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

Amey Hengle♥ **Prasoon Bajpai**♥ **Soham Dan**♡* **Tanmoy Chakraborty**♥,♣
♥Indian Institute of Technology Delhi, India ♣Indian Institute of Technology Delhi, Abu Dhabi
♡Microsoft
`{ameyhengle, prasoonbajpai786}@gmail.com`
`sohamdan@microsoft.com`, `tanchak@iitd.ac.in`

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 bAbI-style tasks that test multi-hop inference, aggregation, and epistemic reasoning. Spanning seven languages, we design `MLRBench` 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 the improved evaluation and training of multilingual LLMs.[1]

1 Introduction

The landscape of Large Language Models (LLMs) has undergone a remarkable transformation in recent years, particularly in their ability to process long-input sequences. Just two years back, LLMs, including early versions of GPT-3 and Llama, used to support a context window of no more than 4*k* tokens in both input and output (Brown et al., 2020). Fast forward to 2025, the context size of these models has expanded by several orders of magnitude, with newer LLMs like Gemini-2.0 capable of processing up to a million tokens at once (Reid et al., 2024). This dramatic increase in context window has unlocked new capabilities, allowing LLMs to infer from extremely long documents. As a result, their adoption and deployment have surged across a wide range of global-facing applications. For instance, applications involving dialogue systems, multi-document question answering, summarising long documents, or debugging lengthy code benefit from LLM's ability to retain and infer over-dispersed information within and across documents (Lee et al., 2022; Rozière et al., 2024; Shah et al., 2024).

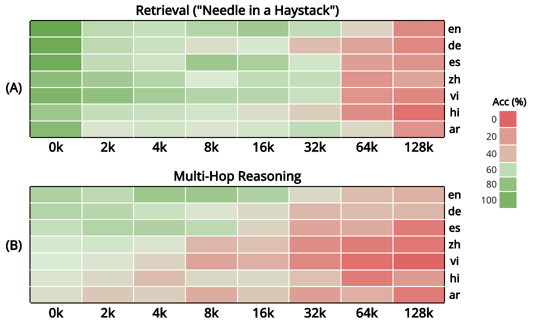


Figure 1: Question-answering performance of Llama-3.1-Instruct on two tasks: (a) simple retrieval/recall versus (b) multi-hop reasoning. As context length increases, the performance drop in (b) is sharper than that in (a), especially for non-Latin languages. This underscores the difficulty of task (b)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

[*] This work was done independent of position at Microsoft.
[1] The source code and dataset are available on Github and HuggingFace 🤗 respectively.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6128–6152
March 24-29, 2026 ©2026 Association for Computational Linguistics

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. There is a growing research interest in evaluating the long-context behavior of LLMs, although most of it is predominantly limited to monolingual English settings (Bai et al., 2024; Liu et al., 2023; Hsieh et al., 2024). Because realistic long-context evaluations are complex and resource-intensive (Karpinska et al., 2024), studies rely on synthetic benchmarks as *proxies* for real-world performance (Hsieh et al., 2024; Bai et al., 2024). One such widely adopted proxy evaluation is the *Needle-in-a-Haystack* (NIAH) test (Kamradt, 2023), which frames the task as a retrieval problem – measuring whether a model can locate specific facts within lengthy input contexts. Recent evaluation benchmarks are built on this framework by including multiple needles (Li et al., 2024; Hsieh et al., 2024) and question-answering (Bohnet et al., 2024; Liu et al., 2023; Zhang et al., 2024). Although these benchmarks certainly add a good amount of depth, complexity, and realism to the task, they still adhere to the core evaluation principle of NIAH.

Recently, several studies have drawn attention to the design limitations of retrieval-centric NIAH tests, particularly their lack of effectiveness in evaluating LLM's reasoning capabilities (Goldman et al., 2024; Karpinska et al., 2024). The central argument is that a model's ability to *retrieve independent facts* from long contexts does not necessarily imply the ability to *synthesize information* across those contexts (Vodrahalli et al., 2024). In other words, successful retrieval alone should not be conflated with true comprehension. Although LLMs may effectively extract isolated pieces of information, even from extended inputs, this does not guarantee they can follow logical connections, resolve contradictions, or maintain coherent reasoning over time (Karpinska et al., 2024). This is shown empirically in Fig 1, where we compare model performance on retrieval- and reasoning-focused tasks under long-context settings. As the context length increases, the model maintains relatively stable performance on retrieval tasks but struggles significantly with reasoning tasks. The gap is particularly pronounced in non-Latin languages. These findings suggest that existing benchmarks may overestimate LLMs' true capacity to process and reason over extremely long input sequences (Vodrahalli et al., 2024).

Apart from these design limitations, some practical issues also limit the effectiveness of existing multilingual benchmarks. Datasets like MLNeedle (Hengle et al., 2025) and mLongRR (Agrawal et al., 2024) risk data leakage, since they draw from open-source text. This raises concerns of *evaluation leakage*[2] or *short-circuiting*[3], which may compromise 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 different bAbI-style reasoning tasks. `MLRBench` can be scaled to any arbitrary context length and is designed to resist evaluation leakage and short-circuiting. Furthermore, rather than focusing solely on surface-level retrieval, tasks in `MLRBench` require symbolic reasoning over multiple facts, implicit relationships, multi-hop connections, temporal or spatial awareness, and so on. These task categories are *minimal yet orthogonal*. 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 bAbI-style 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. Our evaluation setup draws inspiration from BABILong (Kuratov et al., 2024), and extends it to the multilingual setting with variable distractors.
- Through extensive experiments, we study how changes in language, task, and prompting method affect an LLM's ability to reason

[2] Evaluation leakage occurs when the model has either fully or partially seen the evaluation (test) set during pretraining, which may result in misleadingly high-performance scores.

[3] Short-circuiting happens when a model doesn't actually infer the target task (in this case, reasoning over long contexts); but rather "cheats" using parametric knowledge or other shortcuts (Lee et al., 2024).

across long, multilingual contexts. Our results reveal a significant performance gap between high- and low-resource languages, particularly for tasks involving multi-step reasoning or uncertainty resolution.

- We undertake an exhaustive analysis of retrieval and reasoning tasks, finding that existing benchmarks may overstate an LLM's 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 an 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 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.

[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).

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 C.

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 D, with detailed task instances shown in Appendix Table 7.

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 B.

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 (Grattafiori and et al., 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 occurs in-context, thereby avoiding 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 E.

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., 2025), we apply normalization (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 retrieved for a given question*, calculated as the

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

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 32*k*, 64*k*, and 128*k*). "Drop %" denotes the percentage drop in accuracy when moving from context size 2*k* to 128*k* tokens. "Baseline" refers to the no-distractor condition, where the model is evaluated on the original short-context bAbI input only (<= 2k 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 32*k* while the model claims to support 128*k* tokens.

size of the intersection between the retrieved and relevant sets, divided by the total number of relevant sentences (Monigatti, 2025).

4 Experimental Results

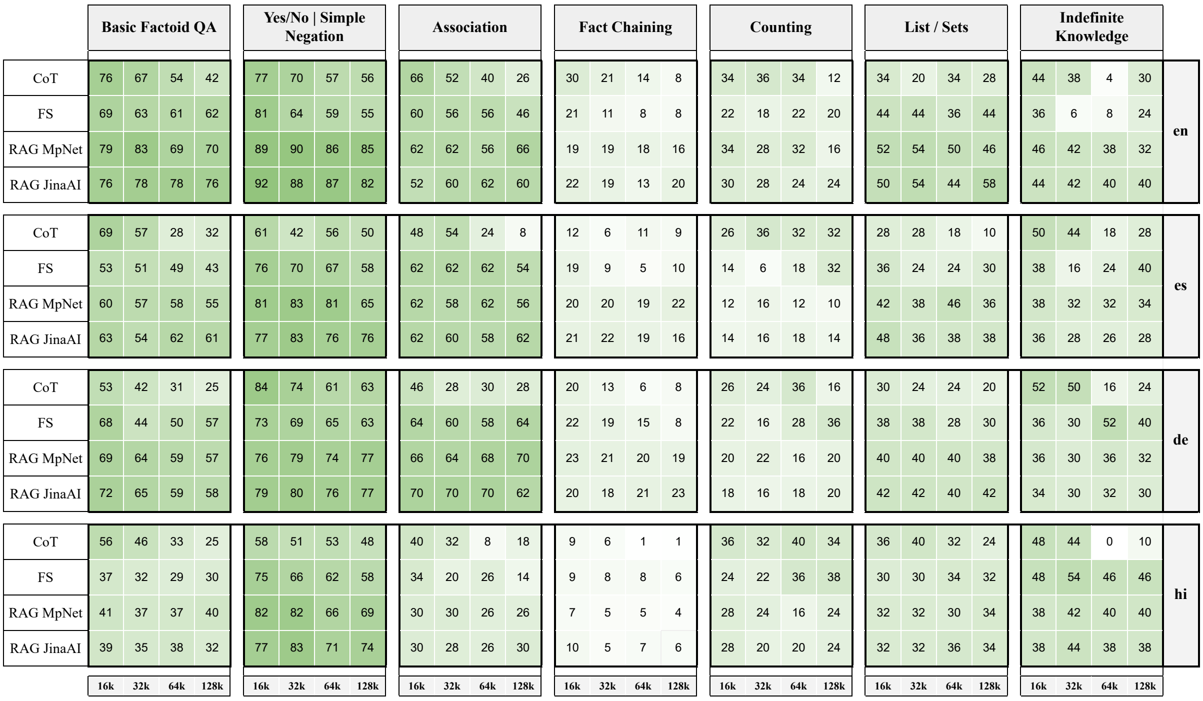
We conduct inference experiments across a range of context sizes, beginning with the baseline setup (no distractors) and extending up to 128*k* tokens. Table 1 presents a summary of results for all baselines, averaged across the seven languages [6]. Notably, even the best-performing method at the baseline, 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 32*k* 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 consistently outperform ZS and ICT across all context lengths and languages, with an average accuracy margin of at least 6%. However, prompt-based methods are also highly sensitive to an increase in context size. Their performance declines significantly, particularly beyond a context window of 16*k* tokens. For instance, CoT accuracy drops from 0.598 at baseline to 0.253 at 128*k*, reflecting a relative 57.6% performance loss. Prompt-based methods lose between 38% to 60% of their baseline accuracy at longer context windows (≥ 32*k*), and nearly all prompting methods begin to deteriorate sharply after 8*k* to 16*k* 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 2*k* and 128*k* 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 4*k*–16*k* 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 (≥ 32*k*), the highest among all methods. Furthermore, we observe that RAG-MpNet outperforms RAG-JinaAI at both short and long contexts (Appendix Table 11).

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

[6] Unless stated otherwise, baseline refers to the no-distractor setting, where the input consists only of the original bAbI passage (short-context, <= 2k tokens), and serves as the reference point for measuring long-context performance.



	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 task-wise performance with increasing context length (16k–128k tokens) for a subset of languages (English, German, Spanish, and Hindi). All values represent accuracy scores (in %), with darker shades indicating higher performance.

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*	4*k*	8*k*	16*k*	32*k*	64*k*	128*k*
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
$\Delta_{\max-\min}$		↓ 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
$\Delta_{\max-\min}$		↓ 0.472	↓ 0.565	↓ 0.568	↓ 0.560	↓ 0.413	↓ 0.385

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

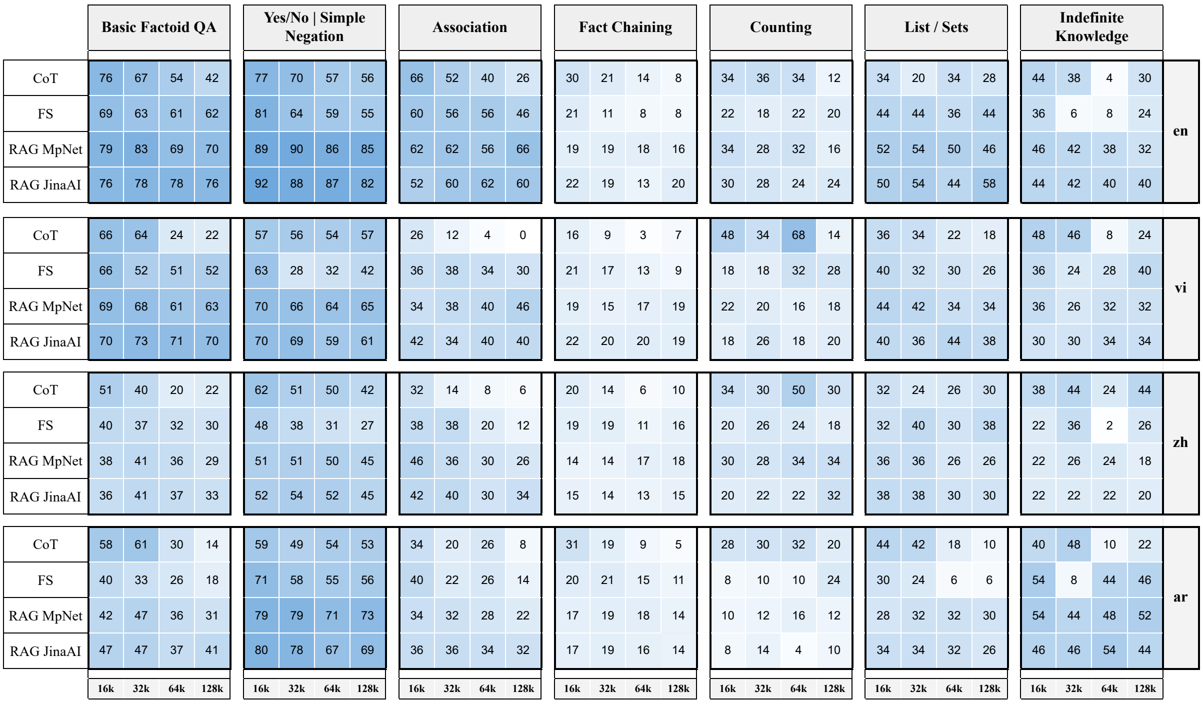
remains 20%–40% below the best short-context baseline. This suggests that while RAG extends usable context, significant progress is still needed to handle extremely long inputs effectively.

Trend 3: Multilingual performance drops sharply with increasing linguistic distance from English. Unsurprisingly, English achieves the highest performance across both short and long contexts. For long-context evaluation, we compare accuracy at 4*k* tokens with the mean accuracy across all context lengths ≥32*k*. 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 10). 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[7]. Figure 4(c) compares RAG methods

Type of Noise / Context Size	2*k*	4*k*	8*k*	10*k*	12*k*	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.

[7] We compute this using lang2vec (Littell et al., 2017), which provides vector-similarity functions to compute syntactic



	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 task-wise performance with increasing context length (16k–128k tokens) for the remaining languages. All values represent accuracy scores (in %), with darker shades indicating higher performance. This figure reports the same metric and experimental setup as Figure 2 and should be read as its complementary language-wise view.

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 effectively handling these languages.

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.

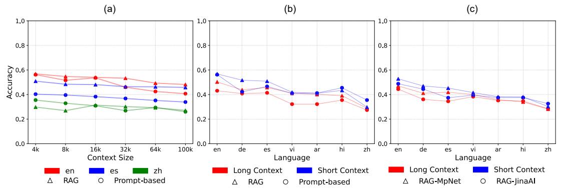


Figure 4: Performance comparison across languages: (a) Prompt-based vs. RAG methods for the best (en), second-best (es), and worst-performing (zh) languages. (b) Comparison of these methods across languages for short ($4k$) and long ($\geq 32k$) context lengths. (c) Performance of two RAG methods across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

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 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

and phonetic distance between languages. These values are reported in Table 8

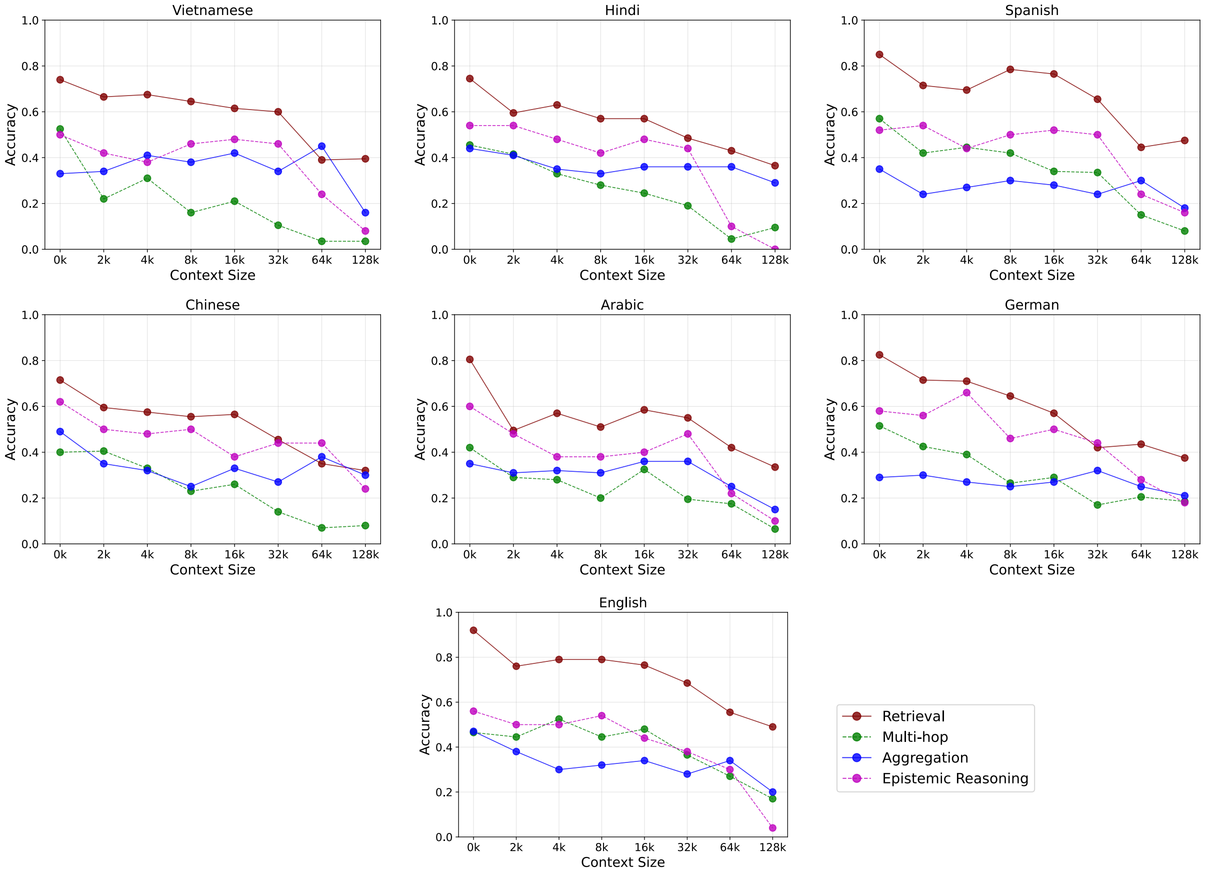


Figure 5: Task-wise performance of Llama-3.1-Instruct on selected languages, grouped into four categories as discussed in Section 2.3.

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 F.

Effective Context Size Table 1 reports the effective context window (ECW) for all baselines, defined as the maximum context length at which accuracy stays within 30% of the no-distractor baseline. The best prompt-only and RAG methods achieve ECWs of roughly 16k and 32k tokens, respectively, indicating the model effectively uses only 25%–30% of it's advertised context window.

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.

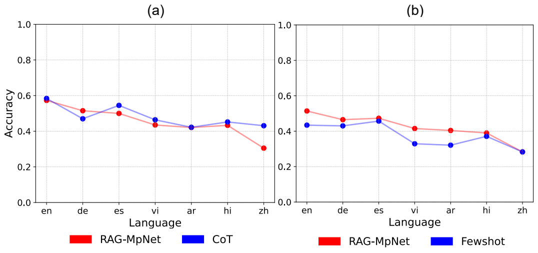


Figure 6: Performance comparison between the best RAG method and the best prompt-only method for (a) short context lengths ($4k - 16k$) and (b) long context lengths ($\geq 32k$).

2) Effect of distractor type. We compare synthetic, natural, and random noise distractors (Section C.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.

3) Effect of temperature sampling. We assess the impact of decoding strategy by varying

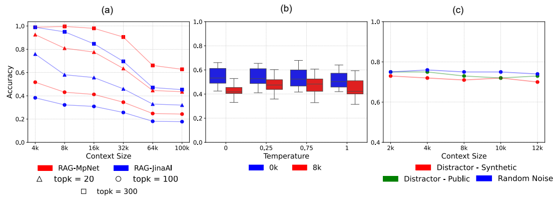


Figure 7: Ablation studies: (a) Performance of two different RAG methods with different numbers of retrieved documents. (b) Effect of temperature scaling at 0k and 8k context sizes. (c) Performance across different context sizes by virtue of the type of distractors used to increase the context size.

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 F) showing convergence beyond 250 samples.

4) Effect of model choice. We perform a model ablation using three open-weight LLMs (Llama-3.1-8B-Instruct, Qwen2-7B-Instruct, and Mistral-7B-Instruct-v0.2) under the best-performing prompt variant (CoT) from Table 1. Owing to computational constraints, we evaluate on three representative languages: English (the highest-performing language overall) and Chinese and Hindi (among the lowest-performing). All other evaluation settings are held constant, and results are summarised in Table 9. We observe similar qualitative trends across all models. First, accuracy in the no-distractor baseline remains far from saturation, affirming that MLRBench tasks are non-trivial and cannot be solved through shallow pattern matching. Second, performance degrades monotonically with increasing context length, independent of the underlying model. Although Qwen2 and Mistral achieve slightly higher absolute accuracy than Llama-3.1 in some conditions, the relative ordering across context lengths and languages is preserved.

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., 2022a), 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, these tests primarily assess *retrieval* rather than *reasoning* over the long context (Vodrahalli et al., 2024; Goldman et al., 2024). Recent work has begun to explore multilingual long-context evaluation, introducing benchmarks that combine retrieval and reasoning tasks across multiple languages (Agrawal et al., 2024; Nezhad and Agrawal, 2025; Kim et al., 2025). In contrast, our study targets a complementary axis of long-context evaluation: narrative, world-state, and symbolic reasoning. We do this by extending bAbI-style tasks to multilingual settings, framed in a SQuAD-style question–answering format (Weston et al., 2015; 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 drop in accuracy across categories such as epistemic reasoning, suggesting 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 deploying these LLMs in global, high-stakes applications.

7 Limitations

Recent API-based models such as GPT-4 (OpenAI et al., 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. Furthermore, `MLRBench` uses bAbI-style synthetic reasoning tasks and therefore evaluates models' ability to perform basic logical reasoning (e.g., deduction, entity tracking, aggregation) under long, noisy, multilingual contexts. While such tasks are useful for controlled analysis, we do not claim that performance on `MLRBench` directly correlates with all forms of real-world long-context reasoning, which may involve richer semantics, domain knowledge, and open-ended goals. Establishing such correlations remains an important direction for future work. Lastly, an interesting extension of `MLRBench` would be cross-lingual evaluation, in which the question, passage, or background texts are in different languages. Although `MLRBench` supports such experiments, this was outside the scope of this study.

8 Ethics Statement

`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.

9 Acknowledgement

We extend our gratitude to the central HPC facility (Padum) at IIT Delhi for providing computing resources. We also sincerely thank the Anusandhan National Research Foundation (CRG/2023/001351) for financial support. Tanmoy Chakraborty acknowledges the support of the Google GCP Grant and the Rajiv Khemani Young Faculty Chair Professorship in Artificial Intelligence.

References

Ameeta Agrawal, Andy Dang, Sina Bagheri Nezhad, Rhitabrat Pokharel, and Russell Scheinberg. 2024. Evaluating multilingual long-context models for retrieval and reasoning. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 216–231, Miami, Florida, USA. Association for Computational Linguistics.

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.

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2024. L-eval: Instituting standardized evaluation for long context language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14388–14411, Bangkok, Thailand. Association for Computational Linguistics.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. Accessed: April 10, 2025.

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 Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Pengfei Cao, Yuheng Chen, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. 2024. One mind, many tongues: A deep dive into language-agnostic knowledge neurons in large language models.

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. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16576–16586, Miami, Florida, USA. Association for Computational Linguistics.

Aaron Grattafiori and Abhimanyu Dubey et al. 2024. The llama 3 herd of models.

Amey Hengle, Prasoon Bajpai, Soham Dan, and Tanmoy Chakraborty. 2025. Multilingual needle in a haystack: Investigating long-context behavior of multilingual large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5165–5180, Albuquerque, New Mexico. Association for Computational Linguistics.

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

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. 2024. 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.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. 2025. Nolima: Long-context evaluation beyond literal matching.

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: April 10, 2025.

Sina Bagheri Nezhad and Ameeta Agrawal. 2025. Enhancing large language models with neurosymbolic reasoning for multilingual tasks.

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

OpenAI, Josh Achiam, and Steven Adler et al. 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.

Qwen. 2024. Qwen2-7b-instruct. `https://huggingface.co/Qwen/Qwen2-7B-Instruct`. Accessed: 2025-02-04.

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. 2022a. Language models are multilingual chain-of-thought reasoners.

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. 2022b. 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, Subhabrata Dutta, Manish Borthakur, and Tanmoy Chakraborty. 2023. Multilingual LLMs are better cross-lingual in-context learners with alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6292–6307, Toronto, Canada. Association for Computational Linguistics.

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.

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

A Background

Multilingual reasoning. The world is inherently multilingual, and for LLMs to be truly deployable in global-facing applications, they must be capable of thinking, understanding, reasoning, and responding across a broad spectrum of languages (Shi et al., 2022b; Zhao et al., 2024; Cao et al., 2024). In recent years, there has been steady progress in enhancing LLMs' multilingual reasoning abilities, enabled by advancements in in-context prompting (Tanwar et al., 2023), multi-step instruction following (Huang et al., 2023), and pre-translation methods (Ahuja et al., 2023). A key driver of this progress has been the development of reliable benchmarks like MGSM (Shi et al., 2022a), XCOPA (Ponti et al., 2020), and XL-WiC (Raganato et al., 2020). These allow researchers to evaluate various aspects of reasoning, such as arithmetic reasoning, commonsense understanding, and semantic inference. However, current work in multilingual reasoning focuses on short, isolated prompts and does not evaluate LLMs in extended, multilingual contexts. Our work builds on this line of inquiry by investigating how LLMs behave under extended multilingual contexts, a setting which is an increasingly common scenario in real-world applications where the input can span across multiple sources, dialogues, or documents (Lee et al., 2022; Rozière et al., 2024; Shah et al., 2024; Modarressi et al., 2025).

Long-context evaluation. Currently, long-context evaluation is dominated by the popular *Needle-in-a-Haystack (NIAH)* tests (Kamradt, 2023), where an LLM is tasked with finding one or more pieces of information (the *"needle"*) from long, irrelevant background texts (the *"haystack"*). Existing multilingual long-context evaluation benchmarks have extended the NIAH test to multiple languages (Agrawal et al., 2024; Hengle et al., 2025; Kim et al., 2025; Huang et al., 2025). While these are suitable for evaluating surface-level *retrieval* or *recall*, they are not sufficient to evaluate multilingual reasoning under long-context settings (Goldman et al., 2024). Subsequently, a growing body of research has started to look beyond the retrieval-focused NIAH tests for long-context evaluation, albeit limited to monolingual English settings (Kuratov et al., 2024; Vodrahalli et al., 2024; Karpinska et al., 2024). For instance, Vodrahalli et al. (2024) proposed the Latent Structure Queries (LSQ) framework, which necessitates the LLM to understand the *latent information* within extended contexts. More recently, Kuratov et al. (2024) proposed multiple synthetic reasoning tasks by extending the bAbI dataset to long-context settings. We take inspiration from them while curating `MLRBench`.

Synthetic and realistic benchmarks. Creating realistic benchmarks for long-context evaluation is often challenging due to factors like increased complexity, limited resources, and the high cost of human annotation (Hsieh et al., 2024). As a result, recent research has increasingly turned to synthetically generated datasets as proxies for real-world performance (Vodrahalli et al., 2024). Because synthetic datasets are easier to construct and evaluate, most existing long-context benchmarks are synthetic, commonly following the NIAH framework (Kamradt, 2023; Hsieh et al., 2024; An et al., 2024; Bai et al., 2024; Kim et al., 2025). In contrast, realistic benchmarks typically center on tasks that reflect practical applications, such as summarization (Kim et al., 2024) and question answering (Kuratov et al., 2024; Shaham et al., 2023; Hengle et al., 2025). While synthetic benchmarks offer greater control and flexibility, realistic benchmarks are more reliable for assessing real-world performance. `MLRBench` brings together the strengths of both approaches — it builds on the synthetic bAbI dataset while presenting tasks in the SQuAD format (Rajpurkar et al., 2016), aligning with the structure and demands of real-world applications.

B 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.

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.

We formally define the input passage as:

$$IP = \{N_i \mid i = 1, 2, \ldots, n\}, \tag{1}$$

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

$$N_1 \prec N_2 \prec \cdots \prec N_n, \tag{2}$$

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:

$$D = \{D_j \mid j = 1, 2, \ldots, m\}, \tag{3}$$

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

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

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.

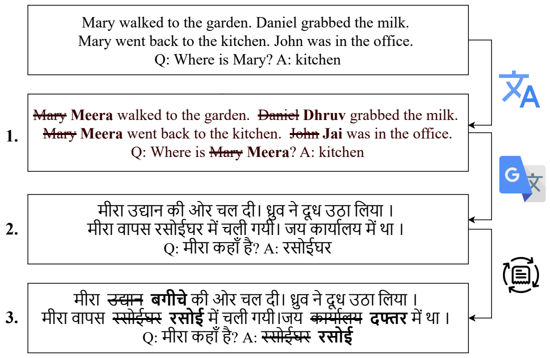
C Dataset

C.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., 2025). 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.

C.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 benchmarks such as MLQA (Lewis et al., 2020) and MLNeedle (Hengle et al., 2025). 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[8] to translate the passage, question, and answer from English to all selected languages [9]. 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.



Mary walked to the garden. Daniel grabbed the milk.
Mary went back to the kitchen. John was in the office.
Q: Where is Mary? A: kitchen

1. ~~Mary~~ **Meera** walked to the garden. ~~Daniel~~ **Dhruv** grabbed the milk.
~~Mary~~ **Meera** went back to the kitchen. ~~John~~ **Jai** was in the office.
Q: Where is ~~Mary~~ **Meera**? A: kitchen

2. मीरा उद्यान की ओर चल दी। ध्रुव ने दूध उठा लिया ।
मीरा वापस रसोईघर में चली गयी। जय कार्यालय में था ।
Q: मीरा कहाँ है? A: रसोईघर

3. मीरा ~~उद्यान~~ **बगीचे** की ओर चल दी। ध्रुव ने दूध उठा लिया ।
मीरा वापस ~~रसोईघर~~ **रसोई** में चली गयी।जय ~~कार्यालय~~ **दफ्तर** में था ।
Q: मीरा कहाँ है? A: ~~रसोईघर~~ **रसोई**

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.

C.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 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 shown in Table 9, bAbI comprises discrete, atomic factual statements that follow a 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-

[8] Google Translate API

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

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.

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 [10], which is one of the most high-performing and widely used machine translation tools [11]. We assess the quality of the translated data in MLRBench by conducting a translation quality estimation (QE) following prior work (Takahashi et al., 2020) [12]. 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.

C.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 LLMs, (ii) *natural distractors* sampled from a public text corpus, and (iii) *random noise* to serve as a control condition. Below, we briefly describe each of them.

Synthetic distractors Recent work, such as Hsieh et al. (2024), has explored the use of synthetically generated distractors to control task complexity and make evaluation *a posteriori identifiable* [13]. Building on this idea, we construct our own corpus of in-domain synthetic distractor texts by prompting GPT-4 (OpenAI et al., 2024), Llama3.1-Instruct (Grattafiori and et al., 2024), and Qwen2-Instruct (Qwen, 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) → action (verb) → object (or location/entity). These facts serve as seed examples for generating distractors. For each fact, we prompt the models 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 prompt 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

[10] Google Translate API

[11] Performance report of Google Translation API

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

[13] 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.

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.

D 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:
  - **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.

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.

- **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

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.

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.

E 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[14] (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.

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.

F 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.

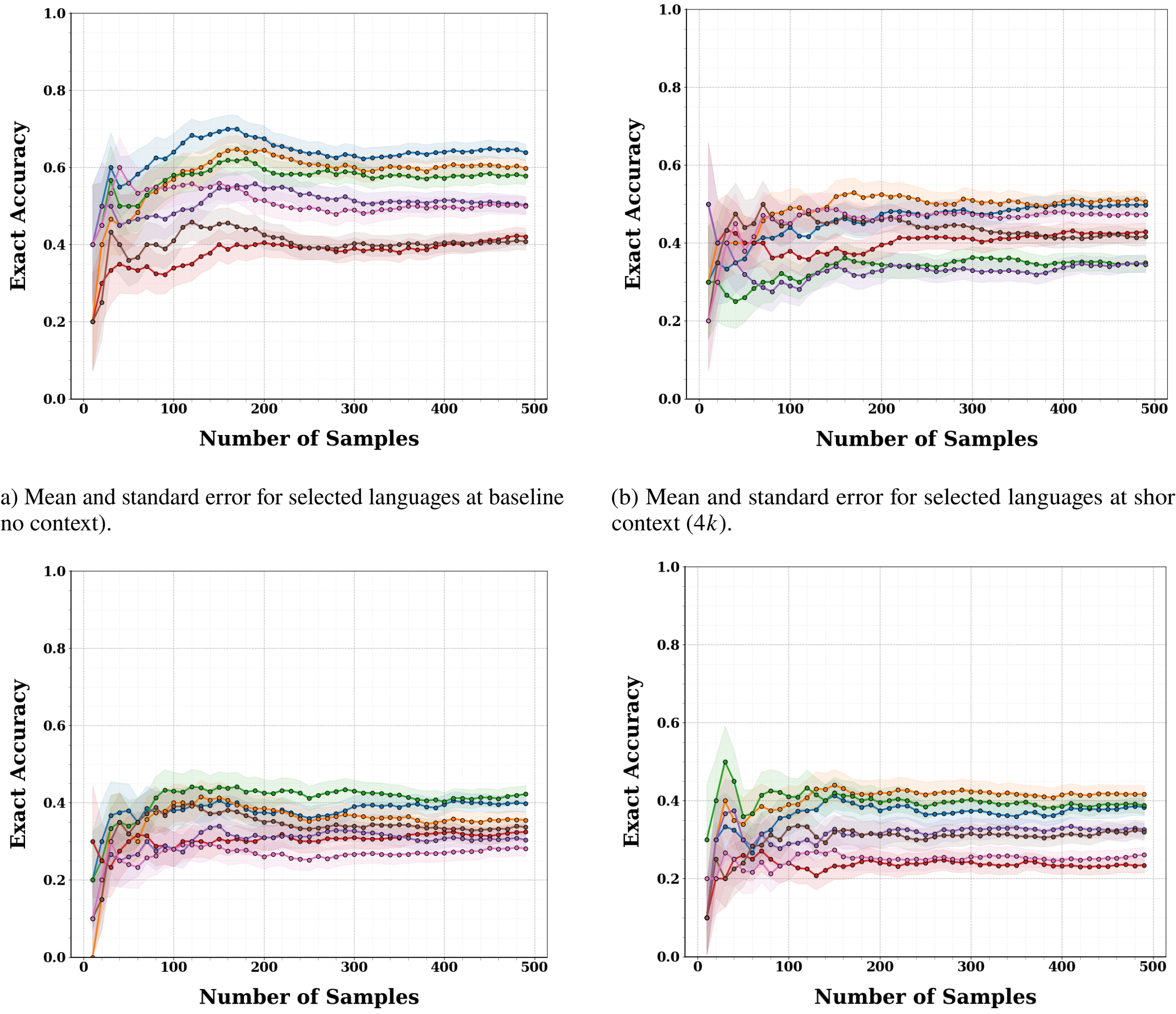
We find that across all selected languages, the exact accuracy begins to stabilize between 250 and 500 samples. Additionally, the standard error decreases significantly as the number of evaluation samples exceeds 250. This indicates that using 250 or more samples is sufficient to yield reliable and consistent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

[14] 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.



(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.

Model	**Lang**	**Baseline**	**2*k***	**8*k***	**32*k***	**128*k***
Llama-3.1-8B-Instruct	en	0.673	0.582	0.577	0.495	0.353
	zh	0.567	0.485	0.417	0.335	0.268
	hi	0.602	0.520	0.432	0.377	0.285
Qwen2-7B-Instruct	en	0.686	0.595	0.587	0.505	0.364
	zh	0.751	0.620	0.592	0.561	0.490
	hi	0.626	0.586	0.516	0.436	0.312
Mistral-7B-Instruct-v0.2	en	0.672	0.564	0.558	0.470	–
	zh	0.655	0.646	0.554	0.438	–
	hi	0.630	0.529	0.435	0.378	–

Table 9: Model-choice ablation on MLRBench using COT prompting. Results are reported for three representative languages. While absolute accuracy varies across models, all exhibit consistent degradation as context length increases.

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

```
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.

		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 11: 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.

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.

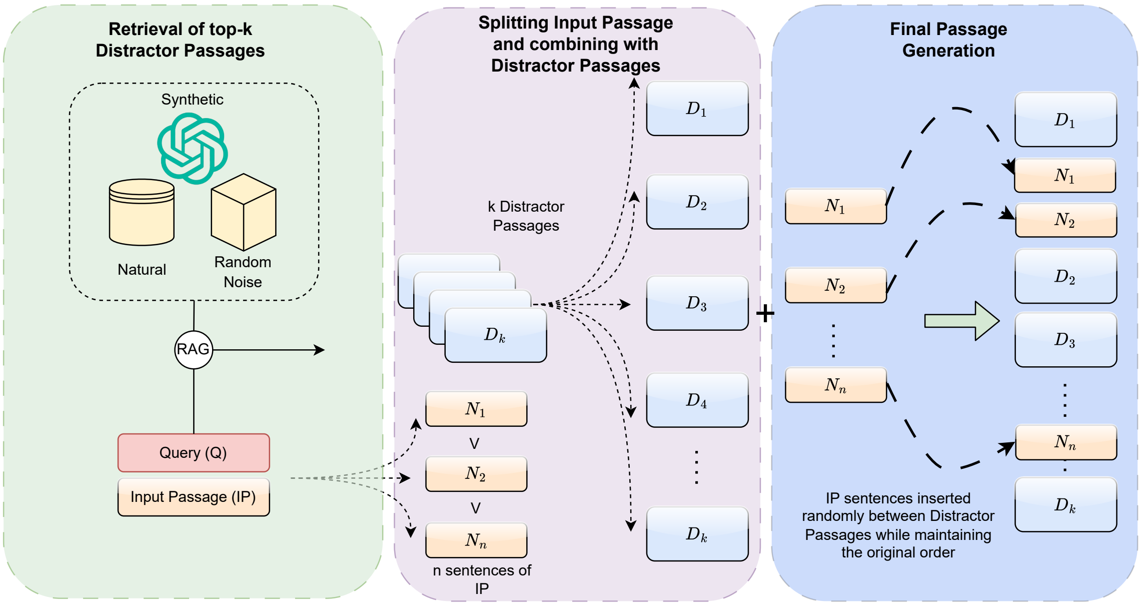


Figure 19: Overview of our pipeline for creating MLRBench. An Input Passage (IP) is split into n sentences. Given IP and the query, top-k distractor passages are retrieved from sources — (1) synthetic passages from GPT-4, (2) public datasets, and (3) random token-level noise. The independent sentences 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.