Multilingual Needle in a Haystack: Investigating Long-Context Behavior of Multilingual Large Language Models

Anonymous ACL submission

Abstract

While recent large language models (LLMs) demonstrate remarkable abilities in responding to queries in diverse languages, their ability to handle long multilingual contexts is unexplored. As such, a systematic evaluation of the long-context capabilities of LLMs in multilingual settings is crucial specifically in the context of information retrieval. To address this gap, we introduce the *MultiLingual Needle-in-a-Haystack* (MLNeedle) test, designed to assess a model's ability to retrieve relevant information (the *needle*) from a collection of multilingual distractor texts (the *haystack*). This test serves as an extension of the multilingual question-answering task, encompassing both monolingual and cross-lingual retrieval. We evaluate four state-of-the-art LLMs on MLNeedle. Our findings reveal that model performance can vary significantly with language and needle position. Specifically, we observe that model performance is the lowest when the needle is (i) in a language outside the English language family, and (ii) located in the middle of the input context. Furthermore, although some models claim a context size of $8k$ tokens or greater, none demonstrate satisfactory cross-lingual retrieval performance as the context length increases. Our analysis provides key insights into the long-context behavior of LLMs in multilingual settings to guide future evaluation protocols. To our knowledge, this is the first study to investigate the multilingual long-context behavior of LLMs.

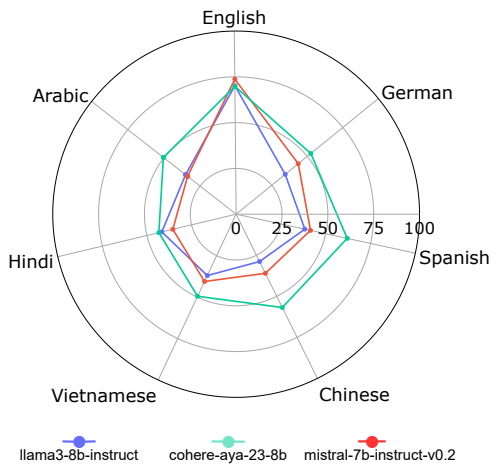

Figure 1: Monolingual long-context performance (accuracy in radial axis) for various LLMs averaged across different context sizes (4K, 8K, 16K, and 32K). We observe a considerable drop in performance for all languages except English, suggesting that multilingual LLMs struggle to process non-English (or non-Latin) long input contexts.

1 Introduction

In recent years, Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks, including text generation, translation, and question-answering. A critical aspect of these models is their ability to handle long input contexts effectively — a capability that is essential for applications such as document summarization, long-form content generation, and multi-turn dialogue systems (Petroni et al., 2020; Lee et al., 2022; Thoppilan et al., 2022). This ability directly impacts the relevance and accuracy of LLM responses over long inputs (Khandelwal et al., 2018; Mallen et al., 2023a; Shaham et al., 2023a; Kandpal et al., 2023a).

A recent study (Hsieh et al., 2024a) has shed light on the potential and limitations of LLMs in handling extended sequences. They evaluate various attention mechanisms and model architectures, revealing that while some models can technically manage long contexts, their effective utilization of this capacity is often suboptimal. Furthermore, Liu et al. (2023) highlighted a major problem faced by LLMs while handling long contexts: a marked decline in performance when the relevant information is located in the middle of a long input context. The performance

curve is characteristically "U-shaped", where models exhibit better accuracy when the relevant information is at the beginning or end of the context, underscoring the challenges LLMs face in maintaining attention and relevance throughout an entire input sequence. This phenomenon, dubbed the *lost-in-the-middle* problem, suggests that current LLMs are not yet fully equipped to handle long contexts robustly and reliably, particularly in tasks that require retrieval of dispersed information (Liu et al., 2023; Ivgi et al., 2023; Wang et al., 2024).

While these papers shed some light on the long-context capabilities of LLMs, a significant research gap remains: most existing benchmarks and evaluations have focused exclusively on monolingual English settings. This leaves a critical question unanswered: How do LLMs perform when the long input contexts are multilingual, particularly when the context is in a low-resource, non-Latin language? As shown in Figure 1, LLMs usually perform poorly while handling non-English long contexts. Furthermore, multilingual and cross-lingual contexts introduce additional complexities, such as varied syntax, grammar, and semantic nuances, which can significantly affect a model's retrieval and processing capabilities. In this paper, we address this gap by analyzing how LLMs process and retrieve information from long multilingual contexts. Specifically, we introduce the **MultiLingual Needle-in-a-Haystack (MLNeedle)** test, which extends the multilingual question-answering task to assess the LLM's ability to locate and extract relevant information (the *needle*) from a large collection of multilingual distractor texts (the *haystack*). Our experiments systematically vary the language and position of the needle within the haystack to evaluate the robustness of several state-of-the-art LLMs in handling multilingual long contexts. we make the following contributions[1]:

- We introduce the Multilingual Needle in a Haystack (MLNeedle) test, a first step towards systematically evaluating the long-context capabilities of multilingual LLMs. MLNeedle assesses model performance across seven languages in both monolingual and cross-lingual settings, providing a comprehensive benchmark for future research.
- We conduct a series of controlled experiments to examine how changes in the language and position of the needle affect model performance. Our findings reveal that LLM performance is highly sensitive to both the *language* and *position* of the needle in the haystack.
- We demonstrate the relative robustness of LLMs to variations in the language of distractor passages, indicating that the key challenges lie in how LLMs process and retrieve the *needle* from diverse linguistic environments.
- We perform several ablation studies to understand the role of temperature sampling, instruction tuning and the choice of evaluation metric on performance.

[1]The source code and datasets are attached as appendices and will be available publicly upon acceptance of the paper.

2 MultiLingual Needle in a Haystack

In this section, we introduce our MultiLingual Needle in a Haystack (MLNeedle) benchmark. As our goal is to better understand how LLMs process multilingual input contexts, we analyze the performance of the model on a multilingual question answering task, which requires a model to find relevant information (the *needle*) from the input context (the *haystack*) to answer the given question. Specifically, we conduct experiments where we systematically change (i) the position of the needle, (ii) the language of the needle, and (iii) the language of the haystack, and study the effect on performance. If LLMs are able to use information from long multilingual contexts, their performance should remain relatively stable regardless of changes in language or needle position.

2.1 Experimental Setup

In the multilingual question-answering task, the model is provided with a question Q to answer and K documents. Among these documents, exactly one contains the correct answer to the question Q, while the remaining $K-1$ *distractor* documents do not. We denote the document containing the correct answer as N (the *needle*) and the $K-1$ distractor documents as H (the *haystack*). Figure 2 offers an overview of our evaluation setup using a randomly sampled example from the MLNeedle dataset (details in Section 2.2). As illustrated in the figure, we systematically change the language of both the needle (highlighted in green) and the distractor documents (highlighted in red) during our experiments. The content of N and H remains

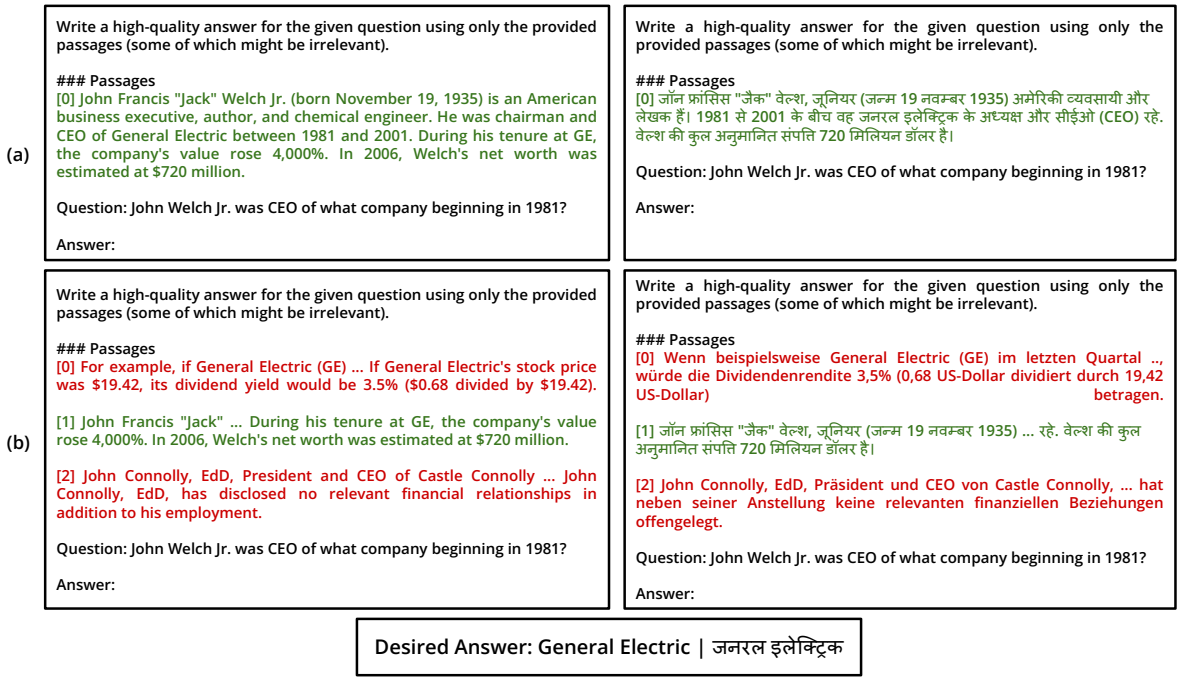

Figure 2: Example of a multilingual question-answering input from the MLNeedle dataset. (Top) There are no distractor documents and the same needle (highlighted in green) is present in English (Left) and Hindi (Right); (Bottom) There are distractor documents present (highlighted in red) and the same needle is present in English (Left) and Hindi (Right).

unchanged, only the language varies. Therefore, the LLM's performance should ideally not fluctuate due to these changes. We also vary the position of N within the input context, as highlighted in Figure 9 (Appendix), to understand the effect on the LLM's ability to retrieve. The language of the question is kept fixed in English.

2.2 The MLNeedle Dataset

We instantiate MLNeedle with the MLQA dataset (Lewis et al., 2020), which consists of over 5K extractive question-answer instances across seven languages (English, Arabic, German, Spanish, Hindi, Vietnamese, and Simplified Chinese) in the SQuAD (Rajpurkar et al., 2016) format. We choose MLQA because of its aligned dataset structure, where each question-answer pair is present in multiple languages. Specifically, for each question-answer instance, there are corresponding versions in at least four different languages, allowing for a direct comparison of how the same question is answered across various linguistic contexts. We highlight this in Figure 2 (a), where for the given question, the needle document with the correct answer can be presented in both English as well as Hindi. This setting allows us to systematically study the effect of changing the language of the needle N.

Constructing the Haystack (H). In MLNeedle, we collect the $K-1$ distractor documents for each question-answer pair using the following procedure: we use Wikipedia passages from mMARCO (Bonifacio et al., 2022), a well-known multilingual passage ranking dataset, as the source of the distractor documents. For each question-answer pair, we use multilingual sentence-BERT (Reimers and Gurevych, 2020) to retrieve $K-1$ documents from mMARCO that are most relevant to the question but do not contain the answer. Appendix A provides a detailed explanation of our retrieval system. In the final input context, we arrange these distractor documents in order of decreasing relevance. As highlighted in Table 1, we conduct experiments across different context sizes ranging from 4K up to 32K tokens. To modulate the context length, we simply increase or decrease the number of distractor documents in H. Further details on varying the context length can be found in Appendix C.

Positioning the Needle (N). We modulate the position of the relevant information within the input context by placing the document with the correct answer (N) at either the *start*, *middle*, or *end* of H (Figure 9 in Appendix), following prior experimental setups (Liu et al., 2023; Ivgi et al., 2023).

2.3 Models

We analyze several state-of-the-art open-source language models for our evaluation, spanning $32K$, $8K$ and $4K$ models. We report **Mistral-7B-Instruct-v0.2** (Jiang et al., 2023), which features a maximum context length of 32,768 tokens and is a multilingual model capable of understanding and generating text in multiple languages. It makes use of a unique positional encoding method, ALiBi (Press et al., 2022), to effectively manage long-range dependencies. Next, we evaluate **Cohere-Aya-23-8B** (Aryabumi et al., 2024), which also supports multilingual capabilities and has a context size of 8,192 tokens. This model is designed to perform well across various language tasks. We include **Llama3-8B-Instruct** (AI@Meta, 2024), an instruction fine-tuned version of the Llama3 base model (Dubey and Jauhri, 2024), which supports a context size of 8,192 tokens. This model is optimized for following instructions and engaging in open-ended dialogue. Lastly, we evaluate **Llama2-7B-Chat** (Touvron and Martin, 2023), which has a maximum context length of 4,096 tokens.

2.4 Evaluation Metric

As we evaluate the model's ability for retrieval in question-answering, we use **exact accuracy** (Kandpal et al., 2023b; Mallen et al., 2023b) as our primary evaluation metric. Additionally, in our ablation study (see Section 4), we report **existence accuracy** (Wang et al., 2024), which evaluates the secondary task of determining whether relevant information is present within the input context. Below, we formally define each of these metrics:

- **Exact Accuracy** measures the proportion of samples where the model's predicted output contains any of the correct answers, as specified in the MLQA dataset. This metric checks whether the ground-truth answer is contained in the model's predictions.
- **Existence Accuracy** is the proportion of samples where the model correctly identifies whether a document containing the correct answer exists within the input context.

We define the evaluation prompt templates for respective tasks in Appendix B. For consistency, we use the same evaluation prompts across all models in our experiments. Furthermore, for each input prompt, the model generates a prediction, which may be in a different language from the ground truth. To ensure accurate comparison, we translate the prediction to English (since each instance has a golden answer in English in MLQA) using the Google Translate API[2]. We then use the translated predictions and the ground truth to calculate Exact Accuracy as defined earlier. Further details of our evaluation framework are given in Appendix D.

3 Experimental Results

We run experiments with context sizes ranging from 4K to 32K tokens to compare the performance of various models on MLNeedle[3]. Table 1 summarises each model's performance, averaged across the seven languages in MLNeedle. As the context size increases, all models show a significant drop in performance. To evaluate the maximum context size that each model can handle effectively, we define the *effective length* as the maximum context length at which the model's performance does not decrease by more than 25% from its baseline accuracy [4]. Our findings show that all models struggle with longer contexts, exhibiting significant drops in accuracy beyond their effective lengths. Figure 1 shows the monolingual (both needle and distractors in the same language) performance of different models on the MLNeedle test. Models consistently perform better in English than non-English languages. In the following sections, we will investigate the effect of modulating the position and language of the needle.

3.1 Effect of Changing the Needle Position

In this experiment, we evaluate how the position of the needle (N) within the input context affects retrieval performance. We systematically place the needle at the *start*, *middle*, and *end* of the input context to understand how its position influences the model's ability to accurately retrieve the correct information. As shown in Figure 4, the model performs best when the needle is placed at the beginning or end of the input context. These findings extend Liu et al. (2023) to multilingual settings. The preference for starting and ending positions to retrieve relevant information indicates a

[2]https://cloud.google.com/translate/docs/reference/api-overview - Google Translate API

[3]All generation experiments were conducted using fixed sampling parameters to control the randomness and diversity of the generated responses. Specifically, we set the temperature to 0.7 and used top-k sampling with $k = 50$. These parameters were kept constant across all selected models to ensure a fair comparison of their performance on the MLNeedle test.

[4]Baseline accuracy refers to the model's performance when there are no distractors in the input context (see Figure 2(a)).

Model	Claimed Length	Effective Length	Baseline	4K	8K	16K	32K	Avg.
Llama2-7B-Chat	4K	<4K	0.335	0.171	–	–	–	0.253
Llama3-8B-Instruct	8K	4K	0.622	0.479	0.295	–	–	0.465
Cohere-Aya-23-8B	8K	4K	0.700	0.460	0.449	–	–	0.536
Mistral-7B-Instruct-v0.2	32K	8K	0.579	0.485	0.455	0.427	0.397	0.469

Table 1: Long-context performance of selected models on the MLNeedle test. Models are evaluated for context lengths ranging from 4K to 32K. Each score is determined by averaging the accuracy of MLNeedle's multilingual question-answering task. The ***effective length*** is the maximum context length beyond which the model's performance decreases by more than 25% from its baseline performance. The accuracy values within 25% of baseline performance are underlined, showcasing the effective length. For all models, we observe that the claimed context size differs from the effective context size.

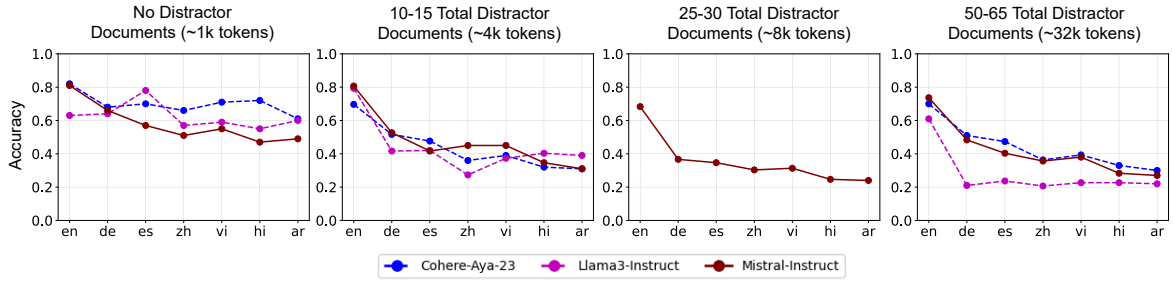

Figure 3: **Effect of changing the language of answer document (needle)**. We observe a similar trend across two differently pretrained families of models: Cohere-Aya-23 trained on highly parallel training data and Mistal-Instruct / Llama-3-Instruct trained on comparatively skewed training data.

potential weakness in the model's ability to maintain effective attention throughout the input sequence (Hsieh et al., 2024b), and exploring this in the multilingual setting is an important future work.

3.2 Effect of Changing the Needle Language

In this experiment, we investigate how changing the language of the document containing the correct answer (N) affects the retrieval performance of selected models. To isolate the effects of needle language, we keep the distractor passages (H) in English. The results, illustrated in Figure 3, show that LLMs perform best when N is either in English or in a language that is *close* to English. However, as we move away from the Latin languages family, we notice a significant drop in performance. The decrease in performance is most noticeable when N is presented in languages significantly different from English, such as Chinese and Arabic. When the language of N is changed from English to German or Spanish, both of which are *close* to English, the performance drop is moderate. This suggests that the model is relatively effective in processing content in languages that share similarities with English. On the other hand, the drop in performance is more pronounced when we change the language of N from English to non-Latin languages such as Chinese and Arabic. The substantial drop in performance indicates that the models struggle to effectively process and retrieve the same content when presented in these linguistically distant languages.

Our findings suggest that although the content of N remains unchanged, LLMs display considerable variability in their ability to retrieve the correct information depending on the language of N. This inconsistency underscores a critical limitation of current LLMs: *The retrieval capability is heavily influenced by the language in which the content is presented.* The models performs better for high-resource languages like English, but their performance diminishes in lower-resource languages such as Hindi.

3.3 Effect of Changing the Haystack Language

Here, we investigate how varying the language of the distractor documents, H, impacts the retrieval performance of LLMs. We keep the language of the needle constant, as English, and then systematically change the language of the haystack. Table 2 shows that changing the haystack language from English to Arabic does not significantly affect the model's performance. This observation suggests that LLMs are relatively robust to changes in the language of non-relevant information, indicating that the model can focus on and retrieve the relevant content

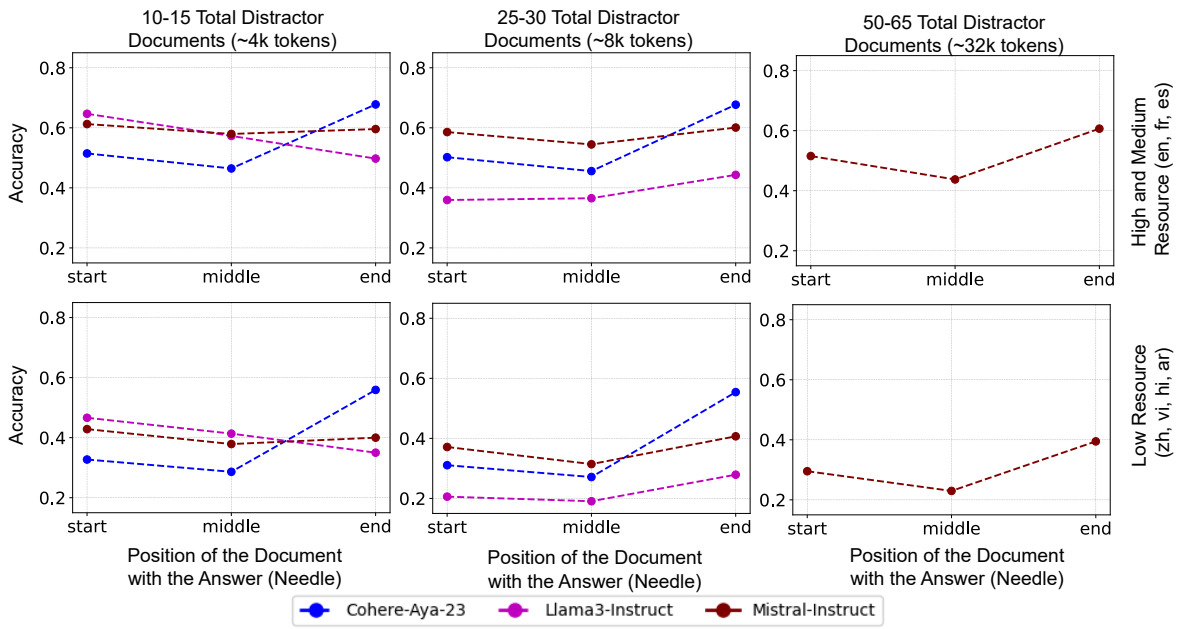

Figure 4: **Effect of changing position of answer document (needle)**. For all models at various context lengths, retrieving relevant information situated in the middle of the context is hard.

without being easily confused by the language of distractor passages. In contrast to the significant decline in performance observed when changing the language of the needle, the models' performance appears comparatively more stable when only the language of the distractor documents is altered. This suggests that the retrieval task's difficulty is more sensitive to the language of the needle than to the language of the haystack. LLMs focus on the content of the needle, rather than being distracted by the language of the haystack, highlighting its ability to prioritize relevant information effectively in this cross-lingual setting.

4 Ablation Studies

Effect of Temperature Sampling. Here, we investigate whether the choice of generation strategy significantly influences the model's performance. Table 4 compares the performance of Mistral-7B-Instruct-v0.2 under two different generation strategies: sampling with fixed parameters (temperature = 0.7, top-k = 50) and greedy decoding. Both strategies yield comparable results with minimal deviation across different context sizes. Furthermore, as shown in Figure 6, the overall accuracy remains consistent regardless of the generation method.

Effect of Instruction Fine-tuning. To understand the impact of instruction fine-tuning on LLMs' use of multilingual long contexts, we compare the MLNeedle test performance of Mistral-7B-Instruct-v0.2 with its base variant (pre-instruction tuning) using the same experimental setup as in Section 2. Table 3 shows that Mistral-7B-Instruct-v0.2 consistently outperforms Mistral-7B-v0.1 across different context lengths. Instruction-tuning also reduces worst-case performance disparity from nearly 70% to 30%. These findings align with prior work showing that instruction-tuning enhances cross-lingual knowledge alignment and improves information retrieval across languages (Shaham et al., 2024; Gao et al., 2024).

Effect of the Task Format. The task of multilingual question answering through fact retrieval from a long context can be considered a challenging task for the choice of models under study. To prove the reliability of our results as being a property of long-context multilingual LMs, we also evaluate the models on a simpler secondary task — identifying the presence of relevant information within the input context. To this end, we defined the **existence accuracy** metric in Section 2.4, which measures the proportion of samples where the model correctly identifies whether the relevant information is present in the provided passages.

Figure 7 shows the results for Mistral-7B-Instruct-v0.2 across four different context lengths. We observe similar results for both Exact Accuracy and Existence Accuracy, showcasing that our findings are not solely because of the nature of the task.

HN	en	de	es	zh	vi	hi	ar
Mistral-7B-Instruct-v0.2							
en	0.68	0.37	0.35	0.30	0.31	0.25	0.24
de	0.71	0.37	0.38	0.24	0.34	0.24	0.25
es	0.70	0.40	0.39	0.31	0.34	0.26	0.28
zh	0.73	0.43	0.39	0.31	0.35	0.28	0.27
vi	0.75	0.47	0.42	0.33	0.37	0.30	0.27
hi	0.80	0.47	0.44	0.36	0.38	0.32	0.33
ar	0.76	0.49	0.43	0.31	0.40	0.33	0.30
Llama3-8B-Instruct							
en	0.61	0.21	0.24	0.21	0.23	0.23	0.22
de	0.64	0.25	0.25	0.15	0.20	0.22	0.19
es	0.66	0.28	0.28	0.19	0.25	0.24	0.24
zh	0.61	0.22	0.23	0.19	0.19	0.20	0.18
vi	0.65	0.29	0.28	0.18	0.30	0.24	0.20
hi	0.70	0.32	0.32	0.26	0.27	0.29	0.26
ar	0.65	0.27	0.29	0.22	0.25	0.25	0.25
Cohere-Aya-23-8B							
en	0.70	0.51	0.47	0.36	0.39	0.33	0.30
de	0.71	0.53	0.46	0.36	0.41	0.33	0.33
es	0.68	0.48	0.62	0.35	0.43	0.35	0.32
zh	0.72	0.47	0.44	0.57	0.39	0.38	0.37
vi	0.73	0.46	0.46	0.31	0.49	0.36	0.36
hi	0.61	0.42	0.44	0.34	0.39	0.35	0.31
ar	0.66	0.45	0.44	0.35	0.41	0.37	0.50

Table 2: Pairwise accuracy results of selected models (averaged across the context lengths) on the MLNeedle test. Language of relevant information and distractor documents is abbreviated as N and H, respectively. We observe that performance is heavily influenced by the language of relevant information (N), whereas the language of distractor documents (H) plays a limited role.

We also observe similar results when we vary the language of distractor passages in the input context. More details can be found in Appendix E.

Statistical Significance. Figure 5 presents the results of significance testing for selected models, computed over linearly increasing sample sizes from 100 to 16,800. We observe that accuracy stabilizes after approximately 2,500 samples. Furthermore, a significant reduction in standard error is observed as the sample size increases from 100 to 2,500. These results underscore that a sample size of 2,500 is sufficient to achieve reliable and consistent evaluation outcomes. Further details are provided in Appendix E.

Model	**Baseline**	**4K**	**8K**	**16K**	**32K**	**Avg.**
Mistral-Base	0.383	0.142	0.100	0.097	0.102	0.165
Mistral-Instruct	0.586	0.478	0.453	0.436	0.398	0.470
$\Delta_{\text{model}^\dagger-\text{model}^*}$	↑ 0.203	↑ 0.336	↑ 0.353	↑ 0.339	↑ 0.296	↑ 0.305

Table 3: Effect of instruction fine-tuning.

Model	**Baseline**	**4K**	**8K**	**16K**	**32K**	**Avg.**
Mistral-Instruct$_{(GD)}$	0.586	0.478	0.453	0.436	0.398	0.470
Mistral-Instruct$_{(TS)}$	0.580	0.485	0.455	0.427	0.398	0.469
$\Delta_{\text{model}^\dagger-\text{model}^*}$	↓ 0.006	↑ 0.007	↑ 0.002	↓ 0.009	0.000	↓ 0.001

Table 4: Effect of temperature sampling vs greedy decoding.

5 Related Work

Multilingual Question Answering and Information Retrieval. Multilingual question answering (QA) and information retrieval (IR) have become increasingly important as LLMs are deployed in diverse linguistic environments (Shaham et al., 2024). Historically, QA datasets and benchmarks have been monolingual, primarily focusing on English. However, efforts such as MLQA (Lewis et al., 2020) and XQuAD (Artetxe et al., 2019) have introduced datasets supporting multiple languages, facilitating cross-lingual evaluation of QA systems. Despite these advancements, much of the research in multilingual QA has focused on scenarios where the context is relatively short or where the question and context are in the same language (Artetxe et al., 2019; Lewis et al., 2020; Longpre et al., 2021). A significant gap persists between monolingual and multilingual QA performance, particularly when models encounter cross-lingual scenarios or low-resource languages (Loginova et al., 2021; Guo et al., 2023). The MLQA (Lewis et al., 2020) and MKQA (Longpre et al., 2021) datasets, for instance, provide a foundation for evaluating cross-lingual extractive QA, but do not address the complexities introduced by long contexts. Our work extends this line of inquiry by examining how LLMs perform in retrieving relevant information from long multilingual contexts, where input context may span multiple documents — an increasingly common scenario in real-world applications.

Long-context Language Models. The ability of LLMs to effectively handle long input contexts is a critical area of research, underpinning tasks such as document summarization, long-form text generation, and multi-hop question answering (Qin et al., 2023; Wang et al., 2024). Transformer-based models (Vaswani et al., 2017), traditionally limited by their quadratic complexity relative to sequence length, have spurred the development of various techniques to scale attention mechanisms and manage long contexts more efficiently (Dai et al., 2019; Dao et al., 2022). Recent innovations have extended context windows significantly, with models now capable of processing up to 100K tokens in some cases. Hsieh et al. (2024a) explored the practical usability of long-context LLMs and the effectiveness of different attention mechanisms. Despite the recent improvements, the practical utility of LLMs in long-context

scenarios is often constrained by issues related to attention decay, memory management, and the ability to accurately retrieve relevant information from within extended sequences (Hsieh et al., 2024a; Li et al., 2024; Qin et al., 2023). Liu et al. (2023) highlighted a significant challenge in this domain: a marked decline in LLM performance when relevant information is situated in the middle of a long context. This study revealed a "U-shaped" performance curve, where models performed best when relevant information was at the beginning or end of the context, with performance dropping significantly for information located centrally. These findings underscore ongoing challenges in designing LLMs that can robustly handle long contexts, particularly in maintaining attention (Hsieh et al., 2024b) and relevance (Hsieh et al., 2024a) across the entire sequence.

Long-context Benchmarks and Tasks. Benchmarking the ability of LLMs to handle long contexts is crucial for understanding their real-world applicability (Hsieh et al., 2024a). Previous benchmarks, such as ZeroSCROLLS (Shaham et al., 2023b) and LongBench (Bai et al., 2024), have provided insights into the potential of LLMs in processing extended input sequences. These benchmarks typically involve tasks such as long-document QA or multi-hop reasoning (Liu et al., 2023; Hsieh et al., 2024a). L-Eval (An et al., 2023) curates tests using realistic data, which is filtered manually to ensure quality. Infinite-Bench Zhang et al. (2024) includes tasks with length greater than 100K tokens. However, these benchmarks primarily focus on monolingual English contexts, leaving a significant gap in understanding how LLMs perform in multilingual settings with long input sequences.

Recent findings by Zhao et al. (2024) reveal that LLMs process multilingual content in three stages: first, they convert input into an English-centric representation, then process it during task solving, and finally, generate output in the original language. Our results extend this hypothesis by suggesting that models struggle with retrieving information when dealing with non-Latin languages (Figure 3).

Furthermore, a recent study (Hsieh et al., 2024b) highlights the intrinsic positional attention bias in Transformer-based architectures (Vaswani et al., 2017), where models disproportionately focus on tokens at the beginning and end of a sequence. This explains the U-shaped performance curve

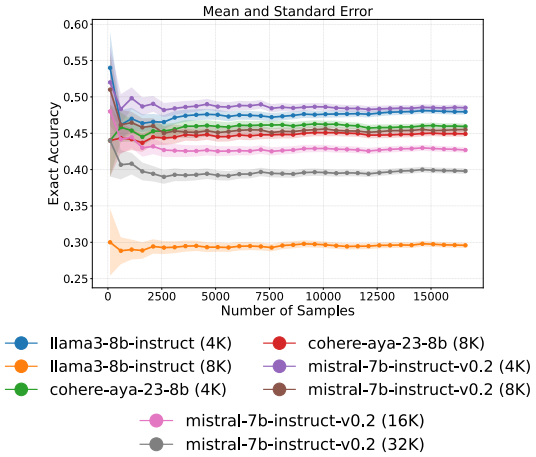

Figure 5: Exact accuracy of models on varying sample sizes for evaluation. Solid lines denote the accuracy, and the shaded area denotes the standard error.

observed in Figure 4. We further speculate that since the model's attention mechanisms are already biased due to positional encodings, the additional complexity of processing low-resource languages could exacerbate the difficulty in retrieving and utilizing middle-positioned information.

6 Conclusion

In this study, we introduced the MultiLingual Needle in a Haystack (MLNeedle) test, designed to systematically evaluate the long-context capabilities of multilingual LLMs. Through a series of controlled experiments, we investigated how changes in the language and position of relevant information in a long-context affects the LLMs' retrieval performance. Our findings revealed that LLMs exhibit significant sensitivity to both the language and position of the relevant information, particularly struggling when the relevant content is in non-Latin languages or positioned in the middle of a long context. Conversely, the models demonstrated relative robustness to variations in the language of distractor passages, suggesting that the primary challenges lie in how LLMs process and retrieve the *needle* from diverse linguistic contexts. These findings underscore the need for further research to enhance the multilingual capabilities of LLMs, particularly in handling scenarios where relevant information may be dispersed across different languages and positions within the input. Our work represents a first step towards systematically evaluating the long-context behavior of multilingual LLMs.

7 Limitation

The 'needle' documents used in our MLNeedle test are derived from the MLQA dataset - which is curated from Wikipedia. Thus, it is challenging to determine whether the models are utilizing the context or using their parametric knowledge. We plan to address this by expanding our MLNeedle test to include needles from other tasks like Ancestral Trace Challenge (ATC) or key-value retrieval. We select the distractor passages (haystack) purely based on semantic-similarity with the query. We do not train any specialized retriever model which might improve the quality of distractors. Moreover, due to resource and budget constraints, we were unable to evaluate LLMs greater than 8 billion parameters - like Command-R or LongAlapaca. Furthermore, we were also not able to assess API-based models such as GPT-4 and Claude. We keep this for future work.

8 Ethics Statement

The MLNeedle dataset is curated using the publicly available MLQA (Lewis et al., 2020) and mMARCO (Bonifacio et al., 2022) datasets. Our data curation strictly adheres to the usage policies of the original datasets, and we plan to release MLNeedle under the same open-source license. No human annotation or evaluation was conducted in this study. Statistical tests supporting our results are provided in both the main manuscript and the Appendix E.

References

AI@Meta. 2024. Llama 3 model card. *ArXiv*.

Chen An, Shansan Gong, Ming Zhong, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2023. L-eval: Instituting standardized evaluation for long context language models. *ArXiv*, abs/2307.11088.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan Gomez, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. mmarco: A multilingual version of the ms marco passage ranking dataset.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness.

Abhimanyu Dubey and Abhinav Jauhri. 2024. The llama 3 herd of models.

Changjiang Gao, Hongda Hu, Peng Hu, Jiajun Chen, Jixing Li, and Shujian Huang. 2024. Multilingual pretraining and instruction tuning improve cross-lingual knowledge alignment, but only shallowly. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6101–6117, Mexico City, Mexico. Association for Computational Linguistics.

Yiduo Guo, Yaobo Liang, Dongyan Zhao, Bing Liu, and Nan Duan. 2023. Analyzing and reducing the performance gap in cross-lingual transfer with fine-tuning slow and fast. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4002–4017, Toronto, Canada. Association for Computational Linguistics.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024a. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T. Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. 2024b. Found in the middle: Calibrating positional attention bias improves long context utilization.

Maor Ivgi, Uri Shaham, and Jonathan Berant. 2023. Efficient long-text understanding with short-text models. *Transactions of the Association for Computational Linguistics*, 11:284–299.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023a. Large language models struggle to learn long-tail knowledge.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023b. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 284–294, Melbourne, Australia. Association for Computational Linguistics.

Mina Lee, Percy Liang, and Qian Yang. 2022. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *CHI Conference on Human Factors in Computing Systems*, CHI '22. ACM.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Ekaterina Loginova, Stalin Varanasi, and Günter Neumann. 2021. Towards end-to-end multilingual question answering. 23(1):227–241.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. Mkqa: A linguistically diverse benchmark for multilingual open domain question answering.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023a. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation.

Guanghui Qin, Yukun Feng, and Benjamin Van Durme. 2023. The NLP task effectiveness of long-range transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3774–3790, Dubrovnik, Croatia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2024. Multilingual instruction tuning with just a pinch of multilinguality.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023a. Zeroscrolls: A zero-shot benchmark for long text understanding.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023b. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7977–7989, Singapore. Association for Computational Linguistics.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng,

Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications.

Hugo Touvron and Louis Martin. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Hengyi Wang, Haizhou Shi, Shiwei Tan, Weiyi Qin, Wenyuan Wang, Tunyu Zhang, Akshay Nambi, Tanuja Ganu, and Hao Wang. 2024. Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞bench: Extending long context evaluation beyond 100k tokens.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism?

A Process of retrieving the distractor documents

We randomly sample 10,000 Wikipedia passages from mMARCO for each of the languages under study. We encode every question in MLQA and every passage in mMARCO in a 768 dimensional dense embedding space using paraphrase-multilingual-mpnet-base-v2, trained on 50+ languages. Following that, we rank 300 most similar Wikipedia passages for every language against each question in MLQA using cosine similarity score.

B Prompt Templates

In our experiments, we adopt a vanilla prompt template commonly used in multi-document question answering, following the linear **`<Instruction> + <Documents> + <Query>`** input sequence. Table 5 and Table 6 highlight the respective prompt templates used for evaluating `exact_accuracy` and `existence_accuracy`. Additionally, Figure 2 and Figure 9 provide a example of actual input prompt for `exact_accuracy` evaluation using three documents.

C Controlling Context Length

To control the size of the input context within provided context size proposed by the model specifications, we allow for maximal number of distractor passages to fit within the size allowance. Using mMARCO passages as distractor passages we observed following number of documents to appear in the input context for each context size : 4K - ~10-15 distractor documents, 8K - ~25-30 distractor documents, 32K - ~50-65 distractor documents.

D Automated Evaluation Framework

This section outlines the key steps involved in our automated evaluation process. To better understand our evaluation setup, we provide an example of a positive and negative instance from MLNeedle in Figure 10.

For each input prompt, the model generates a prediction (`y_pred`), which may be in a different language from the ground truth (`y_true`). Each question-answer instance in MLNeedle dataset from MLQA has a golden answer provided in English, regardless of its language. To accurately

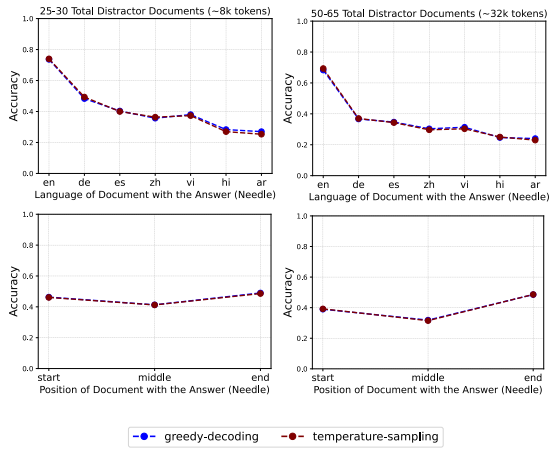

Figure 6: Performance of mistral-7b-instruct-v0.2 using different decoding strategies. We report similar trends for two different context lengths: ~8K tokens and ~32K tokens.

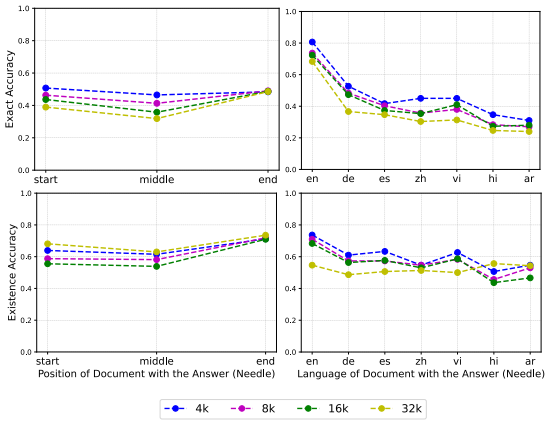

Figure 7: Exact Accuracy and Existence Accuracy plots for Mistral-7B-Instruct-v0.2 across four different context lengths on varying (from left to left) : (i) Position of Document with the Answer (Needle) and (ii) Language of Document with the Answer (Needle).

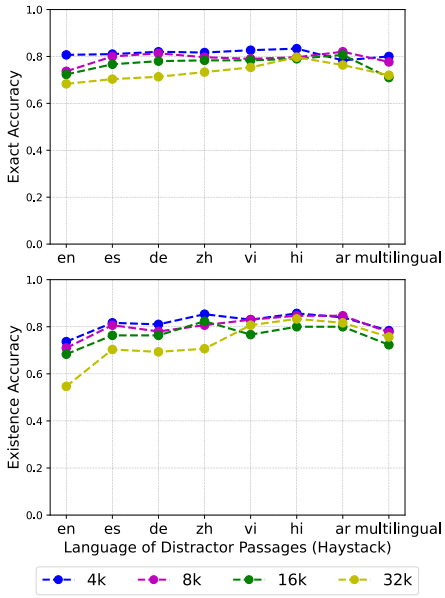

Figure 8: Exact Accuracy and Existence Accuracy plots for Mistral-7B-Instruct-v0.2 across four different context lengths on the varying language of distractor passages (Haystack).

compare `y_pred` with `y_true`, both must be in the same language. However, since `y_pred` can be generated in any language, we translate all model predictions into English using Google Translate.

As shown in Figure 10, the translation step ensures that the prediction is in the same language as the ground truth, allowing for a direct and reliable comparison. Once the predictions are translated, we proceed to compute the exact accuracy metric as defined in section 2.4. This comparison helps us determine if the model has successfully identified and retrieved the correct information from the input context.

By translating all model outputs to English before comparison, we minimize the risk of false negatives due to language differences. This process ensures that all comparisons are reliable and that the evaluation accurately reflects the model's ability to retrieve relevant information, regardless of the input language.

E Additional Results

E.1 Ablation Study: Effect of Evaluation Metric

In Section 4, we discussed how we obtained similar plots for 'Exact Accuracy' and 'Existence Accuracy'. From Figure 8, we observe the similarity between plots obtained on changing the language of the distractor documents in the input context. Both 'Existence Accuracy' and 'Exact Accuracy' do not vary in changing the language of the distractor documents in the input context.

E.2 Statistical Significance

To ensure the reliability of our evaluation, we conduct a hypothesis test for exact accuracy as defined in section 2.4. We conduct the test for selected models under a binomial distribution $\text{Binomial}(1, p)$, where p is the probability of success on an individual trial. The standard error (SE) of this test is computed as follows:

$$SE = \sqrt{\frac{p(1-p)}{s}}, \tag{1}$$

where s is the number of trials (evaluation samples). We vary the sample size linearly, starting from 100 samples and increasing by 500 samples at each step, i.e., 100, 600, 1100, 1600, and so on, until reaching the full sample size of MLNeedle (16, 800 samples). Note that at each step, we randomly select the samples from the MLNeedle dataset. Figure 5 highlights the results of our statistical tests. We observe that the exact accuracy stabilizes after approximately 2, 500 samples, and the standard error decreases significantly as the sample size increases from 100 to 16, 800.

Prompt Template for Exact Match	`Write a high-quality answer for the given question using only the provided passages (some of which might be irrelevant).`
	
	`### Passages`
	`{input_passages}`
	
	`Question: {question}`
	
	`Answer:`

Table 5: Prompt template used for evaluating `exact_accuracy`. The model is asked to generate an answer based solely on the provided passages.

Prompt Template for Existence Match	`Read the following list of passages and indicate whether any of the passages contain the right answer for the given question. Format your output strictly as 'Yes' or 'No'.`
	
	`### Passages`
	`{input_passages}`
	
	`Question: {question}`
	
	`Answer [Yes/No]:`

Table 6: Prompt template used for evaluating `existence_accuracy`. The model is asked to determine if the correct answer is present within the provided passages.

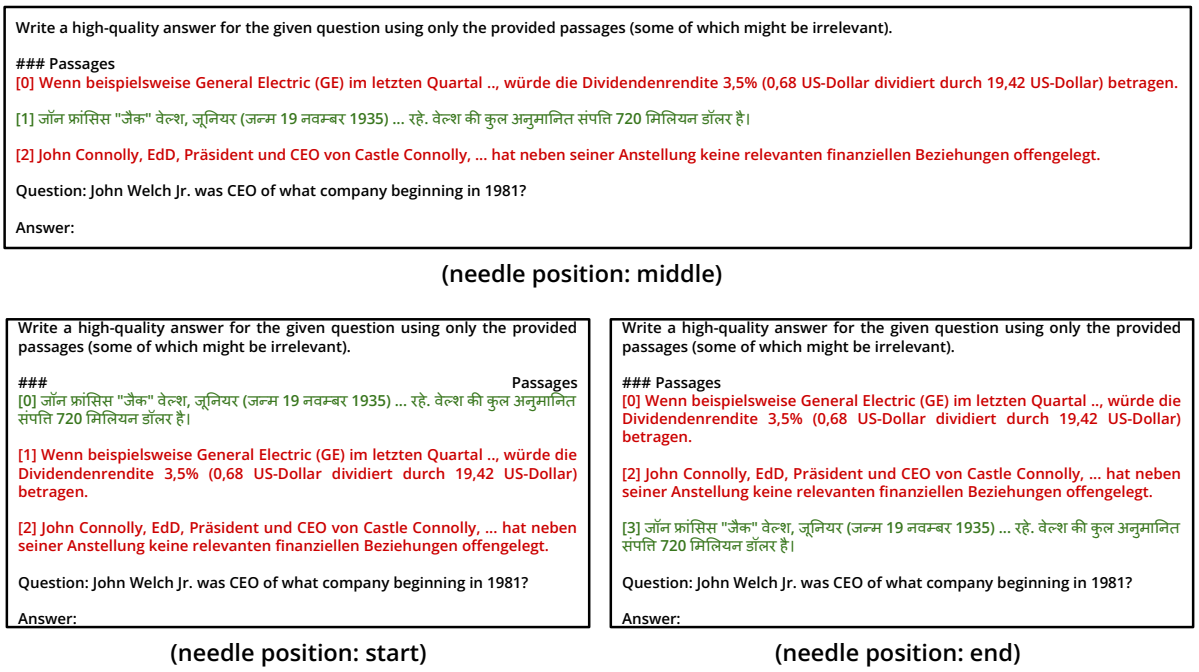

Figure 9: Modulating the position of relevant information ('needle') within the input context ('haystack') presented in Figure 2.

Ground-truth (y_true)	**Model prediction (y_pred)**	**Translation (y_pred_translated)**	**Exact Accuracy**

{
“जनरल इलेक्ट्रिक”,
“General Electric”
}
	"جنرال الكتريك“	“general electric”	1

{
“noroeste”,
“northeast”
}
	“đông về phía bắc”	“east to north”	0

Figure 10: An example of a positive and a negative instance while computing exact accuracy. The translation step enables a direct comparison between y_true and y_pred, which in turn helps reduce false-negatives.