Can LLMs reason over extended multilingual contexts? Towards long-context evaluation beyond retrieval from haystacks

Anonymous ACL submission

Abstract

Existing multilingual long-context benchmarks, often based on the popular needle-in-a-haystack test, primarily evaluate a model's ability to locate specific information buried within irrelevant texts. However, such a retrieval-centric approach is myopic and inherently limited, as successful recall alone does not indicate a model's capacity to reason over extended contexts. Moreover, these benchmarks are susceptible to data leakage, short-circuiting, and risk making the evaluation a priori identifiable. To address these limitations, we introduce `MLRBench`, a new synthetic benchmark for multilingual long-context reasoning. Unlike existing benchmarks, `MLRBench` goes beyond surface-level retrieval by including tasks that assess multi-hop inference, aggregation, and epistemic reasoning. Spanning seven languages, `MLRBench` is designed to be parallel, resistant to leakage, and scalable to arbitrary context lengths. Our extensive experiments with an open-weight large language model (LLM) reveal a pronounced gap between high- and low-resource languages, particularly for tasks requiring the model to aggregate multiple facts or predict the absence of information. We also find that, in multilingual settings, LLMs effectively utilize less than 30% of their claimed context length. Although off-the-shelf Retrieval Augmented Generation helps alleviate this to a certain extent, it does not solve the long-context problem. We open-source `MLRBench` to enable future research in improved evaluation and training of multilingual LLMs.[1]

1 Introduction

The landscape of large language models (LLMs) has evolved rapidly, especially in processing long-input sequences. Just two years ago, LLMs like GPT-3 and Llama supported a context window of no more than $4k$ tokens (Brown et al., 2020). By 2025, models like Gemini-2.0 can handle up to a million tokens (Reid et al., 2024), enabling inference over extremely long documents. This development has fueled their adoption in tasks such as dialogue systems, multi-document QA, long-document summarisation, and code debugging, which benefit from the ability to retain and reason over dispersed information (Lee et al., 2022; Rozière et al., 2024; Shah et al., 2024).

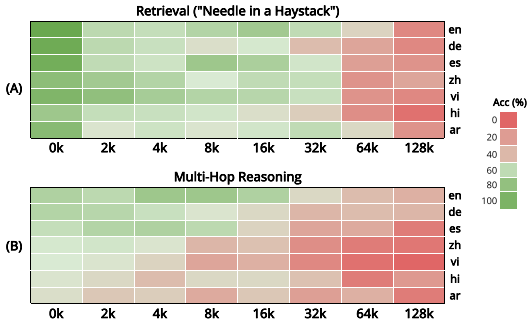


Figure 1: Question-answering performance of Llama-3.1-Instruct on two tasks: (a) simple retrieval/recall versus (b) multi-hop reasoning. As context length increases, the performance drop in (b) is sharper than that in (a), especially for non-Latin languages. This underscores the difficulty of task (b) over task (a). Note that the tasks are (a) Basic Factoid QA and (b) Fact Chaining, respectively.

Notwithstanding, some critical questions persist as we enter the long-context era of LLMs – Can LLMs fully utilize and reason over their claimed context window? Do they maintain comparable performance in multilingual settings, particularly for agglutinative, morphologically rich , and low-resource languages like Arabic or Hindi, relative to English? Confronting these questions is vital, as any attempts to increase LLM's context window are futile unless we achieve a conceptual and empirical understanding of their behavior in such settings.

Drawbacks of existing evaluation frameworks While long-context evaluation has received growing

[1]We plan to open-source the dataset, code, and experimental details after the anonymity period.

attention, it remains mostly focused on English (Bai et al., 2024; Liu et al., 2023; Hsieh et al., 2024a). Due to the complexity of realistic evaluations (Karpinska et al., 2024a), studies often use synthetic benchmarks as proxies (Hsieh et al., 2024a; Bai et al., 2024). A common approach is the *Needle-in-a-Haystack* (NIAH) test (Kamradt, 2023), which evaluates whether a model can retrieve specific facts from long contexts. Many benchmarks build on this by adding multiple needles (Li et al., 2024; Hsieh et al., 2024a) or incorporating QA tasks (Bohnet et al., 2024; Liu et al., 2023; Zhang et al., 2024). Multilingual benchmarks like MLNeedle (Hengle et al., 2024), mLongRR (Agrawal et al., 2024), and ONERULER (Kim et al., 2025) extend NIAH to other languages but still emphasize *retrieval* and *recall* (Vodrahalli et al., 2024).

However, recent work highlights limitations of such retrieval-centric evaluations (Goldman et al., 2024; Karpinska et al., 2024a). Retrieving facts does not imply the ability to synthesise information across contexts (Vodrahalli et al., 2024). LLMs may extract isolated facts from long inputs but struggle with logical connections, contradiction resolution, or coherent reasoning (Karpinska et al., 2024b). Figure 1 empirically illustrates this: performance on retrieval tasks remains stable as context grows, but reasoning degrades sharply, especially in non-Latin languages, suggesting that current benchmarks may overestimate long-context reasoning ability (Vodrahalli et al., 2024).

Beyond design, practical issues also limit existing multilingual benchmarks. Datasets like MLNeedle (Hengle et al., 2024) and mLongRR (Agrawal et al., 2024) risk data leakage, since they draw from open-source text. This raises concerns of *evaluation leakage* and *short-circuiting*[2], compromising evaluation integrity.

Contributions To address these issues, we introduce Multilingual Long-Context Reasoning (`MLRBench`), a synthetic benchmark evaluating LLMs beyond retrieval. `MLRBench` spans seven languages and includes tasks involving retrieval, multi-hop inference, aggregation, and epistemic reasoning. It scales to arbitrary context lengths and avoids evaluation leakage by using synthetic, reproducible generation processes. Unlike benchmarks focused solely on retrieval, `MLRBench` requires reasoning across facts, implicit relations, and temporal or spatial dependencies. Its task types are *minimal yet orthogonal*[3], offering comprehensive yet non-redundant evaluation. In summary, we make the following contributions:

- We propose `MLRBench`, a synthetic benchmark to evaluate the multilingual long-context behaviour of LLMs. Going beyond surface-level retrieval, `MLRBench` includes complex reasoning tasks in seven languages. For each language, `MLRBench` provides $1,000$ QA instances evaluated across eight context lengths, from baseline (no distractors) to $128k$ tokens, a total of $8,000$ evaluation prompts per language.
- Through extensive experiments, we study how changes in the language, task, and prompting method affect the LLM's ability to reason over long multilingual contexts. Our results reveal a significant performance gap between high-resource and low-resource languages, particularly for tasks involving multi-step reasoning or resolving uncertainty.
- We undertake exhaustive analysis of retrieval and reasoning tasks, finding that existing benchmarks may overstate LLMs' true long-context reasoning abilities.
- We conduct a detailed ablation study examining the impact of various hyperparameters, such as the type of noise and sampling strategy, on the LLM's ability to reason over long contexts.

2 Dataset

We curate `MLRBench` as a multilingual, long-context adaptation of the bAbI dataset (Weston et al., 2015). bAbI provides a suite of English QA tasks to evaluate core reasoning abilities such as temporal and spatial awareness, association, counting, induction, and so on. As shown in Figure 9, each bAbI data point consists of a passage, a question, and an answer. The passage is a sequence of independent statements (facts) – each simulating an interaction between a "character" and an "object" such as "*John took the football to the garden*" or "*Mary went to the kitchen*". The accompanying

[2]Evaluation leakage occurs when models have seen test data during pretraining. Short-circuiting refers to solving tasks via shortcuts like memorized knowledge, bypassing true reasoning (Lee et al., 2024).

[3]As defined by Vodrahalli et al. (2024), "minimal yet orthogonal" evaluation designs use a few complementary tasks to cover diverse performance aspects.

questions are designed to test different aspects of reasoning, e.g., spatial awareness ("*Where was the football before the garden?*") or counting ("*How many people visited the garden?*"). Therefore, the bAbI setup requires models to understand the interplay of different facts and to draw inferences based on the overall context as it unfolds throughout the passage.

2.1 Design Principles

We construct MLRBench based on five design principles as outlined below.

1) **Parallelism**: MLRBench is designed to be highly parallel; each task instance is available across all the selected languages with aligned QA pairs. Such a parallel data structure enables reliable cross-lingual comparisons as it decouples language-specific performance from sample-level difficulty (Lewis et al., 2020). This makes the evaluation results more interpretable and consistent across languages.

2) **Leakage and short-circuiting:** As discussed in Section 1, evaluation benchmarks often face risks of evaluation leakage and short-circuiting. To address this, we construct MLRBench using synthetically generated texts, which substantially lowers any chance of data leakage during pretraining. Additionally, all tasks in MLRBench have non-trivial difficulty, which prevents LLMs from relying on superficial patterns or memorised knowledge to answer correctly, thereby reducing the risk of short-circuiting.

3) **Minimal yet orthogonal tasks:** MLRBench is designed with a minimal yet orthogonal approach (Vodrahalli et al., 2024). It includes a set of core tasks, each targeting a distinct aspect of long-context reasoning, such as retrieval, multi-hop inference, aggregation, and uncertainty handling. This design has two key advantages: (i) it enables broad evaluation of long-context understanding without overlap, and (ii) it ensures modularity so that future versions of the dataset can easily incorporate new reasoning tasks.

4) **Domain-relevant distractors:** If background texts (i.e., distractors) are entirely out-of-distribution, they can make the evaluation *a priori identifiable* [4], and the task artificially easy. To avoid this, distractors in MLRBench are designed to be in-distribution, i.e., they closely resemble the structure and content of the relevant information rather than being obviously unrelated. This helps prevent the tasks from turning into a simple pattern-matching problem, where the correct passages or facts stand out too clearly. MLRBench offers fine-grained control over the type of distractors used, supporting three modes: (i) synthetically generated, (ii) sampled from external text corpora, and (iii) random noise.

5) **Scalability:** MLRBench is scalable to any arbitrary context length by simply varying the number and placement of distractor passages. Thus, MLRBench can be used to test any current or new open-weight of API-based LLMs without changing the underlying task complexity.

2.2 Languages

MLRBench covers seven typologically diverse languages: English (en), German (de), Spanish (es), Hindi (hi), Arabic (ar), Vietnamese (vi), and Simplified Chinese (zh). More details on language selection are provided in Appendix B.

2.3 Task Categories

MLRBench includes seven reasoning tasks from bAbI, grouped into four categories: retrieval, multi-hop inference, aggregation, and uncertainty. **Retrieval** tasks follow the needle-in-a-haystack framework (Kamradt, 2023), testing associative recall through basic factoid QA and yes/no questions, where answers lie in a single supporting sentence (Weston et al., 2015). **Multi-hop inference** tasks require combining multiple facts, such as chaining events over time or reasoning about inter-entity relationships, to reach a conclusion. **Aggregation** tasks focus on synthesis over chaining, including counting relevant entities and listing items or characters based on events, requiring the model to maintain state across the context (Vodrahalli et al., 2024). Finally, **Uncertainty** tasks evaluate epistemic reasoning, where the model must detect ambiguity or incomplete information and respond cautiously without overcommitting. Descriptions, formal definitions, and illustrative examples of all task categories are provided in Appendix C, with detailed task instances shown in Appendix Table 7.

[4] In long-context evaluation, *a priori identifiable* means that the relevant information (or facts) can be identified purely from surface-level cues, without requiring any deeper understanding of how that information relates to the final task (Vodrahalli et al., 2024).

Method	Prompt Type	Baseline	2*k*	4*k*	8*k*	16*k*	32*k*	64*k*	128*k*	Mean LC	Drop (%)	ECW
Prompting Baselines	Zeroshot	0.513	0.406	0.405	0.366	0.357	0.325	0.311	0.317	0.318	21.89	4*K*
	Translation	0.540	0.414	0.403	0.384	0.368	0.331	0.316	0.323	0.324	21.82	4*K*
	CoT	0.598	0.481	0.477	0.439	0.448	0.382	0.285	0.253	0.307	47.36	16*K*
	Fewshot	0.521	0.468	0.433	0.408	0.423	0.348	0.333	0.334	0.338	28.69	16*K*
RAG (JinaAI, *top-k*=100)	Zero-shot	-	0.429	0.397	0.381	0.340	0.334	0.330	0.300	0.321	30.11	8*K*
	Translation	-	0.435	0.390	0.388	0.364	0.368	0.349	0.342	0.353	21.42	8*K*
	CoT	-	0.473	0.455	0.443	0.446	0.438	0.412	0.400	0.417	15.35	32*K*
	Few-shot	-	0.473	0.455	0.443	0.440	0.431	0.403	0.393	0.409	16.86	32*K*
RAG (MpNet, *top-k*=100)	Zeroshot	-	0.424	0.405	0.388	0.378	0.366	0.339	0.337	0.347	20.62	8*K*
	Translation	-	0.435	0.412	0.387	0.401	0.393	0.357	0.349	0.366	19.71	8*K*
	CoT	-	0.473	0.462	0.449	0.445	0.439	0.416	0.414	0.423	12.33	32*K*
	Fewshot	-	0.473	0.462	0.449	0.449	0.432	0.409	0.401	0.414	15.11	32*K*

Table 1: We highlight the overall performance (averaged across all languages) of the selected model. "Mean LC" denotes the mean long-context performance (averaged across context windows $32k$, $64k$, and $128k$). "Drop %" denotes the percentage drop in accuracy when moving from context size $2k$ to $128k$ tokens. Effective context window (ECW) is the context size up to which accuracy remains within 30% of the best prompting baseline (0.598). Across results, we observe that the ECW is $32k$ while the model claims to support $128k$ tokens.

3 Experimental Setup

3.1 Task Overview

Our evaluation setup follows the widely adopted needle-in-a-haystack paradigm (Liu et al., 2023), where sentences from a relevant passage (IP) are dispersed within the set of distractor passages (D) while preserving their original chronological order. As shown in Figure 9, the input passage, derived from the bAbI dataset, contains a sequence of independent facts that must maintain their original order when embedded into the distractors. The distractors are sampled from synthetic, natural, or random-noise sources and do not overlap with the relevant content. Each final prompt includes a task-specific question (Q) and its corresponding ground-truth answer (A), requiring the model to reason over the embedded (IP) while filtering through irrelevant background text. This setup allows precise control over context length and complexity by varying the number of distractors. More details on the construction of long-context prompts and formal notations are provided in Appendix A.

3.2 Language Model

We conduct all our experiments using the open-weight Llama 3.1-Instruct[5] model, which is an instruction-finetuned version of Llama 3.1 (Meta AI, 2024).

3.3 Baselines

Prompting We experiment with four prompting strategies – *zeroshot* (ZS), *fewshot* (FS), *chain-of-thoughts* (CoT), and *in-context translation* (ICT). ICT is a type of cross-lingual prompting in which the model is instructed to interpret the context and respond in English. Translating non-English prompts into English before inference has been shown to improve performance on certain multilingual tasks (Mondshine et al., 2025; Qin et al., 2023). ICT is similar to the widely adopted practice of *pre-translation* (Ahuja et al., 2023), except that translation happens in-context, and thus, it avoids any additional overhead or information loss (Intrator et al., 2024; Nicholas and Bhatia, 2023).

Retrieval Augmented Generation (RAG) RAG provides a way to overcome the fixed-context windows of LLMs by dynamically retrieving and re-ranking relevant content from a larger context (Gao et al., 2024). In this work, we experiment with two off-the-shelf RAG retriever models: Jina-reranker (JinaAI) (Multilingual, 2024) and multilingual-mpnet-base-v2 (MPNet) (Reimers and Gurevych, 2020). Further details on the prompting strategies, prompt templates, and RAG pipeline are provided in Appendix D.

3.4 Evaluation Metrics

We use **exact accuracy** as our primary evaluation metric. Following prior work (Wang et al., 2024), exact accuracy is defined as the proportion of samples where the model's predicted output includes the ground-truth answer in any of the target languages in `MLRBench`. Similar to (Hengle et al., 2024), we apply normalisation (e.g., "1" and "one" or yes/no) during evaluation. Additionally, we evaluate RAG performance using **Recall@k**, which measures the *proportion of relevant sentences*

[5]We employ the **meta-llama/Llama-3.1-8B-Instruct** model checkpoints from Huggingface.

retrieved for a given question, calculated as the size of the intersection between the retrieved and relevant sets, divided by the total number of relevant sentences (Monigatti, 2025).

4 Experimental Results

We conduct inference experiments across a range of context sizes, beginning with the baseline setup (no distractors) and extending up to 128k tokens. Table 1 presents a summary of results for all baselines, averaged across the seven languages. Notably, even the best-performing method at the baseline, i.e., CoT, achieves only a moderate 60% success rate, suggesting that questions in `MLRBench` have non-trivial difficulty even in the absence of distractors. Across all methods, we observe a consistent decline in accuracy as context length increases, with the degradation becoming especially pronounced at longer contexts beyond 32k tokens. In the following section, we present detailed results and highlight the key trends.

Trend 1: Prompt-only methods struggle to maintain accuracy even at moderate contexts. Among prompt-only methods, CoT and FS outperform ZS and ICT consistently across all context lengths and languages, with an accuracy margin of at least 6% on average. However, prompt-based methods are also highly sensitive to an increase in context size. Their performance declines significantly, particularly beyond a context window of 16k tokens. For instance, CoT accuracy drops from 0.598 at baseline to 0.253 at 128k, reflecting a relative 57.6% performance loss. Prompt-based methods lose between 38% to 60% of their baseline accuracy at longer context windows ($\geq$ 32k), and nearly all prompting methods begin to deteriorate sharply after 8k to 16k tokens. This highlights the limited capacity of these methods to handle extended input contexts without additional retrieval mechanisms.

Trend 2: While RAG provides stability, it does not solve the long-context problem. RAG-based methods exhibit smaller accuracy drops between 2k and 128k tokens compared to prompt-only baselines. For instance, MpNet (CoT) shows a 40.41% decline, while CoT alone drops by 95.53%. RAG also leads to flatter degradation curves, indicating more robust and consistent performance as context grows. Without RAG, accuracy degrades rapidly after 4k–16k tokens across most prompting strategies. In contrast, combining RAG with CoT or FS yields more stable performance over increasing context lengths. Among RAG methods, MpNet + CoT performs best, with a mean accuracy of 0.423 across long contexts ($\geq$ 32k), the highest among all methods. Furthermore, we observe that RAG-MpNet outperforms RAG-JinaAI at both short and long contexts (Appendix Table 10).

Figure 6 compares RAG-MpNet with the top-performing prompt-based method per language. CoT leads at short context lengths, while RAG-MpNet surpasses FS at longer ones. Despite improved stability, RAG does not fully eliminate the effects of long-context scaling. Even under optimal settings, RAG accuracy at long context windows remains 20%–40% below the best short-context baseline. This suggests that while RAG extends usable context, significant progress is still needed to handle extreme input lengths effectively.

Trend 3: Multilingual performance drops sharply with increasing linguistic distance from English. Unsurprisingly, English achieves the highest performance across both short and

Language	*top-k*	en	de	es	zh	vi	hi	ar	Avg.
RAG-JinaAI	20	0.423	0.414	0.385	0.217	0.376	0.343	0.289	0.349
	100	0.399	0.151	0.357	0.311	0.389	0.347	0.328	0.326
	300	0.405	0.345	0.350	0.228	0.313	0.326	0.310	0.325
RAG-MpNet	20	0.472	0.414	0.380	0.204	0.366	0.308	0.316	0.351
	100	0.438	0.405	0.406	0.225	0.376	0.320	0.315	0.355
	300	0.425	0.362	0.361	0.223	0.328	0.315	0.320	0.333
MpNet - JinaAI (*top-k*=100)		↑ 0.039	↑ 0.254	↑ 0.049	↓ 0.086	↓ 0.013	↓ 0.027	↓ 0.013	↑ 0.029

Table 2: Performance of RAG-JinaAI and RAG-MpNet across different *top-k* values for various languages. The last row represents the difference between RAG-MpNet and RAG-JinaAI at *top-k*=100.

Method	*top-k*	4k	8k	16k	32k	64k	128k
RAG-JinaAI	20	0.383	0.322	0.309	0.257	0.181	0.178
	100	0.759	0.581	0.557	0.460	0.329	0.320
	300	0.989	0.950	0.847	0.696	0.471	0.453
$\Delta_{\text{max}-\text{min}}$		↓ 0.606	↓ 0.628	↓ 0.537	↓ 0.439	↓ 0.290	↓ 0.275
RAG-MpNet	20	0.517	0.431	0.412	0.345	0.248	0.243
	100	0.926	0.809	0.776	0.635	0.446	0.435
	300	0.989	0.996	0.980	0.905	0.661	0.628
$\Delta_{\text{max}-\text{min}}$		↓ 0.472	↓ 0.565	↓ 0.568	↓ 0.560	↓ 0.413	↓ 0.385

Table 3: Recall at different context windows for JinaAI and MpNET, respectively.

Type of Noise / Context Size	2k	4k	8k	10k	12k	Avg.
Synthetic Distractors (S)	0.73	0.72	0.71	0.72	0.70	0.72
Natural Distractors (N)	0.75	0.75	0.73	0.72	0.73	0.74
Random Noise (R)	0.75	0.76	0.75	0.75	0.74	0.75
N - S	↑ 0.01	↑ 0.03	↑ 0.02	0.00	↑ 0.02	↑ 0.02
R - S	↑ 0.02	↑ 0.04	↑ 0.04	↑ 0.03	↑ 0.04	↑ 0.04

Table 4: Performance comparison on using three types of distractors, namely synthetic, natural, and random noise. Results show that synthetic distractors confuse the model more than natural distractors or random noise.

	Basic Factoid QA				Yes/No | Simple Negation				Association				Fact Chaining				Counting				List / Sets				Indefinite Knowledge				
	16k	32k	64k	128k	16k	32k	64k	128k	16k	32k	64k	128k	16k	32k	64k	128k	16k	32k	64k	128k	16k	32k	64k	128k	16k	32k	64k	128k	
CoT	76	67	54	42	77	70	57	56	66	52	40	26	30	21	14	8	34	36	34	12	34	20	34	28	44	38	4	30	en
FS	69	63	61	62	81	64	59	55	60	56	56	46	21	11	8	8	22	18	22	20	44	44	36	44	36	6	8	24	
RAG MpNet	79	83	69	70	89	90	86	85	62	62	56	66	19	19	18	16	34	28	32	16	52	54	50	46	46	42	38	32	
RAG JinaAI	76	78	78	76	92	88	87	82	52	60	62	60	22	19	13	20	30	28	24	24	50	54	44	58	44	42	40	40	
CoT	69	57	28	32	61	42	56	50	48	54	24	8	12	6	11	9	26	36	32	32	28	28	18	10	50	44	18	28	es
FS	53	51	49	43	76	70	67	58	62	62	62	54	19	9	5	10	14	6	18	32	36	24	24	30	38	16	24	40	
RAG MpNet	60	57	58	55	81	83	81	65	62	58	62	56	20	20	19	22	12	16	12	10	42	38	46	36	38	32	32	34	
RAG JinaAI	63	54	62	61	77	83	76	76	62	60	58	62	21	22	19	16	14	16	18	14	48	36	38	38	36	28	26	28	
CoT	53	42	31	25	84	74	61	63	46	28	30	28	20	13	6	8	26	24	36	16	30	24	24	20	52	50	16	24	de
FS	68	44	50	57	73	69	65	63	64	60	58	64	22	19	15	8	22	16	28	36	38	38	28	30	36	30	52	40	
RAG MpNet	69	64	59	57	76	79	74	77	66	64	68	70	23	21	20	19	20	22	16	20	40	40	40	38	36	30	36	32	
RAG JinaAI	72	65	59	58	79	80	76	77	70	70	70	62	20	18	21	23	18	16	18	20	42	42	40	42	34	30	32	30	
CoT	56	46	33	25	58	51	53	48	40	32	8	18	9	6	1	1	36	32	40	34	36	40	32	24	48	44	0	10	hi
FS	37	32	29	30	75	66	62	58	34	20	26	14	9	8	8	6	24	22	36	38	30	30	34	32	48	54	46	46	
RAG MpNet	41	37	37	40	82	82	66	69	30	30	26	26	7	5	5	4	28	24	16	24	32	32	30	34	38	42	40	40	
RAG JinaAI	39	35	38	32	77	83	71	74	30	28	26	30	10	5	7	6	28	20	20	24	32	32	36	34	38	44	38	38	

Figure 2: Heatmap depicting performance for different tasks with increasing the context window ($16k$ to $128k$ tokens). All values represent accuracy scores on a percentage scale, with higher values getting darker shades and vice versa.

long contexts. For long-context evaluation, we compare accuracy at $4k$ tokens with the mean accuracy across all context lengths $\geq 32k$. Furthermore, Chinese prompts yield significantly lower scores, with a steady decline in reasoning as linguistic distance from English increases. Overall, performance across all `MLRBench` languages remains low, underscoring the difficulty of multilingual reasoning under long-context conditions (refer to Appendix Table 9). To explore this trend, we compare English with Spanish (second-best) and Chinese (lowest), shown in Figure 4(a). Accuracy declines with context length, particularly for prompt-only methods. Interestingly, RAG reduces the performance gap between English and Spanish across all context sizes. Figure 4(b) shows that performance drops more sharply in languages that are more linguistically distant from English[6]. Figure 4(c) compares RAG methods across languages. RAG-MpNet consistently outperforms RAG-JinaAI, especially in languages closer to English. However, for distant languages like Arabic and Chinese, both models perform similarly. This highlights a shared limitation in handling these languages effectively.

[6]We compute this using lang2vec (Littell et al., 2017), which provides vector-similarity functions to compute syntactic and phonetic distance between languages. These values are reported in Table 8

4.1 Task-wise Results

Heatmaps in Figures 2 and 3 show task-wise performance of selected methods across four context lengths: $16k$, $32k$, $64k$, and $128k$. The colour patterns reveal clear differences in task difficulty, with some tasks consistently underperforming across languages. Basic Factoid QA achieves the highest accuracy, indicating models can reliably retrieve isolated facts even at long context lengths. Yes/No (Negation) questions also perform well, but show more variation across languages. In contrast, tasks requiring multi-step reasoning, such as Fact Chaining and Argument Relations, yield low accuracy, highlighting limitations in sustained reasoning. Aggregation tasks (e.g., Counting, Lists/Sets) perform moderately but decline sharply in low-resource or morphologically complex languages. The steepest drop is observed in Indefinite Knowledge tasks, where accuracy remains consistently low across all settings, underscoring the challenge of epistemic reasoning under long-context conditions.

Retrieval versus Reasoning Figure 5 shows performance across the four task categories, namely Retrieval, Multi-hop Inference, Aggregation, and Uncertainty. Across all languages and context lengths, Retrieval consistently yields the

	Basic Factoid QA				Yes/No | Simple Negation				Association				Fact Chaining				Counting				List / Sets				Indefinite Knowledge				
	16k	32k	64k	128k	16k	32k	64k	128k	16k	32k	64k	128k	16k	32k	64k	128k	16k	32k	64k	128k	16k	32k	64k	128k	16k	32k	64k	128k	
CoT	76	67	54	42	77	70	57	56	66	52	40	26	30	21	14	8	34	36	34	12	34	20	34	28	44	38	4	30	en
FS	69	63	61	62	81	64	59	55	60	56	56	46	21	11	8	8	22	18	22	20	44	44	36	44	36	6	8	24	
RAG MpNet	79	83	69	70	89	90	86	85	62	62	56	66	19	19	18	16	34	28	32	16	52	54	50	46	46	42	38	32	
RAG JinaAI	76	78	78	76	92	88	87	82	52	60	62	60	22	19	13	20	30	28	24	24	50	54	44	58	44	42	40	40	
CoT	66	64	24	22	57	56	54	57	26	12	4	0	16	9	3	7	48	34	68	14	36	34	22	18	48	46	8	24	vi
FS	66	52	51	52	63	28	32	42	36	38	34	30	21	17	13	9	18	18	32	28	40	32	30	26	36	24	28	40	
RAG MpNet	69	68	61	63	70	66	64	65	34	38	40	46	19	15	17	19	22	20	16	18	44	42	34	34	36	26	32	32	
RAG JinaAI	70	73	71	70	70	69	59	61	42	34	40	40	22	20	20	19	18	26	18	20	40	36	44	38	30	30	34	34	
CoT	51	40	20	22	62	51	50	42	32	14	8	6	20	14	6	10	34	30	50	30	32	24	26	30	38	44	24	44	zh
FS	40	37	32	30	48	38	31	27	38	38	20	12	19	19	11	16	20	26	24	18	32	40	30	38	22	36	2	26	
RAG MpNet	38	41	36	29	51	51	50	45	46	36	30	26	14	14	17	18	30	28	34	34	36	36	26	26	22	26	24	18	
RAG JinaAI	36	41	37	33	52	54	52	45	42	40	30	34	15	14	13	15	20	22	22	32	38	38	30	30	22	22	22	20	
CoT	58	61	30	14	59	49	54	53	34	20	26	8	31	19	9	5	28	30	32	20	44	42	18	10	40	48	10	22	ar
FS	40	33	26	18	71	58	55	56	40	22	26	14	20	21	15	11	8	10	10	24	30	24	6	6	54	8	44	46	
RAG MpNet	42	47	36	31	79	79	71	73	34	32	28	22	17	19	18	14	10	12	16	12	28	32	32	30	54	44	48	52	
RAG JinaAI	47	47	37	41	80	78	67	69	36	36	34	32	17	19	16	14	8	14	4	10	34	34	32	26	46	46	54	44	

Figure 3: Heatmap depicting performance for different tasks with increasing the context window (16k to 128k tokens). All values represent accuracy scores on a percentage scale, with higher values getting darker shades and vice versa.

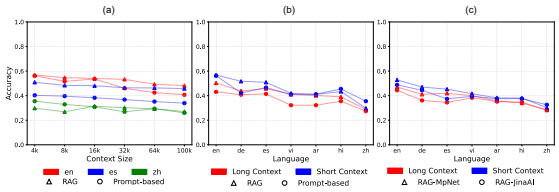


Figure 4: Performance comparison across languages: (a) Prompt-based vs. RAG methods for the best (en), second-best (es), and worst-performing (zh) languages. (b) Comparison of these methods across languages for short (4k) and long ($\geq$ 32k) context lengths. (c) Performance of two RAG methods across short and long context lengths.

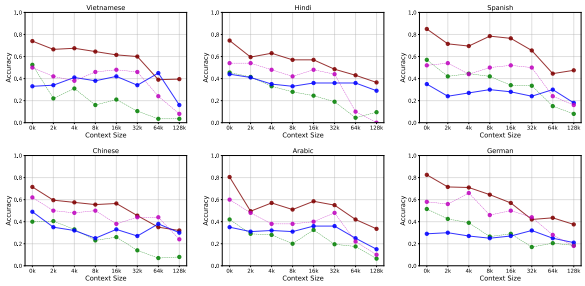


Figure 5: Task-wise performance of Llama-3.1-Instruct on selected languages, grouped into four categories as discussed in Section 2.3.

highest accuracy, indicating that models are effective at locating isolated facts. Aggregation slightly outperforms Multi-hop Inference, though both require integrating multiple pieces of information and present similar reasoning challenges. Uncertainty is the most difficult category, with the lowest accuracy across settings; models often respond with unwarranted confidence rather than expressing ambiguity or deferring judgment. So in summary, while LLMs perform well at surface-level retrieval, their performance declines sharply on tasks requiring deeper reasoning. The pronounced gap between retrieval and reasoning accuracy underscores the need for evaluation frameworks like `MLRBench` that go beyond recall-focused benchmarks. We conducted significance tests for Llama3.1-Instruct and found the results statistically significant. These are described in detail under Appendix E.

4.2 Ablation Study

1) Effect of top-k retrieval. We evaluate the effect of varying the number of retrieved sentences (*top-k* = 20, 100, 300) in the RAG pipeline. As shown in Figure 7(a), performance improves with higher *top-k* across context sizes, with notable gains at longer context lengths for both JinaAI and MpNet. This suggests RAG can partially mitigate information dispersion in long inputs.

2) Effect of distractor type. We compare synthetic, natural, and random noise distractors (Section B.4) for inflating context length. As shown in Figure 7(c), synthetic distractors slightly reduce performance, suggesting that structurally similar, in-distribution distractors make the task harder and thus reflect real-world difficulty.

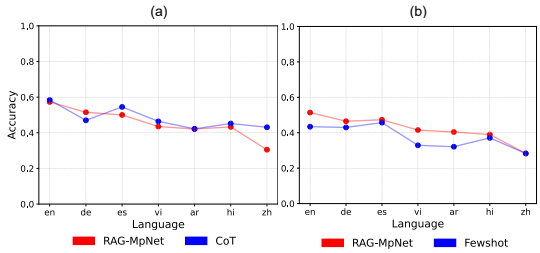


Figure 6: Performance comparison between the best RAG method and the best prompt-only method for (a) short context lengths ($4k - 16k$) and (b) long context lengths ($\geq 32k$).

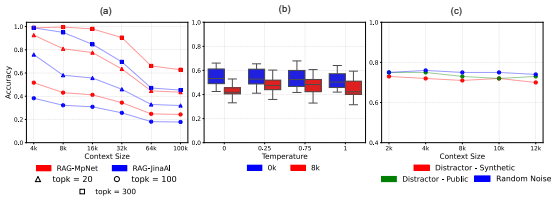


Figure 7: Ablation studies: (a) Performance of two different RAG methods with different numbers of retrieved documents. (b) Effect of temperature scaling at 0k and 8k context sizes. (c) Performance across different context sizes by virtue of the type of distractors used to increase the context size.

3) Effect of temperature sampling. We assess the impact of decoding strategy by varying temperature sampling at 0, 0.25, 0.5, and 1, respectively. As shown in Figure 7(b), accuracy remains largely unaffected, indicating output stability across sampling conditions. Along with our statistical significance tests (Appendix E) showing convergence beyond 250 samples.

5 Related Work

Recent advances in multilingual reasoning have been driven by improved in-context prompting (Tanwar et al., 2023), multi-step instruction following (Huang et al., 2023), and pre-translation methods (Ahuja et al., 2023) and supported by benchmarks like MGSM (Shi et al., 2022), XCOPA (Ponti et al., 2020), and XL-WiC (Raganato et al., 2020). However, most of this work focuses on short, isolated prompts, leaving extended multilingual contexts underexplored despite their relevance in real-world tasks spanning multiple sources and documents (Lee et al., 2022; Rozière et al., 2024; Shah et al., 2024). In long-context evaluation, *Needle-in-a-Haystack (NIAH)* tests (Kamradt, 2023) have become increasingly popular. However, current multilingual NIAH benchmarks (Agrawal et al., 2024; Hengle et al., 2024; Kim et al., 2025; Huang et al., 2025) primarily assess *retrieval* rather than *reasoning* over the long context (Goldman et al., 2024). Some recent efforts have sought to address this by proposing tasks beyond retrieval, such as Latent Structure Queries (LSQ) (Vodrahalli et al., 2024) and the bAbILong dataset (Kuratov et al., 2024). However, these remain mostly English-centric or monolingual. Parallely, the challenge of constructing realistic long-context benchmarks has led to a reliance on synthetic datasets (Kamradt, 2023; Hsieh et al., 2024a; An et al., 2023; Bai et al., 2024; Kim et al., 2025), which offer scalability and control, while realistic ones focus on practical tasks like summarisation (Kim et al., 2024) and QA (Kuratov et al., 2024; Shaham et al., 2023; Hengle et al., 2024). In our study, we build on these directions by combining synthetic evaluation with realistic task formats, extending bAbI into long, multilingual contexts that are framed in a SQuAD-style QA format (Rajpurkar et al., 2016).

6 Conclusion

This work introduces `MLRBench`, a synthetic benchmark to assess LLMs' reasoning capabilities over multilingual long contexts. `MLRBench` offers an evaluation framework with parallel prompts across seven languages and multiple context lengths. Unlike existing multilingual long-context benchmarks, `MLRBench` is explicitly designed to resist evaluation leakage and short-circuiting. Going beyond the popular *needle-in-a-haystack* test, `MLRBench` incorporates tasks that require multi-hop inference, aggregation, and epistemic reasoning. Through extensive experiments on an open-weight LLM, we observe a clear performance disparity between high- and low-resource languages, especially in tasks involving multi-step reasoning and uncertainty resolution. Off-the-shelf RAG methods help extend the *effective context length* but fall short of fully addressing the challenges of reasoning over extremely long contexts. Furthermore, performance is notably skewed toward retrieval tasks, with a significant accuracy drop in categories like epistemic reasoning, which indicates that current benchmarks may substantially overestimate LLMs' true long-context capabilities. In summary, our findings highlight that multilingual long-context reasoning remains an open and unsolved problem. Addressing this is crucial if we are to tackle long-term and long-tail challenges, especially when it comes to deploying these LLMs in global, high-stakes applications.

7 Limitations

Recent API-based models such as GPT-4 (OpenAI, 2024) and Claude (Anthropic, 2024) have shown impressive reasoning capabilities. However, due to their high cost, particularly given the large size of our evaluation prompts, we were unable to include them in our experiments. While we did experiment with a lightweight re-ranking model (JinaAI), future work could explore more complex and dedicated RAG pipelines, such as GraphRAG (Edge et al., 2025). We chose not to pursue this direction in the current work, as our goal was to evaluate retrieval using simple, off-the-shelf methods that introduce minimal computational overhead.

An interesting extension of `MLRBench` would be cross-lingual evaluation, where the question, passage, or background texts appear in different languages. Although `MLRBench` supports such experiments, this was outside the scope of our current study. Another promising direction for future work is a deeper interpretability analysis of model behavior in long-context settings, for instance, comparing retrieval versus reasoning performance or analyzing how models process and attend to information across long contexts. This could involve techniques from mechanistic interpretability or studying self-attention patterns, similar to recent work like (Hsieh et al., 2024b).

8 Ethics Statement

Finally, `MLRBench` is fully synthetic and did not involve any human annotators during its construction. As a result, this work does not require additional ethical review from an institutional review board. All models, datasets, and scientific artefacts used in this study have been properly cited where applicable. Our proposed `MLRBench` benchmark is completely synthetic. We have not involved any human annotators while curating `MLRBench` . Our work does not need any additional ethical review by an institutional review board. All the models, datasets, and scientific artefacts used in this study have been properly cited wherever required.

References

Ameeta Agrawal, Andy Dang, Sina Bagheri Nezhad, Rhitabrat Pokharel, and Russell Scheinberg. 2024. Evaluating multilingual long-context models for retrieval and reasoning.

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

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.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. Https://www.anthropic.com/news/claude-3-family.

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.

Bernd Bohnet, Kevin Swersky, Rosanne Liu, Pranjal Awasthi, Azade Nova, Javier Snaider, Hanie Sedghi, Aaron T Parisi, Michael Collins, Angeliki Lazaridou, Orhan Firat, and Noah Fiedel. 2024. Long-span question-answering: Automatic question generation and qa-system ranking via side-by-side evaluation.

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.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Omer Goldman, Alon Jacovi, Aviv Slobodkin, Aviya Maimon, Ido Dagan, and Reut Tsarfaty. 2024. Is it really long context if all you need is retrieval? towards genuinely difficult long context nlp.

Amey Hengle, Prasoon Bajpai, Soham Dan, and Tanmoy Chakraborty. 2024. Multilingual needle in a haystack: Investigating long-context behavior of multilingual large language models.

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.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in llms: Improving multilingual capability by cross-lingual-thought prompting.

Xu Huang, Wenhao Zhu, Hanxu Hu, Conghui He, Lei Li, Shujian Huang, and Fei Yuan. 2025. Benchmax: A comprehensive multilingual evaluation suite for large language models.

Yotam Intrator, Matan Halfon, Roman Goldenberg, Reut Tsarfaty, Matan Eyal, Ehud Rivlin, Yossi Matias, and Natalia Aizenberg. 2024. Breaking the language barrier: Can direct inference outperform pre-translation in multilingual LLM applications? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 829–844, Mexico City, Mexico. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Gregory Kamradt. 2023. Needle in a haystack- pressure testing llms.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024a. One thousand and one pairs: A "novel" challenge for long-context language models.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024b. One thousand and one pairs: A "novel" challenge for long-context language models.

Yekyung Kim, Yapei Chang, Marzena Karpinska, Aparna Garimella, Varun Manjunatha, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Fables: Evaluating faithfulness and content selection in book-length summarization.

Yekyung Kim, Jenna Russell, Marzena Karpinska, and Mohit Iyyer. 2025. One ruler to measure them all: Benchmarking multilingual long-context language models.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack.

Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien M. R. Arnold, Vincent Perot, Siddharth Dalmia, Hexiang Hu, Xudong Lin, Panupong Pasupat, Aida Amini, Jeremy R. Cole, Sebastian Riedel, Iftekhar Naim, Ming-Wei Chang, and Kelvin Guu. 2024. Can long-context language models subsume retrieval, rag, sql, and more?

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.

Mo Li, Songyang Zhang, Yunxin Liu, and Kai Chen. 2024. Needlebench: Can llms do retrieval and reasoning in 1 million context window?

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

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.

Meta AI. 2024. Llama 3.1 8B Instruct. `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`. Model and documentation available at `https://llama.meta.com/`.

Itai Mondshine, Tzuf Paz-Argaman, and Reut Tsarfaty. 2025. Beyond english: The impact of prompt translation strategies across languages and tasks in multilingual llms.

Leonie Monigatti. 2025. Evaluation metrics for search and recommendation systems. *Online Resources and Tutorials*. Accessed April 9, 2025.

Jina Reranker V2 Base Multilingual. 2024. Jina Reranker V2 Base Multilingual. Accessed: 2025.

Gabriel Nicholas and Aliya Bhatia. 2023. Lost in translation: Large language models in non-english content analysis.

OpenAI. 2024. Gpt-4 technical report.

Edoardo M. Ponti, Olga Majewska Goran Glava, Qianchu Liuand Ivan Vuliand, and Anna Korhonen. 2020. Xcopa: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709, Singapore. Association for Computational Linguistics.

Alessandro Raganato, Tommaso Pasini, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2020. XL-WiC: A multilingual benchmark for evaluating semantic contextualization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7193–7206, Online. 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.

Machel Reid, Nikolay Savinov, and Denis Teplyashin et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.

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.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2024. Code llama: Open foundation models for code.

Shalin Shah, Srikanth Ryali, and Ramasubbu Venkatesh. 2024. Multi-document financial question answering using llms.

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

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners.

Kosuke Takahashi, Katsuhito Sudoh, and Satoshi Nakamura. 2020. Automatic machine translation evaluation using source language inputs and cross-lingual language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3553–3558, Online. Association for Computational Linguistics.

Eshaan Tanwar, Manish Borthakur, Subhabrata Dutta, and Tanmoy Chakraborty. 2023. Multilingual llms are better cross-lingual in-context learners with alignment. *ArXiv*, abs/2305.05940.

Kiran Vodrahalli, Santiago Ontanon, Nilesh Tripuraneni, Kelvin Xu, Sanil Jain, Rakesh Shivanna, Jeffrey Hui, Nishanth Dikkala, Mehran Kazemi, Bahare Fatemi, Rohan Anil, Ethan Dyer, Siamak Shakeri, Roopali Vij, Harsh Mehta, Vinay Ramasesh, Quoc Le, Ed Chi, Yifeng Lu, Orhan Firat, Angeliki Lazaridou, Jean-Baptiste Lespiau, Nithya Attaluri, and Kate Olszewska. 2024. Michelangelo: Long context evaluations beyond haystacks via latent structure queries.

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.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks.

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.

A Experimental Setup

In this section, we describe the construction of our long-context prompt (P) for evaluating our task. Our structure is similar to the widely used needle-in-a-haystack paradigm (Liu et al., 2023), where sentences from a relevant passage (IP) are dispersed within the set of distractor passages (D) while preserving their original chronological order. As shown in Figure 9, each data point in our experiment is derived from the original bAbI dataset and consists of:

- Input Passage (IP) : A structured text composed of n sequentially-ordered independent sentences.
- Question (Q): A query related to the information within IP.
- Ground-truth Answer (A): The correct response to Q.

We formally define the input passage as:

$$IP = \{N_i \mid i = 1, 2, \ldots, n\}, \quad (1)$$

where each N_i represents an independent sentence within IP, following the strict chronological ordering constraint:

$$N_1 \prec N_2 \prec \cdots \prec N_n, \quad (2)$$

where $\prec$ denotes that sentence N_i must appear before N_{i+1} in the final prompt (see Figure 9). Additionally, we define the set of m unique distractor passages as:

$$D = \{D_j \mid j = 1, 2, \ldots, m\}, \quad (3)$$

where each D_j is an independent distractor passage from a synthetic corpus generated using an LLM.

Constructing long-context The long context P is constructed by placing sentences in IP randomly within the distractor passages D. This is done while maintaining the original order of sentences in IP. Formally, the sentences $\{N_1, N_2, \ldots, N_n\}$ from IP must retain their original ordering within P.

Let π be an injective map from the set $\{1, 2, \ldots, n\}$ to the set $\{1, 2, \ldots, m\}$ such that

$$\pi(i) < \pi(j) \quad \forall \quad \{i < j : 1 \leq i, j \leq n\} \quad (4)$$

where $\pi(i)$ represents the position of sentence N_i in P, and D provides the background text, acting as the "haystack". Thus, our setup allows one to dynamically increase the "haystack" to any arbitrary length by adjusting the number of distractor passages m.

Final prompt The final prompt $P = \{P_1, P_2, ..., P_{m+n}\}$ is constructed as follows:

$$P_k = \left\{ \begin{array}{ll} N_i, & \exists i : \pi(i) = k \\ D_j \in D, & \text{otherwise} \end{array} \right\} \quad (5)$$

where:

- D_j is sampled from D without replacement.
- $IP = \{N_1, N_2, \ldots, N_n\}$ remain interspersed throughout D but retain their relative order due to the property in Equation 4.
- The total length of P is significantly larger than IP alone, increasing the complexity of the retrieval.

B Dataset

B.1 Languages

`MLRBench` extends to seven typologically diverse languages: English (en), German (de), Spanish (es), Hindi (hi), Arabic (ar), Vietnamese (vi), and Simplified Chinese (zh). These languages vary substantially in terms of resource availability, ranging from high- to low-resource settings, as outlined by Joshi et al. (2020). They span four major language families: *Indo-European* (en, es, de, hi), *Afro-Asiatic* (ar), *Austro-Asiatic* (vi), and *Sino-Tibetan* (zh), and multiple writing systems – including Latin, Arabic, Devanagari, and logographic scripts. Our language selection also aligns with widely adopted multilingual evaluation benchmarks such as MLQA (Lewis et al., 2020), mMARCO (Bonifacio et al., 2022), and MLNeedle (Hengle et al., 2024). Thus, our proposed dataset allows for (i) a systematic analysis of how script variation and linguistic typology affect long-context comprehension in LLMs, (ii) examining the impact of training data availability across languages with differing resource levels, and (iii) a direct comparison to existing multilingual benchmarks.

B.2 Translation Process

We begin with the extraction of independent facts from the original English bAbI dataset. From an initial pool of $1k$ examples, approximately $8,500$ unique facts are identified. The passage, question, and answer are then translated into six languages, as described in Section 2.2, resulting in a parallel multilingual dataset. As illustrated in Table 9, this structure supports direct cross-lingual comparison and aligns with the format of existing multilingual

en	es	de	zh	vi	hi	ar
Sandra	Alejandra	Sabine	Meiying	Lan	Sangeeta	Layla
Daniel	Santiago	Lukas	Minghao	Dương	Dhruv	Omar
John	Juan	Hans	Jian	Minh	Jai	Youssef
Jeff	Javier	Jens	Zhihao	Phúc	Jatin	Fadi
Julie	Lucia	Julia	Xiuying	Thảo	Jyoti	Noor
Mary	María	Bertha	Lihua	Mai	Meera	Mariam
Fred	Federico	Fritz	Guowei	Hùng	Farid	Faris
Bill	Guillermo	Wilhelm	Weiguo	Bình	Kabir	Bilal

Figure 8: We replace the English bAbI characters with a culturally appropriate name in each of the target languages before translation.

Lang	Passage (Collection of Facts)	Question	Answer
en	*John travelled to the office. Mary journeyed to the kitchen. Daniel journeyed to the bathroom.*	*Where is Mary?*	**kitchen**
de	*Hans reiste ins Büro. Bertha reiste in die Küche. Lukas ging ins Badezimmer.*	*Wo ist Bertha?*	**Küche**
es	*Juan viajó a la oficina. María se dirigió a la cocina. Santiago se dirigió al baño.*	*¿Dónde está María*	**cocina**
zh	*健前往办公室。丽华向厨房走去。明浩走向卫生间。*	*丽华在哪里？*	**厨房**
vi	*Minh đi đến văn phòng. Mai đi vào bếp. Dương đi vào phòng tắm.*	*Mai ở đâu?*	**phòng bếp**
ar	سافر يوسف إلى المكتب. ذهبت مريم إلى المطبخ. ذهب عمر إلى الحمام.	أين مريم؟	**مطبخ**
hi	*जय ने कार्यालय की यात्रा की। मीरा रसोई की ओर बढ़ी। ध्रुव बाथरूम की ओर चला गया।*	*मीरा कहाँ है?*	**रसोईघर**

Figure 9: An example from our proposed `MLRBench` dataset. The original passage, question, and target from bAbI (Weston et al., 2015) are translated from English to six other languages – Spanish, German, Chinese, Vietnamese, Arabic, and Hindi. Thus, `MLRBench` supports a parallel data structure where each dataset instance is available across all languages. This makes it ideal for equitable multilingual evaluation.

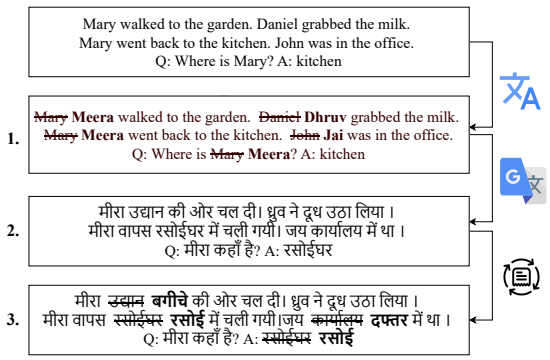


Figure 10: The three-step process used to translate the *passage*, *question*, and *answer* from bAbI to its Hindi equivalent form. The same process is followed for other selected languages from Table 9.

benchmarks such as MLQA (Lewis et al., 2020) and MLNeedle (Hengle et al., 2024). To make the dataset more representative across languages, we replace character names in bAbI with culturally appropriate alternatives for each target language, as shown in Table 8. Following this pre-processing step, we use the Google Translate API[7] to translate the passage, question, and answer from English to all selected languages [8]. Figure 10 provides an example of the three-step translation process. The resulting dataset preserves structural consistency and ensures that the semantics of each passage, question, and answer remain intact across all seven languages.

B.3 Translation Quality

One might argue that this approach introduces risks such as translation drift or quality issues, which could affect benchmark reliability. An obvious

[7] Google Translate API

[8] Note that we translate independent sentences and not the entire text at once. Details are provided in Appendix B

Dataset Instance	es	de	zh	vi	hi	ar
passages	0.89	0.90	0.85	0.88	0.84	0.85
questions	0.89	0.91	0.87	0.88	0.87	0.87
answers	0.89	0.89	0.87	0.87	0.87	0.87

(A) Translation similarity

Dataset Instance	es	de	zh	vi	hi	ar
passages	0.98	0.98	0.98	0.97	0.98	0.98
questions	0.98	0.98	0.98	0.99	0.99	0.98
answers	0.97	0.97	0.97	0.94	0.96	0.96

(B) Back-translation similarity

Table 5: (A) Semantic similarity between source (English) and target languages. We find that source and target texts have high semantic alignment for all languages. (B) The semantic similarity between the source (English) and back-translated texts from the target language. We observe a negligible error rate (high semantic similarity) between the source and back-translated texts for all languages. Results from (A) and (B) prove the robustness of our translation process.

alternative is to use human translators, which would offer higher reliability but are costly and difficult to scale. To understand why we opted against this, it is essential to consider the nature of bAbI dataset. As illustrated in Table 9, bAbI comprises discrete, atomic factual statements that follow a consistent subject–verb–object format. These are constructed from a closed set of entities, actions, and vocabulary items (Weston et al., 2015). Such a closed and formulaic structure substantially reduces the likelihood of translation errors arising from contextual ambiguity, co-reference, or idiomatic usage. Moreover, our translation process is applied at the sentence level, with each unit being an independent, context-free, and unambiguous factual statement (Weston et al., 2015). This further reduces the risk of any translation drift. Furthermore, we use the Google Translate API [9], which is one of the most high-performing and widely used machine translation tools [10]. We assess the quality of the translated data in `MLRBench` by conducting a translation quality estimation (QE) following prior work (Takahashi et al., 2020) [11]. As shown in Table 5 (A), we find high semantic similarity between the English source texts and their translations across all languages. Additionally, the back-translation results in Table 5 (B) indicate that the average information loss during translation is below 2%. We also manually reviewed a randomly selected 5% subset of `MLRBench`, and found no instances of misleading translations. Together, these findings support the quality and reliability of the translated data in `MLRBench`.

B.4 Background Texts

Background texts, or *distractors*, refer to segments of long-context that are irrelevant to the target task or set of facts the model is expected to reason over. Following prior work by Kamradt (2023) and Kuratov et al. (2024), our long-context prompts are constructed by interleaving relevant facts with multiple distractor passages. This setup mimics real-world cases where relevant information is often embedded within unrelated content. To preserve the effectiveness of distractors, it is important that they are not completely out-of-distribution with the task, as this can risk making the relevant information overly salient and undermine the purpose of evaluating a model's ability to filter from noise (Vodrahalli et al., 2024). To mitigate this, we employ semantic similarity as a filtering mechanism for distractor selection. Specifically, given a task instance (passage, query) and a corpus of background texts, we retrieve the top-k most semantically similar distractor passages based on cosine similarity. The value of k is adjusted dynamically to control the overall length of the prompt, allowing our setup to scale the long contexts to any arbitrary length as needed. As a source of background texts, we experiment with three types of distractors: (i) *synthetic distractors* generated using GPT-4, (ii) *natural distractors* sampled from a public text corpus, and (iii) *random noise* to serve as a control condition. Below, we describe each of them briefly.

Synthetic distractors Recent work, such as Hsieh et al. (2024a), has explored the use of synthetically generated distractors to control task complexity and make evaluation *a posteriori identifiable* [12]. Building on this idea, we construct our own corpus of in-domain synthetic distractor texts using GPT-4 (OpenAI, 2024). As illustrated in Table 9, passages in `MLRBench` are a collection of independent factual statements. In total, `MLRBench` includes $8,500$ unique facts, each following a common grammatical structure: subject (character) $\rightarrow$ action

[9] Google Translate API

[10] Performance report of Google Translation API

[11] We compute semantic-similarity between source and translated texts using XLM-Roberta-Large.

[12] If relevant information is *a posteriori identifiable*, it means that the model has to understand and reason over the information before arriving at an answer.

Task ID	Task Name	Total Test Samples (per language)	Prompt Instruction `{Instruct}`
1	Basic Factoid QA	200	Read the provided text, which contains facts about the locations of different individuals mixed with unrelated information. Answer the question based only on these facts. If a person appears in multiple locations, use their most recent location to determine the answer.
2	Yes/No or Negation	200	Read the provided text, which contains facts mixed with unrelated information. Answer the question while accurately interpreting negation with either "Yes" or "No," based only on the provided facts.
3	Fact Chaining	200	Read the provided text, which contains facts about different events mixed with unrelated information. Answer the question by identifying and combining multiple relevant facts.
4	Association	100	Read the provided text, which contains facts about different relationships mixed with unrelated information. Answer the question based on relationships involving two or more entities.
5	Counting	100	Read the provided text, which contains facts mixed with unrelated information. Answer the counting question by identifying and counting the relevant items, then provide the correct number.
6	List / Set	100	Read the provided text, which contains facts mixed with unrelated information. Answer the question by identifying all relevant items and listing them in the correct format.
7	Indefinite Knowledge	100	Read the provided text, which contains facts mixed with unrelated information. Answer the question based only on the given facts. If the necessary information is not explicitly provided, respond with "I don't know (IDK)".

Table 6: Prompt instructions and test sample distribution for each task type.

(verb) → object (or location/entity). These facts serve as seed examples for generating distractors. For each fact, we prompt GPT-4 to generate a short passage or story that is structurally similar (in terms of style, grammar, or tone) but semantically irrelevant to both the original passage and query. To introduce variety, we use four different prompt templates along with three levels of temperature sampling (0, 0.5, and 1.0). The specific templates used in this process are shown in Figures 12 - 15.

Natural distractors Many existing benchmarks source distractor texts from publicly available corpora such as Wikipedia articles, Paul Graham essays, or books. In this study, we use natural long-form passages from the MLQA dataset (Lewis et al., 2020) as our source of distractors. MLQA is well-suited for our multilingual setting, as it provides parallel passages across all seven languages included in `MLRBench`.

Random noise The distractor texts are random character-level tokens that carry no semantic meaning, grammatical structure, or lexical relevance to the primary task. Their purpose is purely to inflate context length without introducing any interpretive content. We use this as a control condition for long-context evaluation.

C Task Categories

`MLRBench` includes multiple reasoning tasks from bAbI covering four task categories – retrieval, multi-hop inference, aggregation, and handling uncertainty. Below, we outline each category and the corresponding tasks. Detailed examples of each task are provided in Appendix Table 7.

- **Retrieval.** This category follows the needle-in-a-haystack framework (Kamradt, 2023), and broadly tests the LLM's *associative recall* over the input context. Here, we include two tasks from the original bAbI dataset:

Task	Description	Example	Task Type
Basic Factoid QA	The answer can be retrieved from one directly relevant fact (independent statement) within the context.	Passage: "Daniel went back to the bedroom." Q: "Where is Daniel?" A: "bedroom"	**Retrieval (Needle in a Haystack)**. The model must locate one key sentence among possibly many. No inference or chaining.
Yes/No or Negation	The answer is binary (yes/no), possibly involving negation or contradiction in the context.	Passage: "Mary is not in the kitchen. Mary is not in the garden." Q: "Is Mary in the garden?" A: "no"	**Retrieval (Needle in a Haystack)**. The model needs to interpret a negated statement and confirm or deny the query.
Fact Chaining	The answer is derived by combining two or more facts across multiple sentences, often requiring tracking over time.	Passage: David bought a football. He forgot it in his bedroom {...} Mary gave it to Sandra {...} Sandra threw it away in the garden. Q: "Where was the football before the garden?" A: "bedroom"	**Multi-hop Reasoning**. The model is required to track a chain of actions across time — for instance where the football moved, and what happened before a specific event.
Association	The answer is inferred through understanding spatial, logical, or relational connections between entities.	Passage: "The bedroom is north of the bathroom. The bedroom is south of the kitchen." Q: "What is the kitchen north of?" A: "bedroom"	**Multi-hop Reasoning**. The model is required to understand and reason over relationships (e.g., directions) rather than just follow entity movements.
Counting	Requires counting the number of relevant entities or objects based on events in the Passage.	Passage: Apple is passed from Mary → Daniel → Mary → Sandra. Ball is passed from Mary → Sandra → Daniel. Q: "How many objects is Daniel carrying?" A: "one"	**Aggregation**. The model is required to track state changes across multiple characters and actions.
List / Set	The answer is a set or list of entities retrieved from the context, often based on participation or possession.	Passage: Mary picked up the milk, grabbed a football, and dropped her purse in the office. After going home, she left the football and picked the cup from kitchen. Q: "What is Mary carrying?" A: "milk, cup"	**Aggregation**. The model must be able to indentify and maintain entity states (e.g., what Mary picked up and didn't drop).
Indefinite Knowledge	The question is about a fact that the Passage makes ambiguous or uncertain — answer is "maybe" or "I don't know"	Passage: "Mary was in school at noon. {...} On her way home, Mary visited multiple stores." Q: "Which store did Mary visit last?" A: "I don't know"	**Refusal (Epistemic Reasoning)**. The model must recognise uncertainty, eliminate false options, and refuse over-committing to ambiguous information.

Table 7: Description and example of tasks covered in our proposed `MLRBench` dataset.

- **Basic factoid QA:** Task 1 consists of a set of questions where the answer is present as a single supporting fact among two or more irrelevant facts. The model must locate one key sentence among possibly many. The questions are framed in the *WhereIsActor* and *WhoWhatGave* formats (Weston et al., 2015), respectively.
- **Yes/No or negation:** Task 2 consists of a set of questions where the answer is binary (yes/no), and can be directly inferred from a single supporting fact within the passage. The questions are framed in the format *IsActorThere* (Weston et al., 2015), and test the model's ability to identify true/false statements or simple negation.

• **Multi-hop inference:** Multi-hop inference requires the LLM to combine two or more pieces of information (facts) from the context in order to arrive at the answer. This includes temporal, logical, and spatial chaining, where each fact contributes to the final inference.

  - **Fact chaining:** Task 3 consists of questions where the answer depends on two or more facts chained across multiple sentences,

often requiring tracking over time. The model is required to track a chain of characters and their actions sequentially, for instance, where an object moved and what happened before a specific event.
  - **Argument relations:** Task 4 consists of questions where the answer has to be inferred by understanding the relationship or interplay between characters. The model must reason over relationships (e.g., spatial directions or entity roles) rather than follow character movements.

- **Aggregation.** Aggregation tasks test the language models' ability to compile or summarise different pieces of information. This includes counting specific entities or compiling sets of items across the context. Unlike multi-hop reasoning, aggregation focuses on synthesis rather than chaining (Vodrahalli et al., 2024).
  - **Counting:** In Task 5, the questions require counting the number of relevant entities or objects based on events in the passage. The model is required to track state changes across multiple characters and actions, making it challenging.
  - **List / Sets:** In Task 6, the answer is a set or list of entities retrieved from the context, often based on participation or possession. The language model must identify and maintain character and object states (e.g., what someone picked up and didn't drop).

- **Uncertainty.** Uncertainty or epistemic reasoning evaluates the language models' ability to recognise ambiguity or incomplete knowledge and respond appropriately. Rather than guessing, the model must either eliminate impossibilities or refrain from over-committing to ambiguous inputs.
  - **Indefinite knowledge:** In Task 7, the question is about a fact that the story makes ambiguous or uncertain. The model must recognise uncertainty, eliminate false options, and avoid over-committing to ambiguous information.

D Prompting

Prompt Templates For all our experiments, we follow a standard Instruction → Context → Question prompt format. Each evaluation prompt begins with a task-specific instruction, as defined in Table 6. In-context demonstrations for few-shot prompting are retrieved via semantic matching (*top-k* selection) using sentence-transformers[13] (Reimers and Gurevych, 2020). As context and few-shot examples form the main body of the prompt, they are enclosed under the <context></context> and <example></example> tags, respectively. To avoid any post-processing step, we force the LLM to format its output in JSON. Prompt templates used for ZS, ICT, CoT, and FS are defined in Figures 16, 17, and 18, respectively.

Language	Syntactic Distance	Phonological Distance
en	0.00	0.0002
de	0.42	0.3277
es	0.40	0.3433
hi	0.59	0.3433
vi	0.57	0.4270
zh	0.57	0.5687
ar	0.57	0.5687

Table 8: Syntactic and phonological distance between English and other languages computed using lang2vec.

RAG We adopt a simple RAG pipeline based on $top-k$ retrieval. As defined in Section 3.1, an evaluation prompt P consists of n input passages IP interspersed within m distractor texts D. The retriever models encode and re-rank the sentences within P based on their relevance to the question Q. Subsequently, a new condensed prompt is constructed using the $top-k$ most relevant sentences with respect to Q, while preserving the original chronological order of sentences from both IP and D. Finally, this new RAG prompt is used for inference experiments.

E Statistical Tests

We conduct binomial significance tests on our main results for Llama3.1-Instruct. Figure 11 shows the significance tests across four settings: baseline, short-context, and long-context, respectively. These tests are performed over evaluation (test) sets of increasing size, ranging from 10 to 500 samples.

We find that across all selected languages, the exact accuracy begins to stabilize between 250 and 500 samples. Additionally, the standard error decreases significantly as the number of evaluation samples exceeds 250. This indicates that using

[13] We use the multilingual Sentence-BERT model to retrieve *top-k* instances from the dev set, such that they are most relevant to both the passage and question while ensuring they do not contain the answer.

	Context size	Prompt based methods	RAG methods	Average
en	$4k$	0.562	0.568	0.565
	$\geq 32k$	0.431	0.503	**0.467**
de	$4k$	0.422	0.517	0.470
	$\geq 32k$	0.408	0.438	0.423
es	$4k$	0.466	0.509	0.488
	$\geq 32k$	0.414	0.458	0.436
hi	$4k$	0.455	0.434	0.444
	$\geq 32k$	0.355	0.391	0.373
ar	$4k$	0.411	0.410	0.411
	$\geq 32k$	0.322	0.401	0.362
zh	$4k$	0.355	0.298	0.327
	$\geq 32k$	0.274	0.289	0.281
vi	$4k$	0.410	0.419	0.414
	$\geq 32k$	0.322	0.410	0.366

Table 9: Performance of prompt-based and RAG methods for $4k$ and for longer context lengths ($\geq 32k$) for different languages.

		RAG-JinaAI	RAG-MpNet	ZS	ICT	CoT	FS
Short Context	**en**	0.526	0.573	0.536	0.540	0.584	0.492
	de	0.492	0.515	0.481	0.494	0.470	0.529
	es	0.415	0.500	0.459	0.466	0.545	0.430
	vi	0.416	0.435	0.378	0.391	0.464	0.403
	hi	0.428	0.433	0.423	0.417	0.452	0.425
	ar	0.410	0.421	0.367	0.386	0.422	0.413
	zh	0.336	0.305	0.238	0.238	0.431	0.416
	Avg.	0.432	**0.455**	0.412	0.419	0.481	0.444
Long Context	**en**	0.480	0.514	0.434	0.437	0.418	0.434
	de	0.410	0.465	0.432	0.444	0.327	0.430
	es	0.382	0.473	0.419	0.410	0.386	0.457
	vi	0.404	0.415	0.302	0.326	0.331	0.329
	hi	0.392	0.390	0.366	0.359	0.324	0.371
	ar	0.398	0.404	0.321	0.327	0.320	0.321
	zh	0.294	0.283	0.240	0.220	0.297	0.283
	Avg.	0.394	0.421	0.359	0.360	0.343	0.375

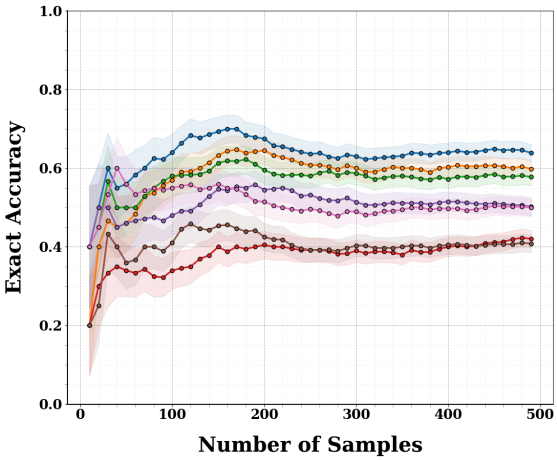
Table 10: Performance of prompt-based and RAG methods across context lengths: "Short Context" ($4k$–$16k$), "Long Context" ($\geq 32k$). For short contexts, RAG-MpNet and cot perform best among RAG and prompt-based methods, respectively. For long contexts, RAG-MpNet and few-shot perform best.

250 or more samples is sufficient to yield reliable and consistent evaluation results. Since our main experiments are conducted using 500 samples per language, we can confidently state that our results are statistically significant. We detail our hypothesis testing approach below.

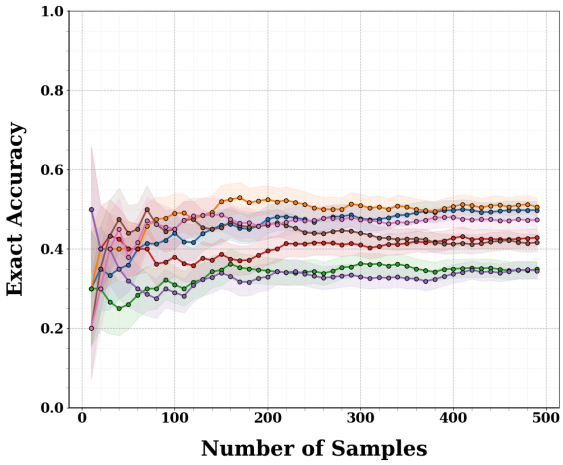
Let $\text{Binomial}(1, p)$ represent the binomial distribution, where p is the probability of success on an individual trial. The standard error (SE) is computed using the formula:

$$SE = \sqrt{\frac{p(1-p)}{s}}, \tag{6}$$

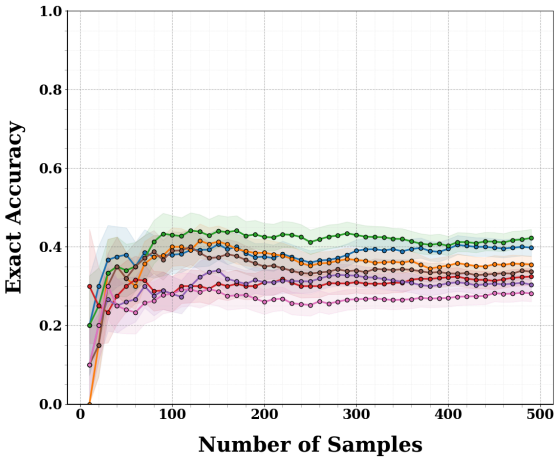
where s is the number of trials (i.e., evaluation samples). We vary s from 10 to 500 in increments of 10, i.e., 20, 30, 40, and so on – randomly sampling instances from `MLRBench` for each selected language at every step.



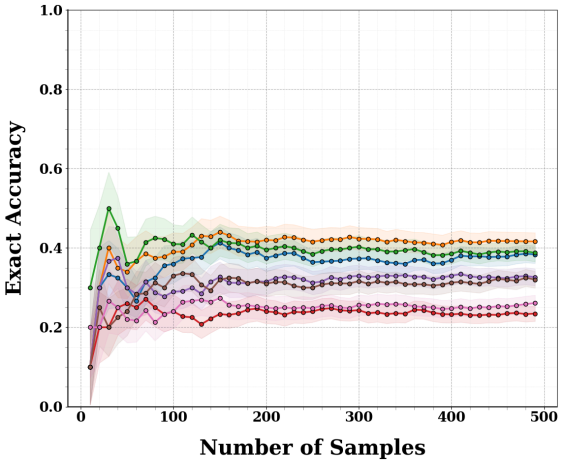
(a) Mean and standard error for selected languages at baseline (no context).



(b) Mean and standard error for selected languages at short context ($4k$).



(c) Mean and standard error for selected languages at long context ($32k$).



(d) Mean and standard error for selected languages at long context ($128k$).

Figure 11: Exact accuracy of Llama 3.1-Instruct after varying the sample size in the evaluation (test) set. Solid lines represent accuracy, while shaded areas indicate the standard error. Legend (by color): en (blue), de (orange), es (green), zh (red), vi (purple), hi (brown), ar (pink).

You are given an independent factual statement from the bAbI dataset, which follows the Subject → Action → Object structure.

Sentence:
{Independent Factual Statement}

Task:
Generate a story or passage (100-200 words) that captures the essence of the action but never mentions the subject or the object.

Constraints:
- The action may appear, but it must be used in a completely different meaning or context than in the original sentence.
- The passage must contain multiple sentences that expand on the topic with background information, examples, or applications.
- The passage must have no semantic or narrative connection to the original sentence.
- The passage must be complex enough to act as a distractor when answering a question about the original sentence.
- The passage could include any topic such as science, sports, economics, politics, medicine, philosophy, history, or technology.
- The passage must be in the same language as the input sentence.

Format your response strictly in JSON {"Distractor Passage": }

Figure 12: Prompt template for generating distractors based on an independent statement (fact) in `MLRBench`, constrained to capture only the action, omitting subject & object.

You are given an independent sentence from the bAbI dataset, which follows the Subject → Action → Object structure.

Sentence:
{Independent Factual Statement}

Task:
Generate a short story or passage (100-200 words) that includes the subject but never mentions the action or the object.

Constraints:
- The action may appear, but it must be used in a completely different meaning or context than in the original sentence.
- The passage must contain multiple sentences that expand on the topic with background information, examples, or applications.
- The passage must have no semantic or narrative connection to the original sentence.
- The passage must be complex enough to act as a distractor when answering a question about the original sentence.
- The passage could include any topic such as science, sports, economics, politics, medicine, philosophy, history, or technology.
- The passage must be in the same language as the input sentence.

Format your response strictly in JSON {"Distractor Passage": }

Figure 13: Prompt template for generating distractors based on an independent statement (fact) in `MLRBench`, constrained to include only the object, omitting subject & action.

You are given an independent sentence from the bAbI dataset, which follows the Subject → Action → Object structure.

Sentence:
{Independent Factual Statement}

Task:
Generate a story or passage (100-200 words) that describes the object but never mentions the subject or the action.

Constraints:
- The object may appear, but it must be used in a completely different meaning or context than in the original sentence.
- The passage must contain multiple sentences that expand on the topic with background information, examples, or applications.
- The passage must have no semantic or narrative connection to the original sentence.
- The passage must be complex enough to act as a distractor when answering a question about the original sentence.
- The passage could include any topic such as science, sports, economics, politics, medicine, philosophy, history, or technology.
- The passage must be in the same language as the input sentence.

Format your response strictly in JSON {"Distractor Passage": }

Figure 14: Prompt template for generating distractors based on an independent statement (fact) in `MLRBench`. The distractor must capture the action, omitting the subject & object.

You are given an independent sentence from the bAbI dataset, which follows the Subject → Action → Object structure.

Sentence:
{Independent Factual Statement}

Task:
Generate a story or passage (100-200 words) that includes the subject, action, and object but in a completely different context.

Constraints:
- The subject, action, and object must all appear, but they must be used in a completely different meaning or context than in the original sentence.
- The passage must contain multiple sentences that expand on the topic with background information, examples, or applications.
- The passage must have no semantic or narrative connection to the original sentence.
- The passage must be complex enough to act as a distractor when answering a question about the original sentence.
- The passage could include any topic such as science, sports, economics, politics, medicine, philosophy, history, or technology.
- The passage must be in the same language as the input sentence.

Format your response strictly in JSON {"Distractor Passage": }

Figure 15: Prompt template for generating distractors based on an independent statement (fact) in `MLRBench`, with no constraints on subject, action, and object.

Zero-shot Prompt Template

Read the provided text, which contains facts about the locations of different individuals mixed with unrelated information. Answer the question based only on these facts. If a person appears in multiple locations, use their most recent location to determine the answer.

<context>
Alejandra viajó a la cocina. Santiago viajó al baño. Juan salió al pasillo. Alejandra se dirigió al dormitorio.
</context>

QUESTION: ¿Dónde está Santiago

Format your response strictly in JSON, and output the answer only. { "ANSWER": }

Translation Prompt Template

Read the provided text carefully. First, translate all non-English words and phrases into English while keeping the original meaning intact. Then, extract only the relevant facts about the locations of different individuals. If a person appears in multiple locations, use their most recent location to determine the answer.

<context>
美英走到厨房。明浩去了卫生间。简走到走廊上。美英朝卧室走去。
</context>

QUESTION: 明浩在哪里？？

Format your response strictly in JSON, and output the answer only. { "ANSWER": }

Figure 16: Evaluation prompt templates used in *zeroshot* and *in-context translation* experiments.

Chain-of-Thought (COT) Prompt Template

Read the provided text carefully. First, translate all non-English words and phrases into English while keeping the original meaning intact. Then, break down the reasoning step-by-step before answering the question.

Step 1: Identify the two entities mentioned in the question.
Step 2: Extract the relevant relationship between these entities.
Step 3: Translate any non-English words into English while maintaining meaning.
Step 4: Use logical reasoning to determine how the entities relate.

<context>
Der Flur liegt westlich des Badezimmers. Die Küche liegt östlich des Badezimmers.
</context>

QUESTION: Wovon liegt das Badezimmer westlich?

First, generate a detailed step-by-step explanation of your reasoning. Then, provide the final answer in **strict JSON format**:

```
{
   "reasoning": "Step-by-step explanation here",
   "ANSWER": "Final answer here"
}
```

Figure 17: Evaluation prompt template used for all *chain-of-thought (CoT)* experiments.

Few-shot Prompt Template

Read the provided text, which contains facts about the locations of different individuals mixed with unrelated information. Answer the question based only on these facts. If a person appears in multiple locations, use their most recent location to determine the answer.

<example>
संगीता रसोई में चली गयी. संगीता वापस बाथरूम में चली गयी. मीरा शयनकक्ष की ओर बढ़ी। जय बाथरूम की ओर चला गया। संगीता बेडरूम में चली गयी. संगीता ने दालान की ओर यात्रा की। संगीता ने बगीचे की यात्रा की। संगीता बाथरूम की ओर चली गयी. ध्रुव रसोई की ओर गया। जय ने कार्यालय की ओर प्रस्थान किया।
QUESTION: संगीता कहाँ है?
ANSWER: स्नानघर
</example>

<example>
वहां संगीता को सेब मिला. जय बाथरूम में चला गया. ध्रुव वापस दालान में चला गया। संगीता रसोई में चली गयी. संगीता बाथरूम की ओर चली गयी. जय दालान में चला गया.
QUESTION: क्या संगीता ऑफिस में है?
ANSWER: नहीं
</example>

<context>
संगीता रसोई में पहुंची. ध्रुव बाथरूम की ओर चला गया। जय दालान में चला गया. संगीता शयनकक्ष की ओर चली गई।
</context>

QUESTION: ध्रुव कहाँ है?

Format your response strictly in JSON, and output the answer only. { "ANSWER": }

Figure 18: Evaluation prompt template used for *fewshot* experiments.

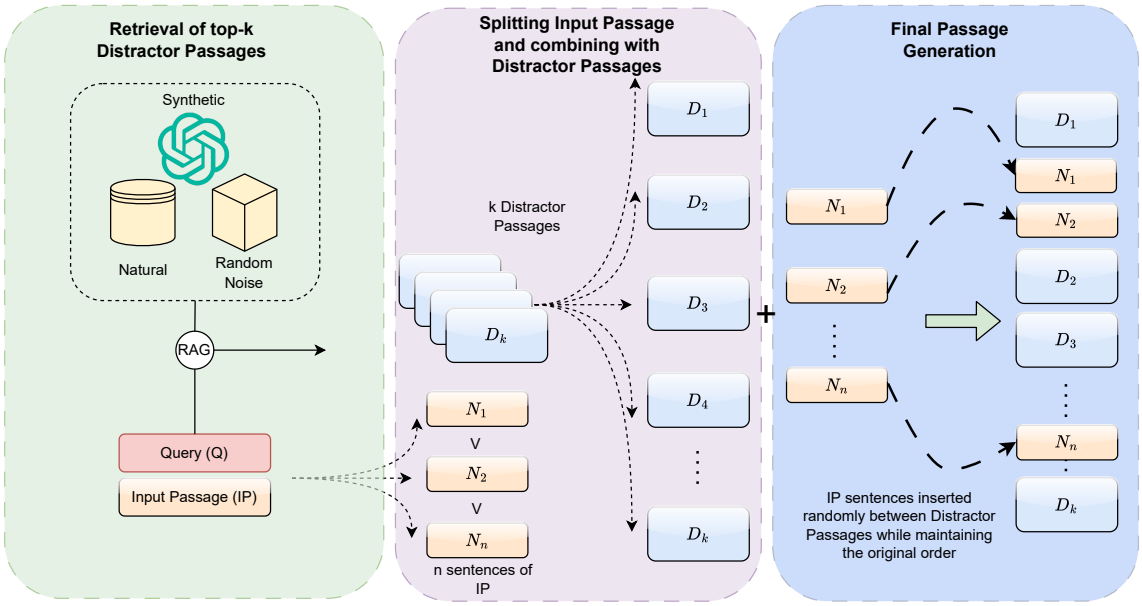


Figure 19: Overview of our pipeline for creating `MLRBench`. An Input Passage (IP) is split into n sentences. Given IP and the query, top-k distractor passages are retrieved from sources — (1) synthetic passages from GPT-4, (2) public datasets, and (3) random token-level noise. The independent sentences from IP are randomly inserted between distractors but maintain their original order. The value of k is dynamically adjusted to control the final context length.