Neurosymbolic Artificial Intelligence
XX(X):1–15

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Robust Long-Context Multilingual Retrieval and Reasoning Enabled by Combined Neural and Symbolic Techniques

Sina Bagheri Nezhad and Ameeta Agrawal
Portland State University, USA

Abstract

Large language models (LLMs) are increasingly deployed for multilingual information retrieval and reasoning over very long documents, yet they often struggle with extracting dispersed facts and synthesizing robust answers across linguistic boundaries. In this work, we propose a hybrid neural-symbolic framework that integrates scalable cross-lingual retrieval with explicit symbolic reasoning. Our approach, **CROSS** (Cross-lingual Retrieval Optimized for Scalable Solutions), efficiently narrows massive multilingual contexts using multilingual embeddings, dramatically improving retrieval accuracy and mitigating the "lost-in-the-middle" problem. Building on this, we introduce **NeuroSymbolic Augmented Reasoning (NSAR)**, which prompts LLMs to extract structured facts and generate executable Python code, enabling deterministic and interpretable multi-target reasoning. We evaluate our methods on the mLongRR-V2 benchmark, spanning seven languages, 49 cross-lingual pairs, and documents up to 512,000 words. Our experiments show that compared to neural-only baselines, CROSS boosts retrieval accuracy up to 92% and NSAR reduces reasoning failures fivefold, while maintaining stable performance across languages and context sizes. These results establish a new standard for robust, scalable, and interpretable multilingual information extraction, demonstrating the promise of hybrid neural-symbolic architectures for future AI systems.

Keywords

neurosymbolic, multilingual

Introduction

Despite recent advances in large language models (LLMs), robust information retrieval and reasoning still degrades in two settings that are often studied independently: *retrieval from very long documents* and *cross-lingual information retrieval (CLIR)*. In this work we target the intersection of these two settings, where evidence may be deeply buried inside an extremely long document *and* written in a different language than the user's query.

Challenge 1: Retrieval from very long documents. When the input spans tens or hundreds of thousands of words, LLMs exhibit sharp drops in evidence-finding reliability due to attention dilution and position effects, often summarized by the "lost-in-the-middle" and "needle-in-a-haystack" phenomena (Liu et al. 2023; Xu et al. 2024). Long-context architectures (e.g., sparse/global attention variants) extend feasible context length, but do not fully eliminate mid-context retrieval failures (Beltagy et al. 2020; Zaheer et al. 2021).

Challenge 2: Cross-lingual information retrieval. CLIR focuses on mapping a query in one language to relevant evidence in another. Classical CLIR emphasized translation-based pipelines (e.g., dictionaries, query translation), while modern approaches increasingly rely on multilingual representation learning that embeds text from different languages into a shared semantic space (Nie 2010; Agrawal et al. 2024). Despite progress, CLIR remains difficult for typologically distant languages and for tasks requiring robust multi-step inference across retrieved evidence.

When the challenges interact. In cross-lingual long-context settings, these difficulties compound: the system must *both* bridge a language mismatch and reliably locate dispersed facts inside a massive context. This combined regime is underexplored at extreme *single-document* lengths and across diverse scripts, motivating our benchmark and hybrid neural-symbolic approach.

Traditional approaches to these challenges have focused primarily on architectural improvements—expanding context windows (Beltagy et al. 2020; Zaheer et al. 2021), enhancing attention mechanisms (Jiang et al. 2024), or developing specialized fine-tuning techniques (Litschko et al. 2022). While valuable, these approaches remain constrained within a purely neural paradigm that forces models to perform both retrieval and reasoning through the same undifferentiated neural processes. This monolithic architecture inevitably creates bottlenecks even when initial retrieval succeeds, leading to the "lost-in-the-middle" phenomenon and cascading errors in multi-step reasoning (Liu et al. 2023; Xu et al. 2024).

Neurosymbolic artificial intelligence offers a promising alternative by integrating the pattern recognition capabilities of neural networks with the logical rigor of symbolic systems (d'Avila Garcez and Lamb 2020). In our previous work (Nezhad and Agrawal 2025), we introduced initial

components of a neurosymbolic framework for multilingual tasks. This paper substantially extends that work, presenting **NeuroSymbolic Augmented Reasoning (NSAR)**—a comprehensive paradigm that fundamentally reconceptualizes how LLMs approach complex information extraction tasks.

The key novelties in this work compared to our previous conference paper include:

- We explicitly describe CROSS as a practical cross-lingual dense-retrieval backbone (aligned with standard RAG designs), tuned for *sentence-level indexing* and *single-document* contexts up to 512,000 words. We additionally report retrieval-stage performance separately from generation to clarify what CROSS contributes versus what reasoning contributes.
- Expansion to fully bidirectional cross-lingual evaluation across 49 language pairs, enabling queries and contexts in any combination of the seven languages, rather than fixed English queries.
- Development of the mLongRR-V2 benchmark, which enhances linguistic diversity, introduces 1-needle retrieval tasks alongside 3-needle reasoning, and supports more rigorous testing of cross-lingual capabilities.
- Incorporation of both 1-needle and 3-needle evaluation protocols for a more comprehensive assessment of retrieval and reasoning performance.
- Detailed empirical analysis demonstrating up to 92% retrieval accuracy in single-needle tasks and a fivefold reduction in reasoning failures, with new insights into error types and cross-script stability.

Rather than treating retrieval and reasoning as inseparable neural processes, NSAR explicitly decomposes them, introducing symbolic components precisely where neural approaches typically falter. We implement this paradigm through two complementary systems:

1. **CROSS** (Cross-lingual Retrieval Optimized for Scalable Solutions): A multilingual RAG framework that embeds and narrows extensive documents to the most relevant segments, effectively addressing the retrieval challenge.
2. **NSAR**: A neurosymbolic reasoning layer that prompts LLMs to (a) extract structured relations from retrieved text, (b) generate executable Python code implementing formal reasoning rules, and (c) execute this code to produce verifiable answers.

This hybrid architecture transforms complex reasoning from opaque neural computations into explicit, auditable symbolic operations guided by neural components. By introducing an intermediate symbolic representation layer between retrieval and final inference, NSAR enables compositional reasoning that remains robust even across linguistic boundaries—a capability that purely neural approaches have struggled to achieve consistently.

Our experiments on the mLongRR-V2 benchmark—spanning seven languages, 49 language pairs, and documents up to 512,000 tokens—provide compelling evidence for this paradigm shift. While baseline GPT-4o-mini and Llama 3.2 models achieved only 37% and 47% retrieval accuracy respectively across diverse language combinations, our CROSS-enhanced approach dramatically improved performance to 87% and 92%. Most strikingly, in the challenging 3-needles reasoning task, traditional approaches exhibited LLM failure rates of 52.2% for GPT-4o-mini and 22.9% for Llama 3.2. NSAR reduced these to just 8.9% and 6.2% respectively—a remarkable fivefold improvement in reasoning capability.

The neurosymbolic approach demonstrated exceptional cross-linguistic stability, with performance variations under 5% across diverse scripts including Latin, Cyrillic, Devanagari, and Arabic. Perhaps most significantly, while purely neural reasoning methods showed declining performance with increasing context complexity, NSAR maintained robust performance regardless of input volume—demonstrating its unique ability to handle complexity through explicit symbolic representation. To our knowledge, this is the first work exploring the promise of neurosymbolic methods in improving multilingual performance over long contexts.

The key findings of this paper include:

- CROSS improves retrieval accuracy from 37-47% in baselines to 87-92% across multilingual settings.
- NSAR reduces reasoning failure rates fivefold in multi-needle tasks.
- The framework maintains stable performance across context lengths up to 512k words and diverse language pairs.
- Neurosymbolic methods show superior robustness to the "lost-in-the-middle" problem and cross-lingual challenges.
- Optimal sentence cap sizes vary by task complexity, with trade-offs between retrieval completeness and reasoning accuracy.

For future research and design, we recommend:

- Extending symbolic representations to richer formalisms like first-order logic or constraint solvers.
- Evaluating on additional low-resource languages and real-world multilingual corpora.
- Integrating more advanced embedding models and exploring adaptive sentence cap selection.
- Developing automated verification mechanisms for generated code in production systems.
- Investigating hybrid architectures for other multimodal long-context tasks beyond text.

This work extends beyond incremental improvements to existing techniques, offering instead a fundamental reconceptualization of how language models can approach complex reasoning tasks. By establishing a principled integration of neural flexibility with symbolic rigor, NSAR

opens new avenues for auditable AI systems in mission-critical domains like healthcare, legal analysis, and scientific research, where reliable cross-lingual understanding of extensive documentation is essential. More broadly, our results suggest that the future of AI for complex language tasks likely lies not in ever-larger neural architectures alone, but in hybrid systems that strategically combine neural and symbolic components to overcome their respective limitations.

Related Works

Retrieval and reasoning over long contexts

A growing body of work documents that LLM accuracy degrades when evidence is embedded deep within long inputs, commonly referred to as the "lost-in-the-middle" effect (Liu et al. 2023; Xu et al. 2024). Long-context architectures such as Longformer and BigBird extend feasible context lengths via sparse or structured attention, but do not fully eliminate mid-context retrieval failures, especially as contexts grow and the evidence becomes sparse (Beltagy et al. 2020; Zaheer et al. 2021). Retrieval-augmented generation (RAG) has therefore become a standard strategy for long-document QA: rather than forcing the generator to attend over the entire input, a retriever first selects a small subset of relevant text and only that subset is provided to the LLM for answer generation (Lewis et al. 2020). Long-context variants such as LongRAG further emphasize retrieval as a mechanism for scalability when document length exceeds practical LLM context limits (Jiang et al. 2024).

Cross-lingual information retrieval

Cross-lingual information retrieval (CLIR) has a long history, traditionally emphasizing translation-based pipelines (dictionary-based translation, query translation, and probabilistic structured queries) (Nie 2010; Yang et al. 2024). More recent CLIR systems increasingly rely on multilingual encoders and dense retrieval in shared embedding spaces, enabling a query in Language A to retrieve evidence in Language B without explicit translation. In multilingual and cross-lingual evaluation, datasets such as mLongRR and BordIRlines highlight that performance remains uneven across language families and scripts, and further degrades as contexts grow longer and more heterogeneous (Agrawal et al. 2024; Li et al. 2024).

RAG pipelines and retrieval granularity

While the retriever–generator paradigm is widely adopted, a key design axis is *retrieval granularity*: whether the corpus is indexed by documents, passages, sentences, or even finer units. Recent surveys and empirical studies show that granularity strongly affects both retrieval recall and downstream QA quality (Gao et al. 2024; Wang et al. 2024). For example, Dense X Retrieval systematically evaluates passage-, sentence-, and proposition-level dense retrieval, demonstrating that smaller retrieval units can improve evidence concentration under a fixed LLM budget (Chen et al. 2024b). These findings contextualize our choice to use *sentence-level* indexing in CROSS: rather than claiming sentence-level retrieval as a new idea, our contribution is to operationalize it in a cross-lingual *single-document* long-context setting (up to 512,000 words) and to separate retrieval-stage performance from reasoning-stage performance.

Neurosymbolic and code-based reasoning

Beyond retrieval, multi-hop and compositional reasoning over retrieved evidence remains brittle when performed purely within neural generation, often leading to hallucinations or inconsistent aggregation. Neurosymbolic approaches aim to combine neural language understanding with explicit symbolic computation, improving interpretability and enabling deterministic verification (d'Avila Garcez and Lamb 2020). Recent work increasingly uses code generation as a practical symbolic layer for verifiable reasoning, where intermediate structured facts are extracted and then executed via a programming language runtime. Our NSAR module builds on this line of work by extracting symbolic facts and generating executable Python code to perform multi-target reasoning deterministically (Nezhad and Agrawal 2025).

Summary. Prior work has separately studied (i) long-context retrieval and (ii) CLIR, and has also explored RAG design choices such as retrieval granularity. However, robust evaluation and methods for the *combined* cross-lingual long-context regime—particularly at extreme *single-document* lengths and for multi-target reasoning—remain limited. Our work addresses this gap with a scalable cross-lingual retrieval backbone (CROSS), a neurosymbolic reasoning layer (NSAR), and a benchmark (mLongRR-V2) that stresses both factors simultaneously.

Methodology

Our proposed architecture is a sequential pipeline designed to bridge the gap between retrieval and reasoning. The workflow proceeds as follows: First, the **CROSS** module tokenizes and embeds the multilingual document (the "haystack"), retrieving only the most semantically relevant sentences. Second, these sentences are passed to the **NSAR** module, which prompts the LLM to extract symbolic facts and generate executable Python code. Finally, the code is executed to derive the deterministic answer.

CROSS Framework

At a high level, CROSS follows the standard dense-retrieval pattern used in many RAG systems: segment text, embed segments and query in a shared space, and retrieve the top-k most similar units (Lewis et al. 2020; Gao et al. 2024). We therefore do *not* claim a new retrieval objective. Instead, CROSS is a pragmatic backbone tailored to the specific regime we study: (i) *single-document* "haystacks" up to 512,000 words (often far beyond practical LLM context limits), (ii) *cross-lingual* query–context mismatch, and (iii) *fine-grained* sentence-level indexing to reduce distractors and make retrieval-stage evaluation explicit. Recent work emphasizes that retrieval granularity is a key RAG design factor (Wang et al. 2024; Chen et al. 2024b); CROSS adopts sentence-level units as the most faithful granularity

for needle-style tasks while keeping the LLM budget fixed via a sentence cap.

The CROSS framework (Cross-lingual Retrieval Optimized for Scalable Solutions) efficiently extracts "needles" of relevant information from extensive, multilingual "haystacks." Using a two-phase approach, CROSS improves retrieval accuracy, ensures cost efficiency, and overcomes the limitations of current models in handling long, cross-lingual contexts.

While CROSS leverages the fundamental architecture of standard dense retrieval (RAG), it distinguishes itself through its specific optimization for massive multilingual contexts (up to 512k words). Unlike typical document-level or chunk-level retrieval which may lose precision in cross-lingual settings, CROSS operates at a strict *sentence-level granularity*. This design choice, combined with the state-of-the-art `BGE-M3` embedding model, allows us to bypass the "translation gap" often found in traditional CLIR systems, serving as a necessary and highly optimized backbone for the subsequent neurosymbolic reasoning.

Two-Phase Retrieval Mechanism CROSS employs a Retrieval-Augmented Generation (RAG) framework that leverages a two-phase retrieval process to enhance precision while minimizing computational overhead.

Phase 1: Tokenization and Embedding The context—potentially comprising hundreds of thousands of words in multiple languages—is segmented into sentences using the Punkt tokenizer (Kiss and Strunk 2006). Each sentence is then embedded using the multilingual "BGE-M3" model (Chen et al. 2024a), which effectively captures semantic nuances across languages. Although operating at the sentence level might seem computationally demanding, particularly if one considers finer granularity, our cost analysis demonstrates that the expense of embedding and retrieval is negligible relative to the cost of LLM processing.

Phase 2: Candidate Selection and Model Input Within this RAG framework, CROSS calculates the semantic distance between each sentence embedding and the query, selecting the top k most relevant sentences based on a tunable hyperparameter. In our experiments, we evaluated k values of 3, 5, 10, 20, and 50. These selected sentences are then passed as concise, contextually rich inputs to the language model (e.g., GPT-4o-mini or Llama 3.2 90b) for final answer extraction. This design ensures that, despite the additional embedding and retrieval steps, the overall token processing by the LLM is drastically reduced, preserving both accuracy and efficiency.

Embedding Model: BGE-M3 for Cross-Lingual Compatibility The BGE-M3 embedding model, with a 1024-dimensional embedding size, is key to CROSS's multilingual capabilities. It embeds sentences from different languages into a shared vector space, enabling CROSS to assess sentence relevance across languages and significantly boosting cross-lingual accuracy. By capturing both syntactic and semantic features, BGE-M3 ensures robustness across diverse linguistic families, supporting accurate retrieval in languages like Persian, Hindi, Russian, and Arabic.

Efficiency and Model Independence CROSS is model-independent, enhancing retrieval accuracy with any language model used in Phase 2. Tested with GPT-4o-mini and Llama 3.2, it dynamically adjusts the number of retrieved sentences, ensuring consistent, cost-effective performance. By focusing on the most relevant context segments, CROSS avoids attention drop-offs in long texts and maximizes precision. Its fixed input length makes it scalable, effectively handling document lengths far beyond the native context limits of most language models.

NeuroSymbolic Augmented Reasoning (NSAR)

Purely neural approaches to long-context question answering often struggle with reliability, interpretability, and logically integrating multiple pieces of information. While recent prompting techniques (e.g., Chain-of-Thought, ReAct, and Self-Reflection) have improved reasoning, they remain reliant on implicit neural processes, leaving little room for explicit verification or modular correction. To address these limitations, we introduce the NeuroSymbolic Augmented Reasoning (NSAR) component which integrates structured symbolic representations within the neural architecture by extracting symbolic facts and generating executable Python code for reasoning. This approach bridges the gap between the *flexibility and fluency* of LLMs and the *interpretability and rigor* of symbolic methods. Neural systems excel at language understanding and generation, but often fail in complex scenarios requiring multiple reasoning steps, such as comparing multiple facts, deducing the "largest" or "smallest" value, or verifying constraints across scattered pieces of information. Symbolic methods, by contrast, excel in structured reasoning but can be brittle when parsing unstructured text. By coupling a language model with a symbolic layer, NSAR enhances interpretability by providing an explicit record of extracted facts and logical steps, enabling users to audit and verify the model's reasoning. Additionally, it improves reliability by reducing errors in compositional tasks by systematically comparing and fusing pieces of information through symbolic code execution.

We design an *NSAR prompt* to append to the retrieved context before querying the LLM. This prompt guides the reasoning process through three distinct stages:

1. Symbolic Fact Extraction First, the model is instructed to identify all relevant facts in the provided context and represent them in a structured, symbolic format. For instance, if the context contains lines such as *"The special magic Cairo number is: 1234567"* and *"The special magic Mumbai number is: 9999999"*, the model generates:

```
FACT("Cairo", "special_magic_number",
    1234567)
FACT("Mumbai", "special_magic_number",
    9999999)
```

2. Python Code Generation Next, the model is prompted to produce concise, executable Python code that uses the extracted symbolic facts to answer the question. Instead of implicitly inferring logic through text, the Python code can contain explicit comparisons (`>`, `<`, `==`), data structures (lists, dictionaries), or domain-specific libraries. In the case of identifying the *largest special magic number*, this code might look like:

```
numbers = [1234567, 9999999]
answer = max(numbers)
```

The logic here can be arbitrarily extended to handle more complex reasoning steps (e.g., filtering facts, applying constraints, or computing aggregates). Once the LLM generates the Python code, it is executed in a controlled environment.

3. Final Answer Extraction Finally, the answer is determined by executing the generated Python code. This guarantees a concise, verified response and prevents any contradictory or incoherent rationales that might arise from purely text-based reasoning. In other words, while the LLM might propose a final textual answer, the *actual* answer delivered to the user is the deterministic output of the code execution.

As such, NSAR prompt structure ensures that the language model provides an interpretable chain of reasoning, resulting in a Python snippet that can be independently executed and verified. The template of NSAR prompt is shown below:

NSAR Prompt Template

You are a helpful assistant that employs a neurosymbolic method. Given the following context and question, please follow these steps:
1. Extract all relevant facts from the context and represent them as symbolic facts using the format `FACT(entity, attribute, value)`.
2. Generate executable Python code that uses the extracted symbolic facts to compute the final answer.
3. Finally, output only the final answer.
#CONTEXT
{text}
#ENDCONTEXT
#QUESTION
What is the largest special magic number?

In this work, "neurosymbolic" denotes the hybrid of explicit FACT-triple extraction with deterministic code execution. While our current triples support only simple attribute logic, extending to richer formalisms (e.g. first-order rules or constraint solvers) would more fully realize the neurosymbolic ideal.

Dataset: mLongRR-V2

The mLongRR-V2 dataset builds on the original mLongRR by Agrawal et al., which evaluated multilingual long-context models on retrieval tasks using five Latin script languages. However, the original mLongRR was limited to a maximum context length of 64,000 tokens and lacked diversity in script types (Agrawal et al. 2024). mLongRR-V2 addresses these limitations by extending the context length to 512,000 words and expanding the language set to include seven languages: English, Vietnamese, Swahili, Persian, Russian, Hindi, and Arabic. This expansion not only enhances linguistic diversity by incorporating non-Latin scripts such as Cyrillic, Devanagari, and Arabic, but also introduces a crucial cross-lingual dimension, allowing for more robust evaluations of retrieval models in multilingual and cross-script scenarios.

The cross-lingual aspect of mLongRR-V2 is designed to rigorously test retrieval models across different language combinations. In this dataset, the haystack is always monolingual, meaning that all context within a given test case is written in a single language. However, the needle (target sentence) is embedded within the haystack in the same language as the haystack itself. The cross-lingual challenge arises from the fact that the query is presented in a different language from the haystack, requiring the model to bridge linguistic differences to retrieve the correct information.

Needle Structure In this task, the model's objective is to locate and extract information from a single target sentence hidden within the context. We adopt the same needle pattern as used in previous studies (Agrawal et al. 2024; Team 2024; Anthropic 2024), which takes the form: **"The special magic {city} number is: {number}"**. Here, {city} is randomly chosen from a list of 23 unique cities worldwide, and {number} is a randomly generated 7-digit number. The list of cities was translated into all the dataset languages to ensure accuracy and linguistic consistency.

Cross-Lingual Language Pairs and Needle Positioning To provide a rigorous assessment, mLongRR-V2 includes 49 cross-lingual language pairs, pairing each language in the prompt and query with every other language in the context. This setup simulates real-world scenarios where queries and contexts are often in different languages, adding complexity to the retrieval task.

Building on the original mLongRR, mLongRR-V2 positions the target information (the "needle") at five distinct locations within the context: the beginning (0%), near the start (25%), in the middle (50%), near the end (75%), and at the end (100%). This systematic positioning tests model robustness across varying depths, addressing challenges like the "lost in the middle" problem, where retrieval accuracy typically drops for mid-context information.

To test the reasoning capability of CROSS, we introduced a **3-needles setup**, where three needles are placed randomly within the context. The task requires the model to identify and reason about the needles to answer a query related to the largest one, further evaluating CROSS's ability to process complex multilingual scenarios.

Context Length mLongRR-V2 significantly extends the context length to a maximum of 512,000 words, enabling the evaluation of models on much longer texts compared to the original mLongRR dataset. The dataset is carefully designed to test models across varying context lengths, consisting of 2k, 8k, 16k, 32k, 64k, 128k, 256k, and 512k words. This range allows for a comprehensive assessment of a model's scalability and performance under diverse conditions.

Evaluation Protocol

To comprehensively evaluate the effectiveness of CROSS, we designed two distinct evaluation tasks: the 1-needle test and the 3-needles test. These tests assess retrieval and reasoning capabilities across diverse cross-lingual scenarios.

1-Needle Test The 1-needle test evaluates the model's ability to retrieve specific information embedded within extensive multilingual contexts. In this task, a single "needle" (a target piece of information) is placed in the context at one of five predefined positions: the beginning (0%), near

the start (25%), in the middle (50%), near the end (75%), or at the end (100%).

The prompt asks the model: “What is the special magic number?”, written in different languages to assess cross-lingual retrieval. The model must locate the relevant information and provide the correct answer, ensuring retrieval accuracy across both language pairs and varying context positions.

3-Needles Test The 3-needles test evaluates the model’s reasoning capability in addition to retrieval. In this setup, three needles are randomly distributed throughout the context. The model is prompted to identify and reason about the needles to answer the query: “What is the largest special magic number?”

This task challenges the model to not only locate multiple relevant pieces of information but also reason over them to produce the correct answer. The random placement of the needles tests robustness against varying context complexity. Each case was tested three times to account for the variability introduced by the random distribution of needles, ensuring a more reliable evaluation of the model’s reasoning capabilities.

Metric: Retrieval Accuracy We use retrieval accuracy as the primary metric to evaluate model performance in both tasks. Accuracy is defined as the percentage of test cases where the model correctly identifies and retrieves the required information. In the 1-needle test, this means correctly locating the "special magic number." In the 3-needles test, accuracy measures the model’s ability to reason and correctly identify the largest "special magic number."

Metric: Retrieval-stage recall (Embedding Recall@k). To isolate retriever performance from the generator, we additionally report *Embedding Recall@k*, defined as the fraction of instances where the gold needle sentence is present in the top-k retrieved sentences *before* the LLM generates an answer. This mirrors standard retriever evaluation in RAG pipelines and is directly computable from our embedding failure analysis (Embedding Recall@k = 1 – Embedding Failure).

Prompts and Queries The prompts and queries used in both the 1-needle and 3-needles tests are carefully crafted to ensure clarity and fairness across languages.

Results and Analysis

This section presents the experimental results of our approach, which combines CROSS’s retrieval framework with Llama 3.2 and GPT-4o-mini as the underlying language model. We analyze its performance across retrieval accuracy, robustness to context length, needle position sensitivity, cross-lingual consistency, and cost efficiency, comparing it to the baseline performance of LLMs alone.

Initial LLM Performance Evaluation without "Contexts"

Before integrating CROSS, we tested both GPT-4o-mini and Llama 3.2 90b independently to verify their ability to understand the prompts and correctly retrieve or reason answers without any provided context. This evaluation was conducted for both the 1-needle (retrieval) and 3-needles (reasoning) scenarios. Each model was tested 10 independent times using prompts and needles alone, without contextual interference. Both models demonstrated flawless performance, successfully identifying the correct answers in all tests. These results confirm that the prompts are clear and fully understandable to the LLMs, establishing a solid foundation for evaluating the impact of CROSS in more complex retrieval scenarios.

Retrieval Accuracy

CROSS achieved significant improvements in retrieval accuracy across all tested languages and language pairs when paired with both GPT-4o-mini and Llama 3.2 90b. Compared to using the language models alone, the CROSS-enhanced approach consistently retrieved the target sentence with higher exact match accuracy, especially in long contexts and complex cross-lingual pairs.

Baseline Definition: In all comparisons, the "Baseline" (e.g., "GPT-4o-mini only") refers to the standard long-context capability of the model, where the *entire* haystack text is provided in the prompt up to the model’s context window limit (128k tokens). This serves as a strong baseline, representing the model’s native ability to handle long contexts without external retrieval.

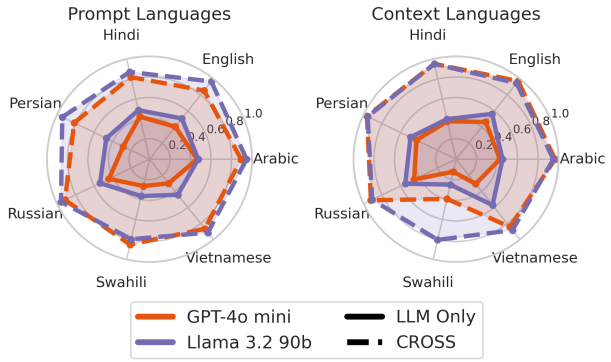
On average, across all 49 language combinations, CROSS with GPT-4o-mini achieved a retrieval accuracy of **87%**, significantly outperforming the baseline GPT-4o-mini, which achieved only **37%**. Similarly, CROSS with Llama 3.2 achieved a remarkable improvement, with accuracy increasing from **47%** for Llama 3.2 alone to **92%** when enhanced with CROSS.

Furthermore, for contexts under 64k words—the length supported by both models—CROSS-enhanced GPT-4o-mini maintained a retrieval accuracy of **88%**, compared to **59%** for GPT-4o-mini alone. Llama 3.2 also showed improvement under 64k words, with accuracy increasing from **75%** for the baseline model to **95%** with CROSS. These substantial improvements across both context lengths and models demonstrate CROSS’s effectiveness in preserving high retrieval accuracy.

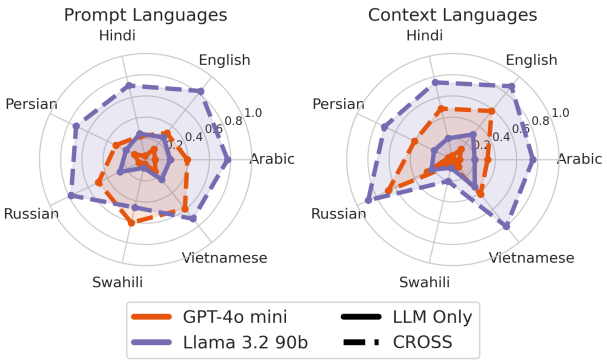
As illustrated in the radar graphs in Figures 1a and 1b, CROSS enhances retrieval and reasoning performance across all prompt and context languages, indicating the robustness of this approach in varied multilingual scenarios. These results highlight the effectiveness of the CROSS framework in maintaining high accuracy across diverse linguistic contexts when paired with both GPT-4o-mini and Llama 3.2.

Context Length Robustness

A key strength of CROSS is its robust performance across varied context lengths. Without CROSS, both models exhibit a notable decline in retrieval accuracy as context length increases, with sharp reductions observed beyond 64k words. In contrast, the CROSS-enhanced approach maintains consistent accuracy across context lengths up to 512k words for both models, showing only minimal reduction (Figure 2a). By narrowing the input to a fixed set of top-relevant sentences, CROSS effectively mitigates the typical accuracy



(a) Comparison of Average Accuracy for Each Prompt and Context Language in the 1-Needle Test

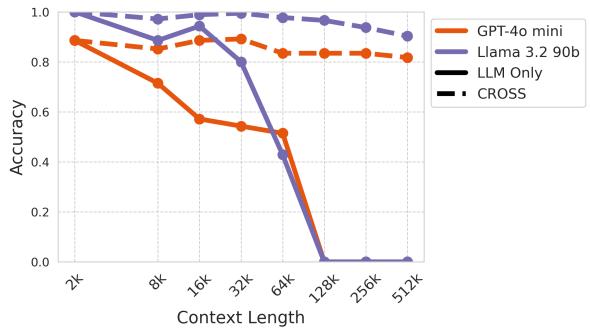


(b) Comparison of Average Accuracy for Each Prompt and Context Language in the 3-Needles Test

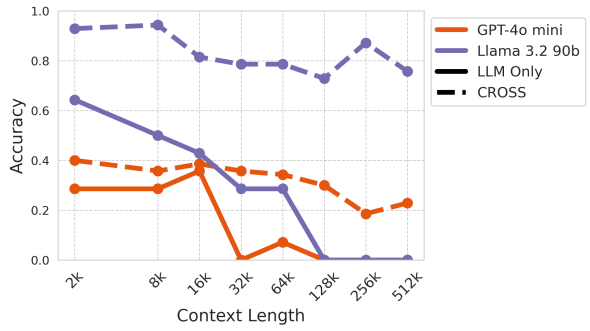
Figure 1. Radar Plots Comparing Average Accuracy in Different Tests. Dashed lines represent the CROSS-enhanced models, while solid lines represent the baseline LLMs. Red indicates GPT-4o-mini and Blue indicates Llama 3.2.

drop-off associated with large contexts, enabling both GPT-4o-mini and Llama 3.2 to perform reliably on larger-scale retrieval tasks.

Notably, this pattern persists in the more challenging 3-needles test. CROSS continues to stabilize retrieval accuracy across increasing context lengths for both models, as shown in Figure 2b, further emphasizing its robustness in scenarios with multiple target sentences.



(a) Comparison of Retrieval Accuracy Across Context Lengths in 1-Needle Test



(b) Comparison of Retrieval Accuracy Across Context Lengths in 3-Needles Test

Figure 2. Retrieval Accuracy Across Different Context Lengths

Needle Position Sensitivity

To assess CROSS's effectiveness in addressing the "lost in the middle" issue, we measured retrieval accuracy across five needle positions (0%, 25%, 50%, 75%, and 100%). Both GPT-4o-mini and Llama 3.2 90b performed best when the needle was at the beginning (0%) or end (100%) of the context. However, GPT-4o-mini exhibited a significant drop in accuracy for mid-context positions (25%, 50%, 75%), with an average accuracy of only **27%**. Llama 3.2, being a newer model, handled mid-context positions better, achieving an average accuracy of **45%**, though it still showed a noticeable reduction compared to its performance at the boundaries.

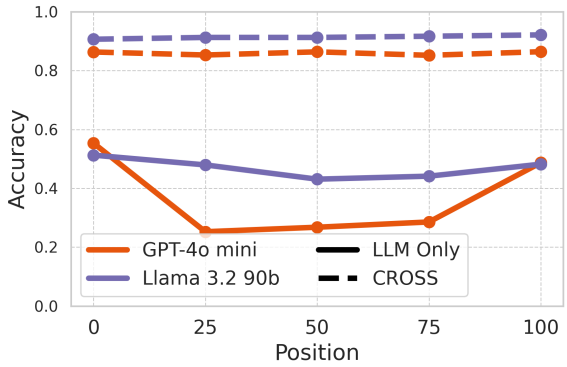


Figure 3. Retrieval Accuracy Comparison Across Needle Positions

When paired with CROSS, both models demonstrated a dramatic improvement in positional resilience. CROSS maintained stable performance across all needle positions, achieving an average mid-context accuracy of **86%** for GPT-4o-mini and **91%** for Llama 3.2. This indicates that CROSS effectively mitigates the loss in retrieval accuracy commonly associated with middle-positioned target information.

Notably, CROSS ensures consistent accuracy regardless of where the needle is located, addressing the challenges inherent in finding information deeply embedded within extensive contexts. This improvement underscores CROSS's ability to generalize effectively across models, resolving the "lost in the middle" problem even for a robust baseline like Llama 3.2 (Figure 3).

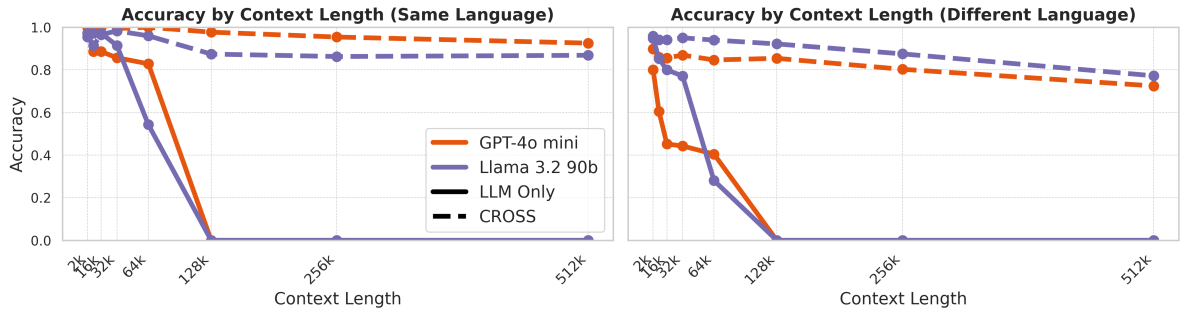


Figure 4. Comparison of Retrieval Accuracy in Same-Language and Cross-Lingual Settings in the 1-needle test. This figure illustrates the retrieval performance of CROSS when the prompt and context are in the same language versus when they differ.

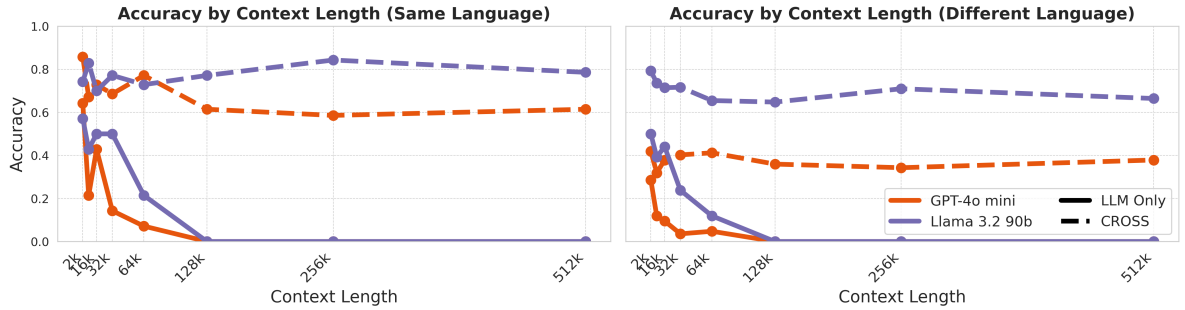


Figure 5. Comparison of Retrieval Accuracy in Same-Language and Cross-Lingual Settings in the 3-needles test. This figure highlights CROSS's performance in reasoning tasks involving multiple targets across different language settings.

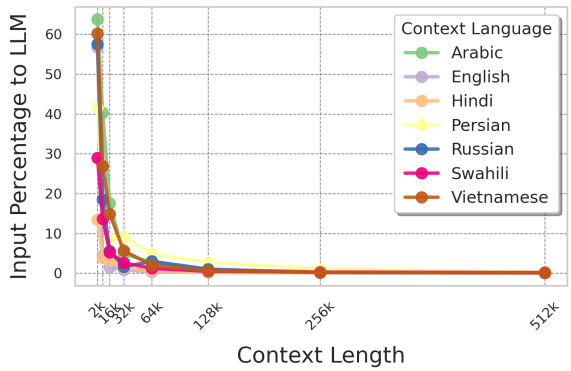


Figure 6. Mean Input Reduction Using CROSS by Context Length for Each Context Language

Cross-Lingual Consistency

Notably, CROSS demonstrates strong performance across linguistically dissimilar language pairs, such as Hindi-Russian, where the prompt and query are in Hindi, and the context is in Russian. In these challenging cross-lingual scenarios, GPT-4o-mini alone exhibits a marked reduction in accuracy, while Llama 3.2 fares better. When paired with CROSS, however, both models maintain high retrieval accuracy, demonstrating robust consistency even across widely varying linguistic structures.

For GPT-4o-mini, CROSS significantly boosts accuracy in cross-lingual pairs, bridging the gap between same-language and cross-language scenarios. Similarly, Llama 3.2 paired with CROSS achieves consistently strong performance, handling diverse language pairs effectively, and demonstrating its adaptability in multilingual contexts. This makes CROSS a robust solution for multilingual applications requiring retrieval across varied linguistic structures and combinations.

Figure 4 illustrates the performance of both models in the 1-needle test, comparing retrieval accuracy when the prompt and context languages are the same versus different. For the 3-needles test, a similar comparison is provided in Figure 5. These figures highlight CROSS's ability to maintain robust retrieval accuracy across both same-language and cross-lingual scenarios, even in more complex reasoning tasks.

Cost Efficiency Analysis

A key advantage of CROSS is its dramatic reduction in the number of tokens processed by the expensive language model. In a conventional setup, the entire context of T tokens is fed into the LLM, incurring a cost that scales linearly with T. In contrast, CROSS first processes the full context with a lightweight embedding model (BGE-M3) and then selects the top k sentences (each averaging roughly T/N tokens , where N is the total number of sentences in the haystack) to pass to the LLM. Thus, the LLM processes approximately

$$k \cdot \frac{T}{N} \quad \text{tokens},$$

instead of T tokens.

Assuming that the computational cost per token for the LLM is proportional to the model's parameter count, and noting that BGE-M3 (568M parameters) is roughly 180 times more efficient per token than Llama 3.2 (90B parameters), the cost incurred by the embedding stage is only a small fraction of that of the LLM. In our experiments, this two-phase approach resulted in an average reduction of token

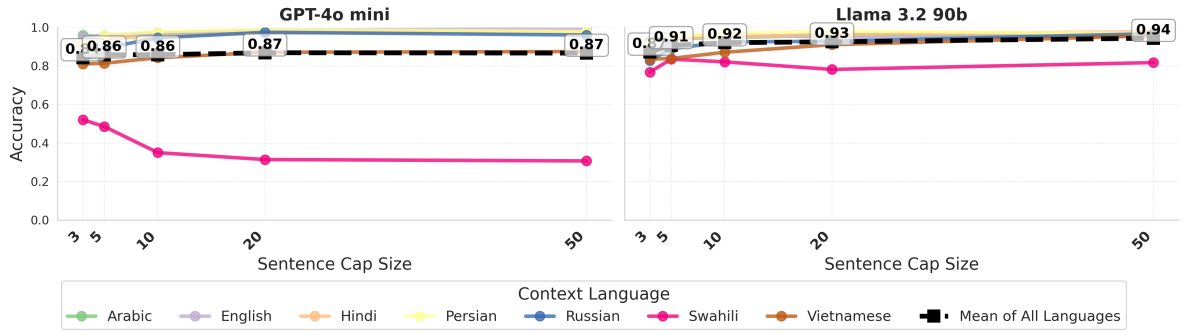


Figure 7. Accuracy of CROSS with varying sentence cap sizes in the 1-needle scenario, showing improved performance as the cap size increases.

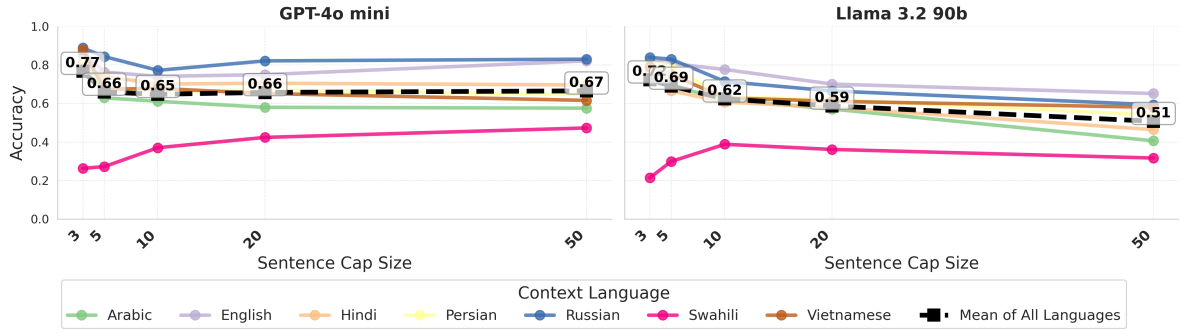


Figure 8. Accuracy of CROSS with varying sentence cap sizes in the 3-needles scenario, showing improved performance as the cap size increases.

usage for the LLM by about 90% across various context lengths (see Figure 6).

In addition to these costs, CROSS requires computing the semantic distances between the embedded query and each sentence's embedding. This involves a vector distance computation (typically a cosine or Euclidean similarity) for each of the N sentence embeddings. The computational cost of these distance calculations is generally:

$$\text{Cost}_{\text{distances}} \propto N \times d,$$

where d is the embedding dimension (e.g., 1024). In practice, since d is relatively small and these computations can be highly optimized (or even performed using approximate nearest-neighbor search techniques), the overall cost of the distance calculations is modest compared to the cost saved by significantly reducing the LLM's input size.

Thus, the overall computational cost of CROSS can be expressed as:

$$\text{Cost} = T \cdot C_{\text{embed}} + \left(k \cdot \frac{T}{N}\right) \cdot C_{\text{LLM}} + N \cdot d \cdot C_{\text{dist}} \quad (1)$$

where:

- C_{embed} is the per-token cost of the embedding model,
- C_{LLM} is the per-token cost of the LLM,
- C_{dist} is the per-dimension cost of computing distances.

Given that $C_{\text{embed}} \ll C_{\text{LLM}}$ and that the cost of the distance computations $(N \cdot d \cdot C_{\text{dist}})$ is relatively low, the overall efficiency gains are overwhelmingly driven by reducing the number of tokens fed into the LLM—an effect that is most pronounced when $k \ll N$.

In summary, by reducing the effective token input to the LLM by up to 90%, and with a minimal overhead for both embedding and distance computations, CROSS offers a scalable and economically efficient solution for handling extremely long contexts.

Ablation Results

Effect of Sentence Cap Size on Accuracy

To evaluate the impact of sentence cap size on the accuracy of CROSS, we conducted tests with cap sizes of 3, 5, 10, 20, and 50 sentences. The results revealed interesting patterns that varied depending on the number of target needles in the context and the underlying language model.

In the 1-needle scenario, as shown in Figure 7, accuracy generally increased with larger sentence cap sizes for both GPT-4o-mini and Llama 3.2. While variations were observed across different context languages, the overall trend demonstrated that providing more top-relevant sentences improved retrieval performance.

Conversely, in the 3-needles scenario, illustrated in Figure 8, accuracy decreased as the sentence cap size increased. This decline highlights a trade-off: although a larger cap size provides more context, it also introduces more distractor sentences, which can confuse the model in multi-needle retrieval tasks. These findings underline the importance of tailoring sentence cap sizes based on the complexity and requirements of the retrieval task.

These findings indicate that the optimal sentence cap size for CROSS depends on the context language and complexity of the retrieval task.

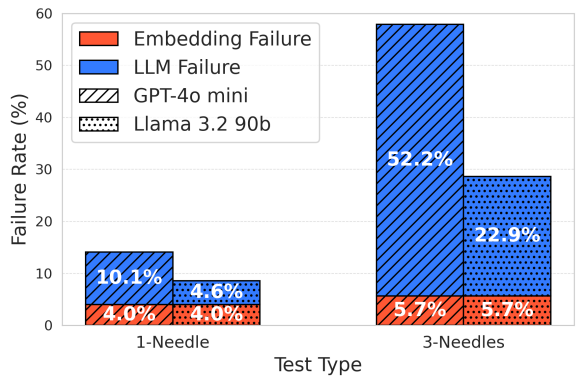


Figure 9. Failure rate analysis

Failure Analysis

While CROSS demonstrates significant improvements in retrieval accuracy, a closer examination of failure cases provides insights into its limitations, particularly in multi-target scenarios. This section analyzes the failure rates in both the 1-needle and 3-needles tests, distinguishing between failures arising in the embedding retrieval phase and those occurring within the language model's response generation.

We categorize retrieval failures into two types:

- **Embedding Failure**: Cases where the target label is absent from the retrieved sentence cap, indicating that the embedding model did not select the relevant sentences.
- **LLM Failure**: Cases where the language model fails to correctly extract or reason about the label, despite it being present in the retrieved sentence cap.

Failure Rates and Trends Figure 9 illustrates the failure rates across different test scenarios. In the 1-needle test, both embedding and LLM failures remain relatively low. The embedding model correctly retrieves the relevant sentence in 96% of cases, with embedding failures accounting for only **4.0%**. Similarly, LLM failures remain low, at **10.1%** for GPT-4o-mini and **4.6%** for Llama 3.2.

However, in the 3-needles test, we observe a substantial increase in LLM failures. Although embedding failures remain marginal at **5.7%**, LLM failures escalate significantly. GPT-4o-mini exhibits a failure rate of **52.2%**, while Llama 3.2, though performing better, still registers a notable **22.9%** failure rate. This indicates that while CROSS reliably retrieves relevant sentences, the challenge in the 3-needles scenario primarily lies in the model's ability to reason over multiple retrieved labels and correctly extract the appropriate one.

Equivalently, these embedding failure rates correspond to Embedding Recall@k of 96.0% in the 1-needle setting and 94.3% in the 3-needles setting, indicating that most residual errors in multi-needle tasks arise from reasoning/extraction rather than retrieval.

Impact of Cross-Lingual Settings on Failure Rates To further dissect the model's limitations, we analyzed failure rates by comparing scenarios where the prompt and context languages were the same against those where they were different. As illustrated in Figure 10, cross-lingual settings consistently amplify failure rates, with the most pronounced impact on the language model's reasoning capabilities.

While embedding failures see a slight increase in the cross-lingual setting (from 2% to 6%), they remain a minor contributor to the overall error rate. This suggests that the 'BGE-M3' model is highly effective at cross-lingual retrieval, though locating needles across languages is inherently more difficult.

The most dramatic trend is the sharp escalation in LLM failures, particularly in the 3-needles reasoning task. In the same-language setting, the LLM failure rate is already notable, but it skyrockets when the context is in a different language. For GPT-4o-mini, the failure rate for 3-needles reasoning jumps from 31% to over 62%, indicating a severe impairment in its ability to synthesize multiple facts presented in a foreign language. Although Llama 3.2 is more resilient, it also experiences a significant increase in reasoning failures in the cross-lingual condition. These results pinpoint cross-lingual, multi-target reasoning—not retrieval—as the primary bottleneck, revealing a critical area for improvement in future models.

Analysis of Increased LLM Failures in 3-Needles Test The increased failure rate in the 3-needles test suggests that the presence of multiple target sentences creates ambiguity, making it more difficult for the language model to consistently select the correct answer. Possible contributing factors include:

- **Increased Distractors:** The presence of multiple similar sentences in the sentence cap introduces additional reasoning complexity, leading to incorrect selections.
- **Inconsistent Answer Prioritization:** The LLM may struggle to determine the "largest" or "most relevant" label when multiple valid answers exist.
- **Ambiguity in Sentence Ranking:** Despite successful embedding retrieval, the semantic similarity between different needle sentences can lead to incorrect prioritization when forming the final response.

To further investigate whether the LLM failure is due to a lack of reasoning capability, we tested the 3-needles scenario on the o1-mini model, which is specifically designed for reasoning tasks (OpenAI 2024). The results, shown in Figure 11, indicate a significant reduction in LLM failure rates. GPT-4o-mini exhibited a LLM failure rate of **52.2%**, whereas o1-mini showed a much lower failure rate of **9.7%**. These results suggest that a model explicitly trained for reasoning can significantly mitigate failure rates in complex retrieval scenarios.

Effect of Sentence Cap Size on Failure Rates To analyze the impact of increasing the sentence cap size on retrieval failures, we evaluated both embedding and LLM failure rates across different cap sizes, as shown in Figure 12. In both the 1-needle and 3-needles scenarios, embedding failures decrease as the sentence cap size increases. This indicates that retrieving a larger number of sentences improves the likelihood of including the correct sentence in the context provided to the language model.

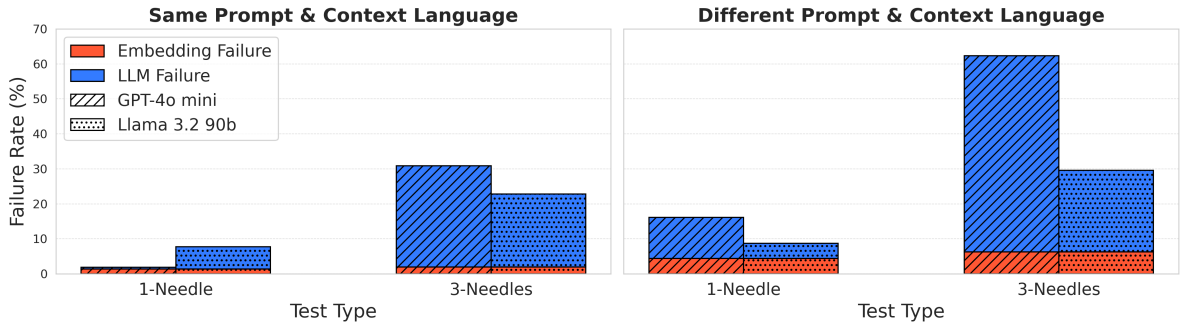


Figure 10. Failure rate in same vs different language

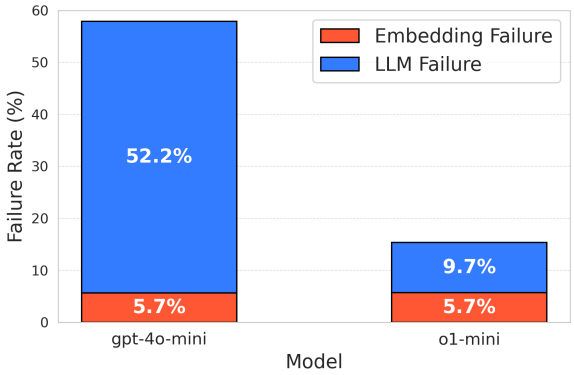


Figure 11. Comparison of failure rates between GPT-4o-mini and o1-mini in the 3-needles scenario.

However, a contrasting trend is observed with LLM failures. As the sentence cap size increases, LLM failures exhibit a noticeable rise, particularly in the 3-needles scenario. This suggests that while a larger cap size helps ensure the retrieval of relevant information, it also introduces additional distractor sentences, making it more difficult for the language model to accurately extract or reason about the correct label. These findings highlight the trade-off between retrieval accuracy and reasoning complexity when determining the optimal sentence cap size.

Types of LLM Failures To better understand the nature of LLM failures, we further categorize them into:

- **Incorrect Answer Failures**: Cases where the model provides an incorrect label despite the correct label being present in the retrieved sentence cap.
- **Unanswerable Failures**: Cases where the model incorrectly responds with "UNANSWERABLE" even though the correct label is available.

Figure 13 presents the breakdown of these failure types in both the 1-needle and 3-needles scenarios. In the 1-needle test, failure rates for both incorrect answers and unanswerable responses remain relatively low. However, in the 3-needles test, we observe a significant increase in unanswerable failures, particularly with GPT-4o-mini, where over 45% of total responses were classified as unanswerable despite the correct label being retrievable.

NeuroSymbolic Reasoning (NSAR)

Although RAG narrows down the context and alleviates many challenges inherent to long input passages, it alone does not guarantee robust multi-target reasoning. To address this shortcoming, we enhance our baseline RAG system with NSAR component. Besides *RAG-Vanilla*, we evaluate three prompting-based methods—*Chain-of-Thought* (CoT), *ReAct*, and *Self-Reflection*—as well as a hybrid approach (*NSAR+3*) that combines NSAR with all three prompting strategies. To ensure a fair comparison, all reasoning baselines (CoT, ReAct, Self-Reflection) utilize the same CROSS retrieval backbone to fetch the relevant context; they differ only in the prompting strategy used to generate the answer.

In the hybrid *NSAR+3* approach, we combine neurosymbolic reasoning with Chain-of-Thought, ReAct, and Self-Reflection strategies via a unified prompt. Specifically, the model is instructed to follow a sequential six-step process: (1) extract relevant facts as symbolic triples; (2) provide a step-by-step Chain-of-Thought explanation; (3) describe the intended action (ReAct); (4) generate executable Python code to compute the result; (5) reflect on the reasoning process to verify soundness. This comprehensive sequence ensures that the symbolic execution is grounded in semantic planning and iterative verification.

As depicted in Figure 14, the *RAG-Vanilla* baseline lags behind in multi-target reasoning, confirming that narrowing the context alone does not suffice to fuse and compare multiple pieces of information. In contrast, our proposed approach NSAR substantially improves accuracy by leveraging explicit symbolic extraction and Python-based reasoning. Moreover, combining NSAR with CoT, ReAct, and Self-Reflection (*NSAR+3*) yields the highest accuracy overall. Notably, for GPT-4o-mini, *NSAR* achieves 91.1%, followed closely by *NSAR+3* and *CoT*, both at 90.2%, whereas for Llama 3.2, *NSAR+3* attains the highest performance at 93.8%. These findings suggest that integrating explicit symbolic reasoning can fill critical gaps in retrieval-augmented generation, particularly for complex tasks that demand robust compositional inference.

Accuracy by context language Figures 15a and 15b provide a more granular perspective on how each approach performs across seven context languages. Each cell represents the accuracy (%) of a particular approach–language pair, revealing where certain strategies excel or fall short.

Across both heatmaps, *NSAR* and *NSAR+3* maintain high accuracy in most languages, confirming the benefits of

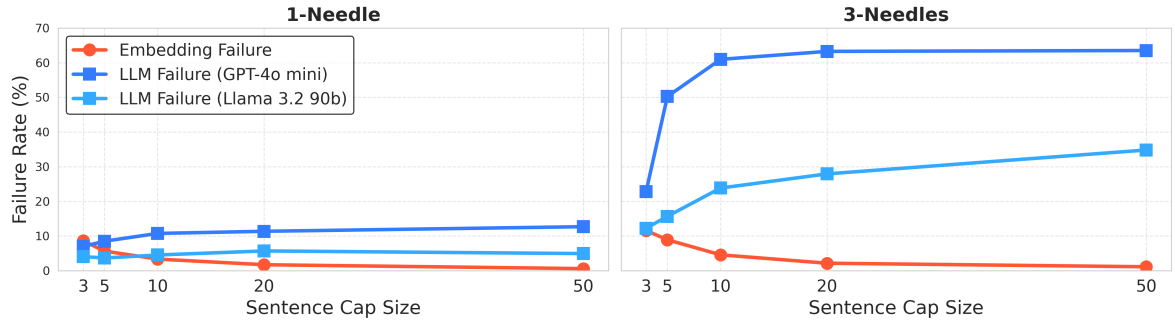


Figure 12. Effect of Sentence Cap Size on Failure Rates in the 1-Needle and 3-Needles Scenarios.

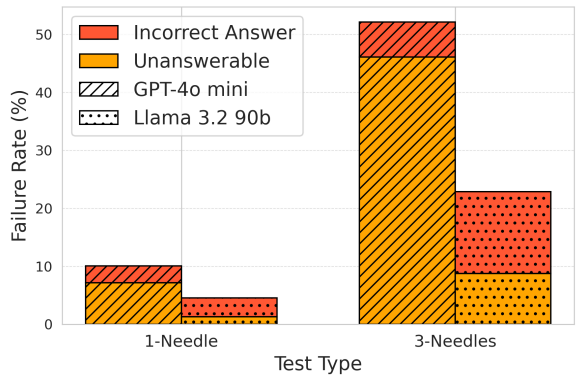


Figure 13. LLM Failure Breakdown

explicit symbolic reasoning for multi-target retrieval tasks. In contrast, baseline methods (*RAG-Vanilla*) and single-step prompting strategies (Chain-of-Thought, ReAct, Self-Reflection) exhibit more variability and struggle in specific languages. Notably, languages such as Swahili and Arabic appear more challenging, yet neurosymbolic approaches still achieve competitive performance. These language-specific patterns underscore the importance of robust, compositional reasoning—particularly in cross-lingual or lower-resource settings.

Effect of k on performance Figure 16 shows how different reasoning strategies perform as we vary the number of retrieved sentences k (3, 5, 10, 20, and 50) for GPT-4o-mini (left) and Llama 3.2 90b (right). Several trends are evident. First, at low values of k, all methods tend to have lower accuracy, likely due to the increased chance of missing key information during retrieval. As k increases, accuracy generally improves up to a point. However, very large k (e.g., 50) can introduce additional distractors, leading to a decline in performance. This aligns with our earlier observations that, while a broader retrieval scope reduces the risk of overlooking relevant facts, it can also complicate the model's reasoning by introducing more non-essential content.

Comparing across models, Llama 3.2 shows more pronounced fluctuations as k increases, suggesting it is more sensitive to context size and potential distractors. In contrast, GPT-4o-mini maintains relatively stable performance at intermediate k values. Notably, NSAR+3 consistently outperforms purely neural prompting methods in Llama 3.2, whereas GPT-4o-mini exhibits closer competition among NSAR and Chain-of-Thought. Overall, these findings highlight the importance of carefully tuning k to balance retrieval breadth and processing load, while also demonstrating that neurosymbolic reasoning can mitigate many of the challenges introduced by larger context windows.

Error analysis for NSAR and NSAR+3 Figure 17 illustrates the distribution of error types for GPT-4o-mini and Llama 3.2 under the *NSAR* and *NSAR+3* methods, categorizing failures into two types:

- Facts: The model retrieved the correct segments but failed to extract the target fact from the input.
- Code: Although the target fact was extracted correctly, the model produced incorrect or incomplete Python code, leading to an erroneous final answer.

The distribution of fact-extraction versus code-generation failures varies notably between the two models and across the two neurosymbolic methods. In *NSAR* for GPT-4o-mini, most errors stem from fact extraction, whereas Llama 3.2 primarily struggles with code generation. This suggests that GPT-4o-mini's neurosymbolic pipeline frequently have problem to locate the correct sentences to extract facts, while Llama 3.2 successfully extracts facts but sometimes produces flawed Python code. Moving from *NSAR* to *NSAR+3* reverses this trend for Llama 3.2, significantly reducing code-generation failures but leading to more fact-extraction issues. Meanwhile, GPT-4o-mini exhibits a small increase in fact-extraction errors and a modest rise in code-generation errors when switching to *NSAR+3*.

Overall, these results imply that while neurosymbolic reasoning substantially mitigates retrieval shortcomings, the balance between accurate fact extraction and correct code generation can shift depending on the underlying model and the specific prompting strategy.

Discussion

Our results demonstrate a significant leap forward in robust, multilingual, long-context reasoning, underscoring the power of a hybrid neural-symbolic approach. The core success of our framework stems from a strategic division of labor: the neural component (the BGE-M3 embedding model) excels at scalable pattern recognition and semantic understanding across languages, while the symbolic component (NSAR's code generation) provides the structured, deterministic, and interpretable reasoning that

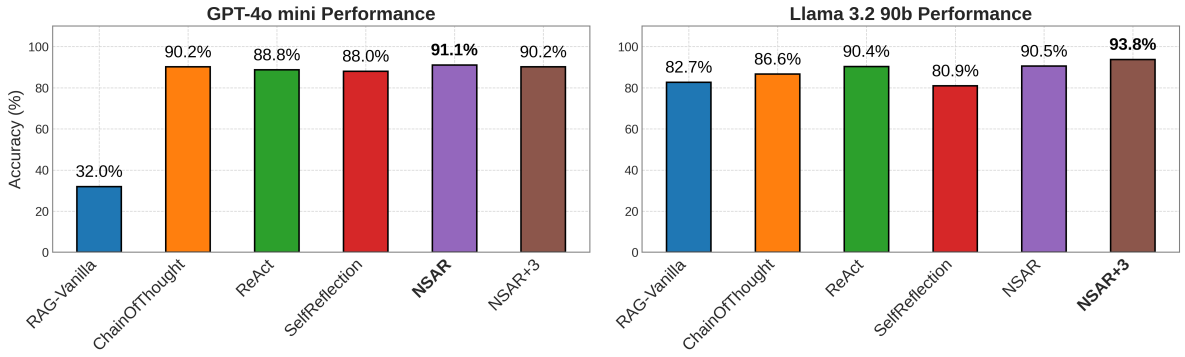
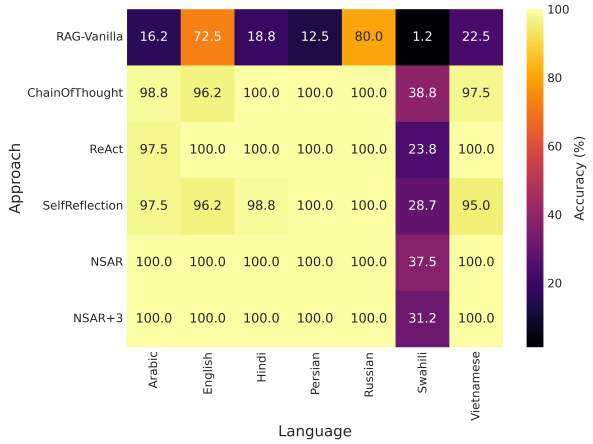
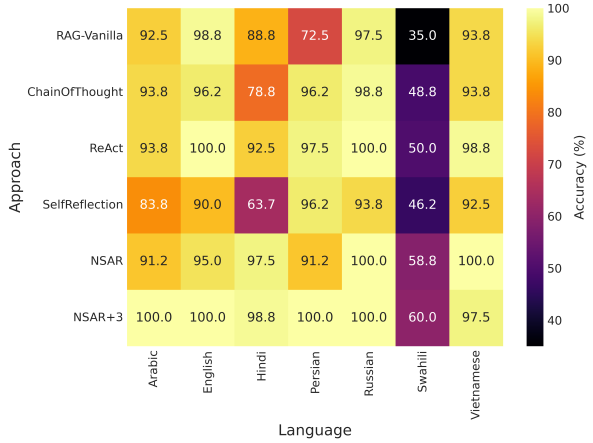


Figure 14. Overall accuracy of GPT-4o-mini (left) and Llama 3.2 90b (right) under different reasoning strategies (RAG-Vanilla, CoT, ReAct, Self-Reflection, NSAR, and a combined approach which combines NSAR with other reasoning methods (*NSAR+3*).



(a) GPT-4o mini.



(b) Llama 3.2 90b.

Figure 15. Heatmaps illustrating the accuracy (%) of different approaches (rows) across seven context languages (columns). Darker cells indicate **lower** accuracy, while lighter cells indicate **higher** accuracy.

purely neural models lack. By decoupling retrieval from reasoning, our system effectively circumvents the inherent limitations of monolithic LLM architectures.

A key finding is the framework's exceptional cross-lingual and cross-script stability. While end-to-end neural models often exhibit performance degradation when faced with typologically distant languages, our approach maintains high accuracy across Latin, Cyrillic, Devanagari, and Arabic scripts (Figure 1). This resilience arises because the shared multilingual embedding space normalizes linguistic differences at the retrieval stage, and the subsequent reasoning is performed in the language-agnostic domain of Python code. This design makes the reasoning process independent of the source language's syntax or structure—a critical advantage for building truly global AI systems.

Furthermore, our CROSS framework offers a pragmatic and highly effective solution to the well-documented "lost-in-the-middle" problem. Instead of attempting to architecturally enhance the attention mechanism of an LLM to handle vast contexts, CROSS reframes the problem by ensuring the LLM only ever processes a small, highly relevant subset of the document. As shown in Figure 3, this simple yet powerful pre-filtering step virtually eliminates performance degradation based on needle position, allowing models like GPT-4o-mini and Llama 3.2 to operate at near-peak efficiency regardless of where information is located.

The ablation studies reveal important trade-offs. The optimal number of retrieved sentences (k) depends critically on task complexity (Figure 16). For single-target retrieval (1-needle), a larger k increases the probability of capturing the correct sentence, improving accuracy. However, for multi-target reasoning (3-needles), a larger k introduces more distractors, increasing the cognitive load on the LLM and leading to a higher rate of reasoning failures. This is precisely where NSAR proves its value. By forcing the LLM to structure its reasoning through explicit fact extraction and code generation, NSAR mitigates the negative impact of distractors, reducing reasoning failures fivefold compared to a vanilla RAG setup (Figure 14).

Nonetheless, our framework is not without limitations. The entire pipeline's performance is contingent on the quality of the initial embedding-based retrieval; if the correct sentences are not among the top k, the failure is irrecoverable. Our failure analysis confirms this, though embedding failures were relatively rare (Figure 9). Moreover, the current symbolic representation—'FACT(entity, attribute, value)'—is tailored to this specific task. Real-world applications will demand richer formalisms capable of representing complex relationships, temporal dynamics, and uncertainty. The errors in code generation, though reduced, also highlight the need for robust verification and sandboxing mechanisms for deployable systems.

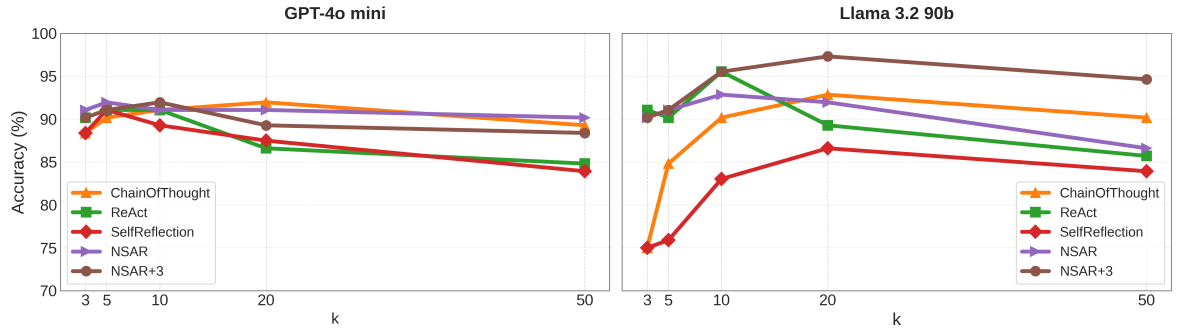


Figure 16. Accuracy versus k (the number of retrieved sentences) for GPT-4o-mini (left) and Llama 3.2 (right).

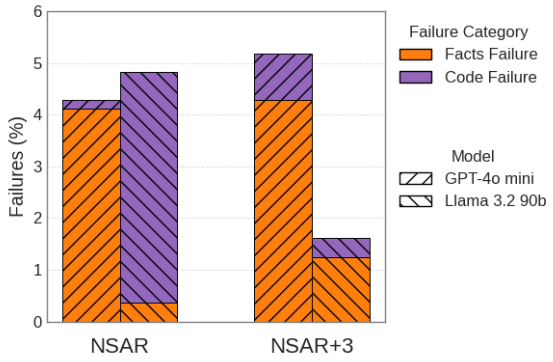


Figure 17. Error sources under *NSAR* and *NSAR+3*.

Regarding the complexity of reasoning, we acknowledge that the current symbolic representation (subject-attribute-value triples) is relatively elementary. While this structure suffices for aggregation tasks like finding the "largest" value, real-world applications often demand richer relational structures. However, the strength of the NSAR framework lies in the *Python code generation* step. Even if the extracted facts are simple, the generated code can implement arbitrarily complex logic (e.g., loops, conditional filtering, and library integrations) that purely formal logic solvers might struggle to scale.

Furthermore, while specialized reasoning models like *o1-mini* demonstrate low failure rates (9.7%) without symbolic augmentation, they remain opaque "black boxes." In contrast, NSAR offers a distinct advantage in auditability: the generated Python code serves as a verifiable proof of the reasoning process. This is critical in high-stakes domains where identifying *why* a model failed is as important as the answer itself. Additionally, the computational cost of running a smaller model (like GPT-4o-mini) with NSAR is significantly lower than employing heavy reasoning-optimized models for every query.

Looking forward, this work opens several promising avenues. Extending the symbolic layer to more expressive systems like first-order logic or probabilistic programs could unlock more sophisticated reasoning capabilities. Future systems could also feature an adaptive mechanism to dynamically select the optimal sentence cap size based on query complexity. Ultimately, our findings suggest a paradigm shift away from the pursuit of ever-larger, monolithic neural models and toward hybrid architectures that thoughtfully integrate neural and symbolic components. For mission-critical domains where reliability, interpretability, and auditability are paramount, such neurosymbolic systems represent the most viable path toward truly intelligent and trustworthy AI.

Conclusion

In this work, we have addressed the persistent challenges of multilingual retrieval and multi-target reasoning in long-context scenarios by introducing a hybrid neural-symbolic architecture. Our framework, **CROSS**, efficiently narrows massive multilingual documents to concise, relevant segments using advanced multilingual embeddings, substantially improving retrieval accuracy and overcoming the "lost-in-the-middle" problem. Building upon CROSS, our **NeuroSymbolic Augmented Reasoning (NSAR)** module brings explicit symbolic inference into the loop, prompting LLMs to extract structured facts and generate executable Python code for robust, interpretable reasoning.

Comprehensive experiments on the mLongRR-V2 benchmark—spanning seven languages, 49 cross-lingual pairs, and contexts up to 512,000 words—demonstrate that our approach consistently outperforms neural-only baselines, achieving state-of-the-art results in both retrieval and reasoning across diverse linguistic and contextual settings. NSAR not only delivers a fivefold reduction in reasoning failures but also maintains stable performance for traditionally challenging languages and extreme context sizes.

Our findings highlight the promise of hybrid neural-symbolic systems for building scalable, transparent, and reliable AI in real-world multilingual environments. By bridging neural flexibility with symbolic rigor, this paradigm sets a new standard for auditable and generalizable information extraction and reasoning. Future work will extend this approach to richer symbolic representations, more complex compositional tasks, and broader application domains, further advancing the frontier of neurosymbolic artificial intelligence.

References

Agrawal A, Dang A, Bagheri Nezhad S, Pokharel R and Scheinberg R (2024) Evaluating multilingual long-context models for retrieval and reasoning. In: Sälevä J and Owodunni A (eds.) *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*. Miami, Florida, USA: Association for Computational Linguistics, pp. 216–231. DOI:10.18653/v1/2024.mrl-1.18. URL https://aclanthology.org/2024.mrl-1.18/.

Anthropic (2024) The claude 3 model family: Opus, sonnet, haiku. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

Beltagy I, Peters ME and Cohan A (2020) Longformer: The long-document transformer. URL https://arxiv.org/abs/2004.05150.

Chen J, Xiao S, Zhang P, Luo K, Lian D and Liu Z (2024a) M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In: Ku LW, Martins A and Srikumar V (eds.) *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, pp. 2318–2335. DOI:10.18653/v1/2024.findings-acl.137. URL https://aclanthology.org/2024.findings-acl.137.

Chen T, Wang H, Chen S, Yu W, Ma K, Zhao X, Zhang H and Yu D (2024b) Dense X retrieval: What retrieval granularity should we use? In: Al-Onaizan Y, Bansal M and Chen YN (eds.) *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics, pp. 15159–15177. DOI:10.18653/v1/2024.emnlp-main.845. URL https://aclanthology.org/2024.emnlp-main.845/.

d'Avila Garcez A and Lamb LC (2020) Neurosymbolic ai: The 3rd wave. URL https://arxiv.org/abs/2012.05876.

Gao Y, Xiong Y, Gao X, Jia K, Pan J, Bi Y, Dai Y, Sun J, Wang M and Wang H (2024) Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* URL https://arxiv.org/abs/2312.10997.

Jiang Z, Ma X and Chen W (2024) Longrag: Enhancing retrieval-augmented generation with long-context llms. URL https://arxiv.org/abs/2406.15319.

Kiss T and Strunk J (2006) Unsupervised multilingual sentence boundary detection. *Computational Linguistics* 32(4): 485–525. DOI:10.1162/coli.2006.32.4.485. URL https://doi.org/10.1162/coli.2006.32.4.485.

Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih Wt, Rocktäschel T, Riedel S and Kiela D (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. In: *Advances in Neural Information Processing Systems (NeurIPS)*. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Li B, Haider S, Luo F, Agashe A and Callison-Burch C (2024) Bordirlines: A dataset for evaluating cross-lingual retrieval-augmented generation. URL https://arxiv.org/abs/2410.01171.

Litschko R, Vulić I and Glavaš G (2022) Parameter-efficient neural reranking for cross-lingual and multilingual retrieval. In: Calzolari N, Huang CR, Kim H, Pustejovsky J, Wanner L, Choi KS, Ryu PM, Chen HH, Donatelli L, Ji H, Kurohashi S, Paggio P, Xue N, Kim S, Hahm Y, He Z, Lee TK, Santus E, Bond F and Na SH (eds.) *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, pp. 1071–1082. URL https://aclanthology.org/2022.coling-1.90.

Liu NF, Lin K, Hewitt J, Paranjape A, Bevilacqua M, Petroni F and Liang P (2023) Lost in the middle: How language models use long contexts. URL https://arxiv.org/abs/2307.03172.

Nezhad SB and Agrawal A (2025) Enhancing large language models with neurosymbolic reasoning for multilingual tasks. In: H Gilpin L, Giunchiglia E, Hitzler P and van Krieken E (eds.) *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning*, *Proceedings of Machine Learning Research*, volume 284. PMLR, pp. 1059–1076. URL https://proceedings.mlr.press/v284/nezhad25a.html.

Nie JY (2010) *Cross-language information retrieval*. Morgan & Claypool Publishers.

OpenAI (2024) Openai o1 system card. https://cdn.openai.com/o1-system-card-20241205.pdf. Accessed: Month Day, Year.

Team G (2024) Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. URL https://arxiv.org/abs/2403.05530.

Wang X, Wang Z, Gao X, Zhang F, Wu Y, Xu Z, Shi T, Wang Z, Li S, Qian Q, Yin R, Lv C, Zheng X and Huang X (2024) Searching for best practices in retrieval-augmented generation. In: Al-Onaizan Y, Bansal M and Chen YN (eds.) *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics, pp. 17716–17736. DOI:10.18653/v1/2024.emnlp-main.981. URL https://aclanthology.org/2024.emnlp-main.981/.

Xu P, Ping W, Wu X, McAfee L, Zhu C, Liu Z, Subramanian S, Bakhturina E, Shoeybi M and Catanzaro B (2024) Retrieval meets long context large language models. URL https://arxiv.org/abs/2310.03025.

Yang E, Nair S, Lawrie D, Mayfield J, Oard DW and Duh K (2024) Efficiency-effectiveness tradeoff of probabilistic structured queries for cross-language information retrieval. URL https://arxiv.org/abs/2404.18797.

Zaheer M, Guruganesh G, Dubey A, Ainslie J, Alberti C, Ontanon S, Pham P, Ravula A, Wang Q, Yang L and Ahmed A (2021) Big bird: Transformers for longer sequences. URL https://arxiv.org/abs/2007.14062.