in Table I

```
Kontekst: {context including needle}
Pytanie: {query}
Jako odpowiedź możesz podać tylko liczbę.
Jeżeli nie znajdziesz wyniku napisz BRAK.
Odpowiedz po polsku.
```

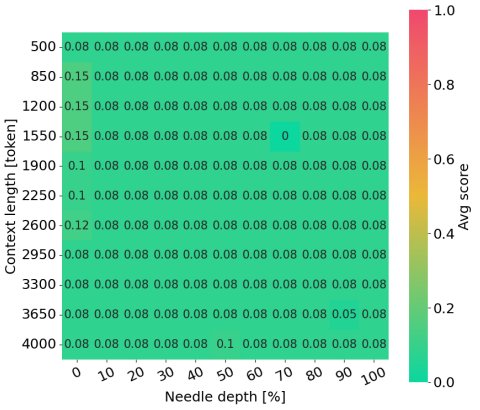
```python
def evaluate_response(response, true_answer):
    pattern = re.compile(r"[0-9]+,[0-9]+|[0-9]+\.[0-9]+|[0-9]+"re.IGNORECASE)
    match = pattern.findall(response.replace(" ", ""))
    if not match:
        return evaluate_response_body(response, true_answer)

    best = 1
    for m in match:
        score = evaluate_response_body(m, true_answer)
        if best > score:
            best = score
    return best
```

(a) Evaluate_response function

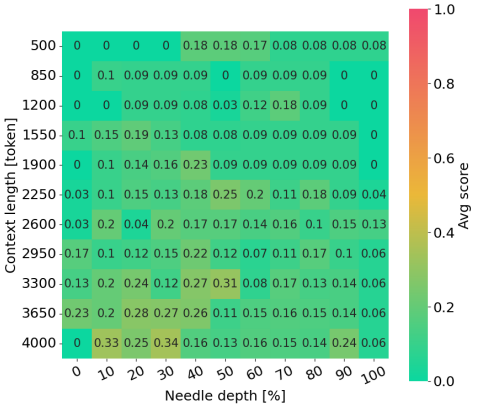
```python
def evaluate_response_body(response, true_answer):
    response = response.replace("%", "").replace(",", ".")
    try:
        response = float(response)
        return abs(true_answer - response) / true_answer
    except ValueError:
        return 1
```

(b) Evaluate_response_body function

Figure 1: Functions used for evaluation the response returned by the LLMs



(a) Bielik-11B-v2.3-Instruct



(b) Bielik-7B-v0.1

Figure 2: The results obtained for models with 4k context length. Part A. X-axis represent the depth of the needle, and Y-axis represent context length.

To systematically assess the effect of input size, each model was tested on 11 different text lengths. Additionally, the influence of the needle's position within the input was evaluated by placing the target information at 11 distinct locations throughout the text, including the beginning, middle, and end. This design enabled a comprehensive investigation into how retrieval performance varies with respect to both input length and the positional salience of the information.

The following models were selected for evaluation using a 4096-token context window:

- Bielik-11B-v2.3-Instruct,
- Bielik-7B-v0.1,
- trurl-2-13b,
- gpt-3.5-turbo-instruct,

and for the context length of 8196 tokens the following models were evaluated

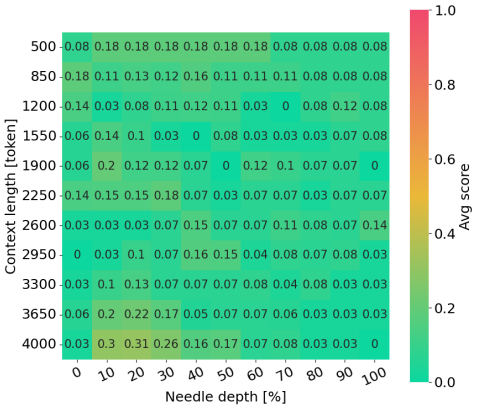
- NeuralDaredevil-8B-Abliterated,
- Llama-3-8B-Omnibus-1-PL-v01-INSTRUCT,
- Kruk-7B-SP-001,
- Starling-LM-7B-alpha,
- OpenChat-3.5-0106-Gemma,
- PLLuM-12B-instruct.

V. Results and Discussion

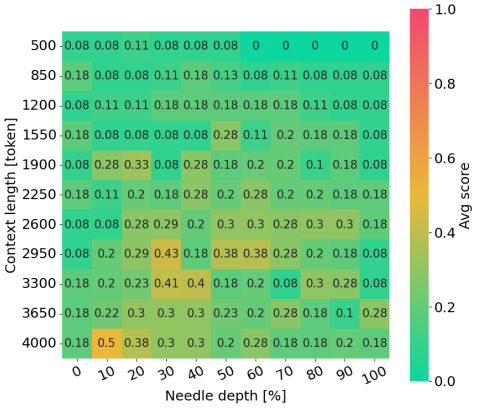
As described in the previous section, the experiments were conducted separately for models with two different maximum context lengths. The results are presented using heatmap visualizations on a two-dimensional grid, where the X-axis represents the relative depth (position) of the needle within the input text, and the Y-axis denotes the length of the input context. Each cell in the heatmap reflects the probability of an error, with values ranging from 0 (indicating no errors across all test samples) to 1 (indicating that the model consistently failed to retrieve the correct information). Accordingly, green indicates perfect accuracy, while red denotes complete failure in retrieval. To facilitate consistent visual interpretation, the same colormap scale was used across all heatmaps, allowing for direct comparison of prediction performance across different models and experimental conditions.

A. 4k Context Length

- **Bielik-11B-v2.3-Instruct** The results are presented in Figure 2a. The findings indicate that the model exhibits



(a) Trurl-2-13b



(b) GPT3.5-turbo-instruct

Figure 3: The results obtained for models with 4k context length. Part B. X-axis represent the depth of the needle, and Y-axis represent context length.

a consistent error probability across the evaluated range, suggesting that the likelihood of an error is not significantly influenced by either the needle depth or the context length. In all cases, the observed errors corresponded to a single test case involving the needle *W porównaniu z lipcem ub.r. wzrosło znaczenie barier niedoboru wykwalifikowanych pracowników (z 17,1% do 21,9%)* and the query *Do ilu procent wzrosło znaczenie barier niedoboru wykwalifikowanych pracowników w porównaniu z lipcem ub. r.?*, where the model consistently returned the first numerical value instead of the second one, despite the query clearly referencing the latter.

- **Bielik-7B-v0.1** The results are shown in Figure 2b. These indicate that the model's performance is highly dependent on the position of the needle. Specifically, the model is particularly sensitive when the needle is located early in the context—around 20% of the total context length. Additionally, its performance deteriorates with longer context lengths. For the longest context length, the model failed in approximately one-third of the cases when the needle was situated at about 20% depth.

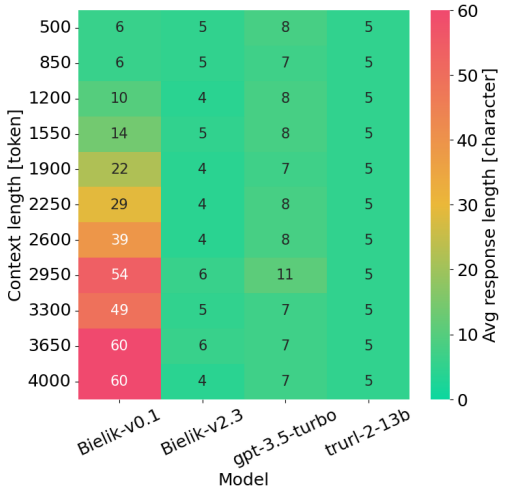


Figure 4: Relation between context length and the length of returned output in tokens for 4k models.

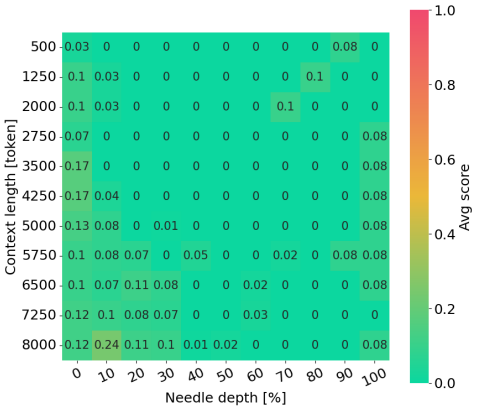
- **Trurl-2-13B** The corresponding results are presented in Figure 3a. The model exhibits a performance pattern similar to Bielik-7B-v0.1, with the most significant errors occurring when the needle is positioned at around 20% of the context depth and when the context length approaches its maximum. However, unlike Bielik-7B-v0.1, Trurl-2-13B shows improved performance when the context length is reduced to 3300 tokens or fewer.
- **GPT-3.5-turbo-instruct** The results are shown in Figure 3b. Among all evaluated models with a 4k token context window, this model demonstrated the weakest performance. In the worst-case scenario, it exhibited an error rate of up to 50% when the needle was located near the beginning of the context (approximately 10%) and at the maximum context length. It also frequently produced incorrect answers for shorter context lengths around 2950 to 3300 tokens, where Trurl-2-13B performed comparatively well. GPT-3.5-turbo-instruct only achieved reliable performance when the context length was limited to 1550 tokens or less.

In summary, the best-performing model with a 4k context length was Bielik-v2, followed by Trurl, while GPT-3.5 surprisingly demonstrated the weakest performance.

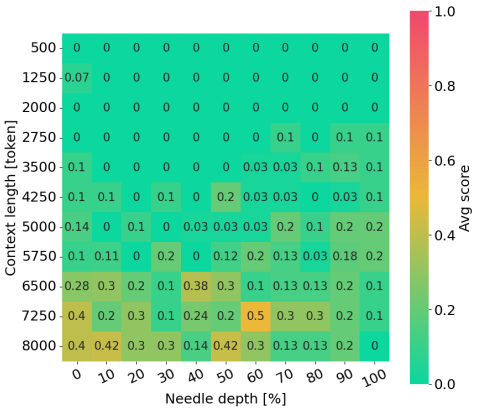
Additional insight is gained by analyzing the length of the output stream generated by the models, as shown in Figure 4. The results indicate that the Bielik-v1 model tends to produce significantly longer outputs, averaging up to 60 tokens when the context length approaches 4k. This is in stark contrast to the expected output, which typically consisted of only a few tokens representing a numerical value. This excessive verbosity contributed to a substantial increase in error rates.

B. 8k Context Length

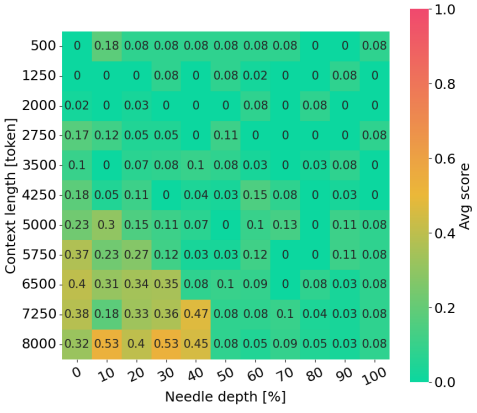
- **NeuralDaredevil-8B-Abliterated** The obtained results are shown in figure 5a. The model for a vary large part of



(a) [NeuralDaredevil-8B-Abliterated

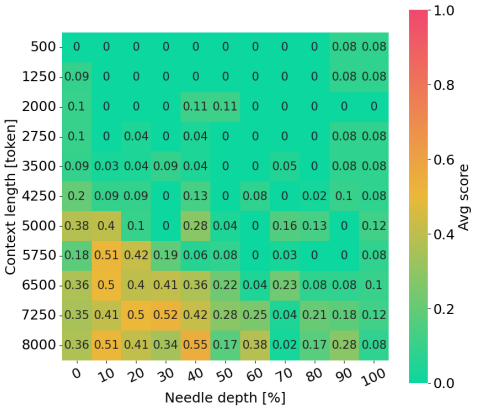


(b) Llama-3-8B-Omnibus-1-PL-v01-INSTRUCT

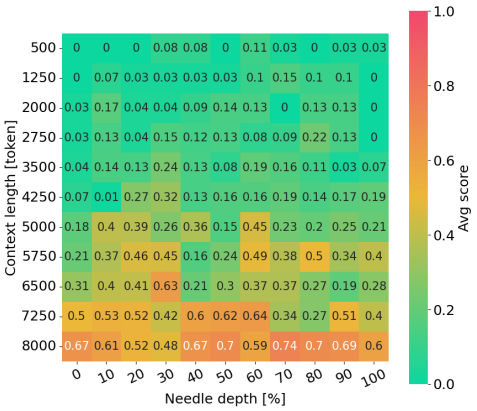


(c) Kruk-7B-SP-001

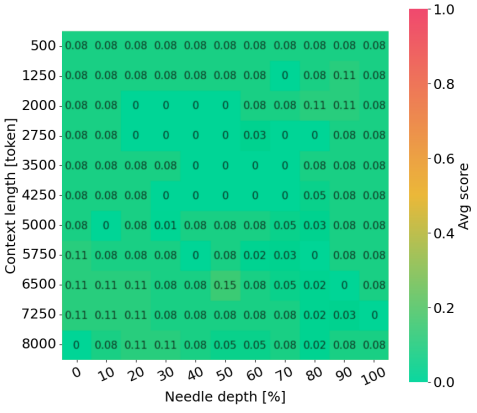
Figure 5: The results obtained for models with 8k context length. Part A. X-axis represent the depth of the needle, and Y-axis represent context length.



(a) Starling-LM-7B-alpha



(b) OpenChat-3.5-0106-Gemma



(c) PLLuM-12B-instruct

Figure 6: The results obtained for models with 8k context length. Part B. X-axis represent the depth of the needle, and Y-axis represent context length.

the experiment space didn't achieve any error. The errors appeared when the needle was located at the beginning of the context as the context length was growing. But the error rate was even though relatively low.

- **Llama-3-8B-Omnibus-1-PL-v01-INSTRUCT** The obtained results are shown in figure 5b. It achieves perfect predictions for short context length up to 4k, but when the context length starts to grow it starts to fail. The worst results are obtained for very long context at its begining.
- **Kruk-7B-SP-001** The obtained results are shown in figure 5c. It has similar behavior to all other models. Its error rate starts to grow when context is getting long and the needle is located at the begining. When compared to the omnibus model it can be observed a significant higher error rates for context length below 4k.
- **Starling-LM-7B-alpha** The obtained results are shown in figure 6a. Starling behaves similarly to Kruk except it has better performance for shorter context. Athough, it is warse for larger context, and achives slighly higher error rates.
- **OpenChat-3.5-0106-Gemma** The obtained results are shown in figure 6b. It is the worst of evaluated 8k models. It got larger error rates for the short context, and significantly worse results for the long context, where the error rates reaches 70% when the context length is close to 8k. But when compared to the 4k models it achieves similar performance reaching 0.32 error rates for context close to 4k.
- **PLLuM-12B-instruct** The obtained results are shown in figure 6c. Again it has very good performance for short context, and similarly when the context grow it achieves larger error rates but at most reaching 34%. This allows to get the second place behind NeuralDerdevil model.

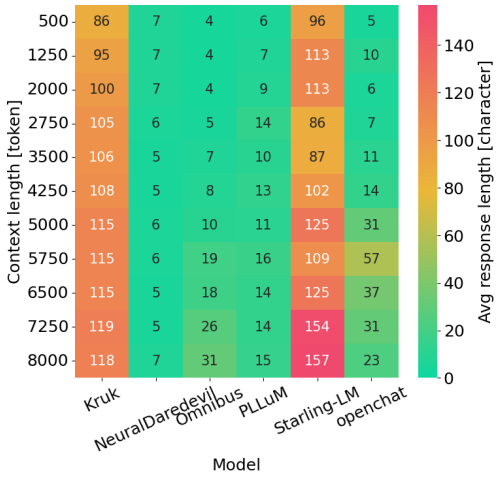


Figure 7: Relation between context length and the length of returned output in tokens for 8k models.

- **NeuralDaredevil-8B-Abliterated**
  The results are presented in Figure 5a. Across a large portion of the experimental space, the model achieved near-perfect performance. Errors occurred primarily when the needle was positioned at the beginning of the context as the overall context length increased. However, even in these cases, the error rate remained relatively low.
- **Llama-3-8B-Omnibus-1-PL-v01-INSTRUCT**
  As shown in Figure 5b, this model achieved perfect predictions for context lengths up to 4k tokens. However, performance degraded as the context length increased, with the most significant errors observed when the needle was located near the beginning of long contexts.
- **Kruk-7B-SP-001**
  The results, depicted in Figure 5c, show a pattern similar to other models. Error rates increase with longer contexts, particularly when the needle is located near the beginning. Compared to the Omnibus model, Kruk exhibits significantly higher error rates for contexts shorter than 4k tokens.
- **Starling-LM-7B-alpha**
  Figure 6a illustrates that Starling's behavior is comparable to Kruk's. It performs better for shorter contexts but shows slightly worse performance as context length increases, with marginally higher error rates overall.
- **OpenChat-3.5-0106-Gemma**
  The evaluation results, shown in Figure 6b, indicate that this model is the weakest among the evaluated 8k-context models. It exhibits higher error rates even for short contexts and significantly poorer performance for long contexts—reaching up to 70% error when the context is close to 8k tokens. However, when limited to 4k contexts, its performance aligns with that of other 4k-context models, reaching an error rate of approximately 32%.
- **PLLuM-12B-instruct**
  The results in Figure 6c show that the model performs very well for short contexts. As with other models, error rates increase with longer contexts but remain relatively low, peaking at around 34%. This strong performance places it second only to the NeuralDaredevil model.

In summary the best performing model among 8k models was NeuralDaredevil followed by PLLuM, and the worst one is OpenChat. Similarly to the 4k models, worth deeper investigation is the output text token length. Such relation is shown in figure 7. It shows that Starling and Kruk has significant longer output length. These models insted of returning precise output value that was queried, returned full sentence, often setence containing the output text. Similarly Omnibus and OpeChat when couldn't find the answer in the text returned full sentence insted of simple and short answer.

C. Summary and Discussion

In summary, among the evaluated models, NeuralDaredevil-8B-Abliterated achieved the best performance, followed by PLLuM and Bielik. The weakest performance was observed for OpenChat - see Figure 8.

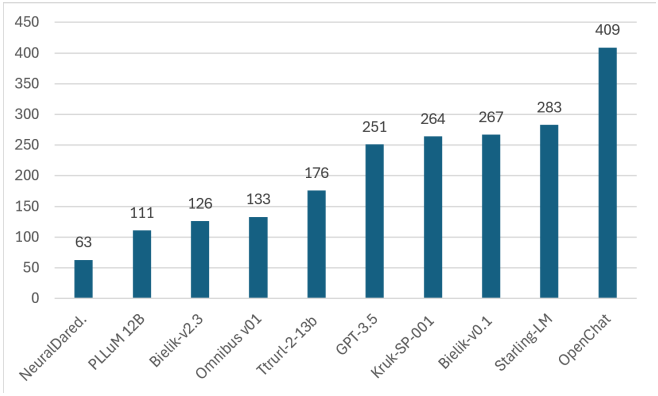


Figure 8: Number of factual errors made by each evaluated model.

It is particularly notable that the NeuralDaredevil-8B-Abliterated model significantly outperformed all other models, including PLLuM and Bielik-v2.3. The error rates for PLLuM and Bielik-v2 were nearly twice as high as those observed for NeuralDaredevil. A closer inspection revealed that the majority of errors across models were linked to a single query: *W porównaniu z lipcem ub.r. wzrosło znaczenie barier niedoboru wykwalifikowanych pracowników (z 17.1% do 21.9%)* and the corresponding question *Do ilu procent wzrosło znaczenie barier niedoboru wykwalifikowanych pracowników w porównaniu z lipcem ub. r.?* In this case, many models incorrectly returned the initial value (17.1%) rather than the correct final value (21.9%). This suggests difficulties in interpreting comparative constructions in Polish, particularly the meaning of prepositions such as "z" (from) and "do" (to).

One possible explanation for NeuralDaredevil's strong performance lies in the abliteration process, which may enhance the model's generalization capabilities. Previous studies have shown that excessive safety alignment or over-optimization can negatively impact a model's reasoning and factual recall abilities [12], [13], [14]. NeuralDaredevil-8B-Abliterated is a fine-tuned variant of the Daredevil-8B model based on the LLaMA-3 architecture. Fine-tuning was conducted with a single pass over the `mlabonne/orpo-dpo-mix-40k` dataset using the Direct Preference Optimization (DPO) method [15]. The abliteration technique, as described by Arditi et al. [16], involves removing the activation direction associated with refusal behaviors in the transformer's residual stream. This modification may allow the model to respond more freely to prompts that might otherwise trigger refusal responses, thereby improving reasoning in neutral tasks such as factual retrieval.

Furthermore, a consistent pattern across all evaluated models was observed: retrieval performance degraded as the input text approached the maximum context length and the target (needle) appeared near the beginning of the document. This is visually evident in all the figures (2,3,5 and 6), where the bottom-left corners (representing early-position needles in long contexts) are more yellow, indicating higher error rates. In contrast, queries located in the first half of the context window often resulted in near-perfect accuracy, as seen with models such as NeuralDaredevil-8B-Abliterated, Llama-3-8B-Omnibus-1-PL-v01-INSTRUCT, and also in Starling-LM-7B-alpha. This suggests that attention limitations in transformer-based models still substantially impact retrieval success in long-context scenarios.

Interestingly, some models—namely Bielik-11B-v2.3-Instruct and PLLuM-12B-Instruct—exhibited relatively stable error rates across the entire context window. This may be a result of their fine-tuning strategies, which could help mitigate the performance degradation typically caused by long input sequences.

Summarizing the length of the returned output shown in Figure 7 and Figure 4 some models tends to return significantly longer responses. In particular Starling and Kruk consistently produced significantly longer output sequences. Instead of returning a concise value in response to the query, these models often generated full sentences that included or paraphrased the expected output. Similarly, Omnibus and OpenChat, when unable to locate the exact answer in the context, tended to return verbose responses rather than the brief, precise values requested. This behavior may contribute to higher error rates and reduced response faithfulness in scenarios requiring exact factual retrieval.

VI. Conclusions

The primary goal of this research was to evaluate the ability of language models to accurately extract factual information from Polish input texts, rather than focusing on the linguistic quality of their output. This problem was formulated to provide a rough estimate of the models' susceptibility to hallucinations when used in Retrieval-Augmented Generation (RAG) systems, particularly in the context of less commonly supported languages such as Polish. Among the evaluated models, NeuralDaredevil-8B-Abliterated clearly outperformed both PLLuM and Bielik-v2. Surprisingly, the commercial GPT-3.5 model performed poorly, exhibiting twice as many errors as PLLuM or Bielik, and four times as many as NeuralDaredevil.

Additionally, a recurring type of error was observed across models: the misidentification of numerical values in comparative statements. Specifically, models often selected the initial value instead of the final one in scenarios describing change over time. This indicates a difficulty in precisely understanding certain linguistic constructs, that is a challenge in correctly interpreting comparative expressions involving "from" (Polish: *od*) and "to" (Polish: *do*).

Finally, our results revealed that all models experienced a decline in retrieval accuracy when the input text approached the maximum context window and the relevant information (needle) was located near the beginning of the document. This highlights a continuing limitation of transformer-based architectures in processing long documents, due to the reduced effectiveness of attention mechanisms over extended contexts. These findings underscore that, despite recent advancements,

long-range dependency handling remains a significant challenge for contemporary language models.

ACKNOWLEDGMENT

This research was supported by the Silesian University of Technology, grants No. BK-227/RM4/2025

REFERENCES

[1] Z. Li, X. Xu, T. Shen, C. Xu, J.-C. Gu, Y. Lai, C. Tao, and S. Ma, "Leveraging large language models for NLG evaluation: Advances and challenges," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024. doi: 10.18653/v1/2024.emnlp-main.896 pp. 16 028–16 045. [Online]. Available: https://aclanthology.org/2024.emnlp-main.896/

[2] Y. Kuratov, A. Bulatov, P. Anokhin, I. Rodkin, D. Sorokin, A. Sorokin, and M. Burtsev, "Babilong: Testing the limits of llms with long context reasoning-in-a-haystack," *Advances in Neural Information Processing Systems*, vol. 37, pp. 106 519–106 554, 2024.

[3] Z. Guo, R. Jin, C. Liu, Y. Huang, D. Shi, L. Yu, Y. Liu, J. Li, B. Xiong, D. Xiong *et al.*, "Evaluating large language models: A comprehensive survey," *arXiv preprint arXiv:2310.19736*, 2023.

[4] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM transactions on intelligent systems and technology*, vol. 15, no. 3, pp. 1–45, 2024.

[5] T. Hu and X.-H. Zhou, "Unveiling llm evaluation focused on metrics: Challenges and solutions," *arXiv preprint arXiv:2404.09135*, 2024.

[6] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[7] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu *et al.*, "A survey on llm-as-a-judge," *arXiv preprint arXiv:2411.15594*, 2024.

[8] "Deepeval," https://docs.confident-ai.com/, 2024.

[9] J. Saad-Falcon, O. Khattab, C. Potts, and M. Zaharia, "Ares: An automated evaluation framework for retrieval-augmented generation systems," *arXiv preprint arXiv:2311.09476*, 2023.

[10] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu, "Evaluation of retrieval-augmented generation: A survey," in *CCF Conference on Big Data*. Springer, 2024, pp. 102–120.

[11] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, vol. 2, no. 1, 2023.

[12] J.-C. Gu, H.-X. Xu, J.-Y. Ma, P. Lu, Z.-H. Ling, K.-W. Chang, and N. Peng, "Model editing harms general abilities of large language models: Regularization to the rescue," *arXiv preprint arXiv:2401.04700*, 2024.

[13] W. Yang, F. Sun, X. Ma, X. Liu, D. Yin, and X. Cheng, "The butterfly effect of model editing: Few edits can trigger large language models collapse," *arXiv preprint arXiv:2402.09656*, 2024.

[14] S. Sonkar, N. Liu, and R. G. Baraniuk, "Regressive side effects of training language models to mimic student misconceptions," *arXiv e-prints*, pp. arXiv–2404, 2024.

[15] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems*, vol. 36, pp. 53 728–53 741, 2023.

[16] A. Arditi, O. Obeso, A. Syed, D. Paleka, N. Panickssery, W. Gurnee, and N. Nanda, "Refusal in language models is mediated by a single direction," *arXiv preprint arXiv:2406.11717*, 2024.