LC-Eval: A Bilingual Multi-Task Evaluation Benchmark for Long-Context Understanding

Anonymous ACL submission

Abstract

Recent advancements in Large Language Models (LLMs) have demonstrated sophisticated capabilities, including the ability to process and comprehend extended contexts. These emergent capabilities necessitate rigorous evaluation methods to effectively assess their performance in long-context understanding. In this paper, we present **LC-Eval**, a bilingual, multi-task evaluation benchmark designed to evaluate long-context understanding in English and Arabic, targeting context lengths ranging from 4k to over 128k tokens. LC-Eval introduces four novel and challenging tasks: multi-document question answering, bilingual question answering, claim verification within a paragraph, and multiple-choice questions based on long contexts. These tasks are designed to assess LLMs' abilities in deep reasoning, document comprehension, information tracing, and bilingual information extraction and understanding. The benchmark includes datasets in both Arabic and English for each task, allowing for a comparative analysis of their performance across different text genres. Evaluations were conducted on both open-weight and closed LLMs, with results indicating that LC-Eval presents significant challenges. Even high-performing models, such as GPT-4o, struggled with certain tasks, highlighting the complexity and rigor of the benchmark.

1 Introduction

Context length of Large Language Models (LLMs) typically indicates how many tokens a language model can process as an input. Although early models can process up to 4k tokens, more recent models have the context length varying from 8k to 128k even to 1M (Anthropic; OpenAI; Dubey et al., 2024; Huang et al., 2024; Cohere; Qwen et al., 2025; DeepSeek-AI et al., 2025). These long context language models (LCLMs) are extremely helpful for understanding long documents, minimizing hallucinations and retrieval augmented generation (RAG).

Since Arabic is one of the major languages spoken by more than 400 million people as their mother tongue (WorldData.info, 2024), a number of Arabic Large Language models that understand both Arabic and English have been released (Bari et al., 2024; Sengupta et al., 2023; Zhu et al., 2024; Abbas et al., 2025), . However, these models are often evaluated using English benchmarks or proprietary datasets, making it challenging to publicly benchmark their performance in Arabic or to assess their capabilities across various tasks. Additionally, although benchmark datasets in English cover aspects such as reasoning, document summarization, and document understanding, most of these benchmarks lack a focus on deep reasoning. Moreover, many task-specific datasets fail to fully evaluate LCLMs across their entire context length, leading to the need for new benchmark datasets both for Arabic and English (Bai et al., 2025).

Another significant challenge is evaluating the responses of LCLMs to open-ended questions. Most existing evaluation methods rely on exact word matching. However, even when queried with the same question twice, the same LCLM can generate responses that are semantically equivalent but phrased differently, making exact word matching an unreliable evaluation criterion. Another alternative approach is to use similarity-based method which addresses some issues of word matching. However, it can also lead to incorrect evaluations. For example, the sentences "The capital of France is Paris" and "The capital of France is Rome" could have a high similarity score, yet their semantic meanings are entirely different.

To address these issues, we introduce `LC-Eval`, a bilingual multi-task evaluation benchmark for English and Arabic long-context understanding, covering context lengths from 4K to more than 128K. `LC-Eval` comprises four challenging tasks:

(i) multi-document question answering (QA), (ii) bilingual QA, (iii) claim verification within a paragraph, and (iv) multiple-choice QA. These tasks collectively assess LCLMs' ability in deep reasoning, document understanding, information tracing, and bilingual information extraction. Additionally, we propose an entity relationship based evaluation method, an approach inspired from (Goodrich et al., 2019), using LLM as a judge that takes semantic meaning into account when evaluating open-ended question answer compared to gold standard answer.

Our dataset was initially generated using GPT-4 (with task-specific prompts), followed by multi-stage refinements to increase complexity. To ensure accuracy, three human annotators validated all data, with majority agreement determining the final verdict. The validation criteria were task-specific, and human validators received specialized training on the respective tasks before beginning the validation process.

Evaluation results show that `LC-Eval` poses significant challenges for LCLMs. Our key contributions are:

1. A large-scale dataset of 7,903 samples, spanning context lengths from 4K to over 128K and targeting deep reasoning, document tracing, and bilingual information extraction (Table 1).

2. Both Arabic and English datasets, enabling a broader assessment of LCLMs' performance across different languages and text genres.

3. An entity-based evaluation approach that accounts for semantic meaning in open-ended question answering (Section 6.2.1).

4. Multiple complementary evaluation metrics—including entity relationships, recall, and accuracy—for comprehensive performance assessment (Section 6.2).

The rest of the paper is organized as follows: related work is presented in section 2. In section 3, we present an overview of all the tasks that are being evaluated, while in section 4 we give the details of how the datasets were curated. Section 5 highlights the data validation process. In section 6, the experiment setup is presented and the actual evaluation and results are presented in section 7. Then we conclude in section 8.

2 Related work

Previous research on LCLMs evaluation can be broadly classified into two categories: synthetic tasks and nonsynthetic tasks focusing on real-world scenarios. Synthetic tasks typically involve artificially generated texts or texts from various sources, into which specific information (referred to as the "needle") is deliberately inserted. The objective of long-context language models (LCLMs) is to accurately retrieve this information from the text (Hsieh et al., 2024; Hengle et al., 2024). Such tasks are commonly referred to as "needle-in-a-haystack" (NIAH) problems and exist in several variations. Since the data in these tasks is synthetically generated, they can be scaled to accommodate infinite context lengths. As NIAH tasks effectively test LCLMs' capabilities in information extraction and document understanding, they are often employed to evaluate the initial performance of these models (Dubey et al., 2024). Despite their utility, NIAH tasks are limited in diversity, and the insertion of the "needle" often results in abrupt topic shifts within the text. These limitations make NIAH tasks insufficient as standalone measures for evaluating the performance of LCLMs.

To address the need for nonsynthetic evaluation datasets, various English and multilingual datasets have been developed. These datasets cover a broad range of tasks, including general reasoning, document understanding, document summarization, and claim verification (Yang et al., 2018; Shaham et al., 2023; An et al., 2023; Bai et al., 2024; Lee et al., 2024; Karpinska et al., 2024; Wang et al., 2024). Additionally, task-specific evaluation datasets have been introduced for domains such as question answering (Pang et al., 2022), summarization (Wang et al., 2022), and coding (Bogomolov et al., 2024). However, datasets constructed before 2024 had a context length of less than 16k tokens (Yang et al., 2018; Pang et al., 2022; Wang et al., 2022). Since the context length of long-context language models (LCLMs) has increased to 128k tokens or more—particularly from early 2024 onward (Anthropic; OpenAI; Dubey et al., 2024; Huang et al., 2024; Cohere; Qwen et al., 2025; DeepSeek-AI et al., 2025)—the development of new benchmarks for LCLMs with extended context lengths has accelerated.

Although some of these datasets are multilingual, there remains a lack of Arabic benchmark data, highlighting the need for the development of new

Arabic benchmark datasets. Additionally, some existing datasets do not include tasks that require deep reasoning. Furthermore, most evaluations rely on exact information matching or BLEU/ROUGE scores (Wang et al., 2024). However, evaluating long-text generation based on exact match is challenging, and BLEU/ROUGE scores do not account for semantic meaning, making them insufficient as sole indicators of an LCLM's performance (Goodrich et al., 2019). To address these limitations, we constructed a new benchmark dataset for both Arabic and English, incorporating four distinct tasks that require deep reasoning. Additionally, we introduced an entity relationship-based evaluation method that considers conceptual meaning for assessing relevant tasks.

3 Tasks Overview

3.1 Multi-document Question Answering

In this task, answering the question requires knowledge from multiple documents. Given the involvement of multiple documents, some serve as distractors, closely resembling the relevant documents from which the answer needs to be derived. The task is to identify correct documents and form a response to the question using correct documents. This task tests analytical depth (e.g., Cross-Document Reasoning, Contextual Understanding), generative proficiency (e.g., synthesis, coherent output) and information tracing (from which documents the answer is derived), ensuring the model can navigate complex, real-world scenarios where information is fragmented and noisy.

3.2 Bilingual Question Answering

Bilingual QA assesses an LCLM's ability to understand and process information across different languages. For instance, a document may be written in one language while the question is in another. Users of an LCLM typically expect responses in the same language as the question, regardless of the document's original language. To address this challenge, we designed a task in which the LCLM must accurately answer a question in the same language as the question, even when the context is in a different language. Since our focus is on Arabic and English, this evaluation demonstrates the model's capacity to comprehend content in one language while generating responses in another, thereby assessing its cross-lingual understanding and generation capabilities.

3.3 Claim Verification

A claim is a statement that can be evaluated as either true or false. When information is extracted from a large document, it may consist of multiple lines, each of which may contain accurate or erroneous information. Given this scenario, the task of claim verification involves identifying each true and false claims within a paragraph. Since this setup simulates real-world scenarios where the statements in a paragraph are not direct extractions from the given context, accurately determining their truthfulness necessitates the reasoning capabilities of LCLMs.

3.4 Multiple Choice Question Answering

Multiple-choice question answering refers to the task in which a question is presented along with a set of possible answer choices. The objective is to identify the correct answer from the given options. This task typically requires a combination of document understanding and reasoning to accurately determine the correct response.

4 Data Curation

Our data collection process drew from both Arabic and English corpora, leveraging multiple publicly available datasets to ensure broad coverage. We utilized the 2024 Wikipedia dumps, WikiNews, WikiHow, and WikiBooks for both languages, providing a rich mix of encyclopedic, instructional, and news content. Additionally, we incorporated English books from Project Gutenberg (Gutenberg) and Arabic books from the Hindawi Organization (Hindawi) to ensure a well-balanced representation of formal and literary language. For timely and relevant news content in both languages, we included articles from the Saudi Press Agency[1]. While these datasets are non-parallel, they provide valuable coverage across diverse domains such as economy, biology, and many more.

Since these datasets may contain harmful content, such as hate speech, we employed a custom word-based dictionary filtering method to remove potentially harmful content. For dynamic tasks (e.g., multi-document QA), we sampled according to domain, word count, and source, distributing samples uniformly across sources (with complete inclusion of sources below 100 samples). For fixed tasks, the sampling aimed for a similar distribution

[1] https://www.spa.gov.sa/

across varying context lengths. In total we obtained 7,903 evaluation samples, Table 1 shows the overall statistics of selected samples. The following subsections describe the curation process for our dataset.

4.1 Multi-document Question Answering

We curated multi-document questions and answers based on Arabic and English inputs using the following steps:

1. For each document (main document) in the corpus, we compiled three sets: most similar, least similar, and same-domain documents. These sets were used to assemble multi-document inputs, with similarity determined using the Min-Hash measure.
2. We randomly selected one to three of the most similar documents and used GPT-4o to generate a question and answer based on the main document and its selected similar documents (see Appendix A.1.1 for the prompt).
3. GPT-4o then evaluated the quality of the generated pairs (see Appendix A.1.2 for the evaluation prompt).
4. We applied GPT-4o across four temperature values $(0.0, 0.3, 0.6, 0.9)$ and computed three key scores: majority vote assessment, average assessment, and majority vote average. High-quality instances were selected based on the following criteria: Accuracy, Grammar and Syntax, Cultural Sensitivity, and Safety ≥ 9, with an average majority vote ≥ 9.0. Instances exceeding 2,666 words (8,000 tokens) were identified as candidates for the long-context evaluation dataset.
5. From the resulting dataset, we selected 1,300 high-quality instances for human validation (see Section 5.1 for details on the validation procedure and Appendix A.2.1 for an example multi-document QA).

4.2 Bilingual Question Answering

We curated English questions and answers from Arabic documents and vice versa using the following steps:

1. Documents exceeding 2,666 words were divided into 1,000-word chunks, ensuring each chunk ended at a sentence boundary (., ?, !). A chunk was randomly selected with probabilities: 60% from the middle, 20% from the beginning (excluding the first), and 20% from the end (excluding the last). This selection facilitated human validation.
2. Using GPT-4o, a question and an answer were generated in English from the selected Arabic chunk (and vice versa). (see Appendix A.1.3 for the prompt).
3. The generated pairs were evaluated for quality using GPT-4o (see Appendix A.1.4 for the evaluation prompt).
4. We applied GPT-4o across four temperature values $(0.0, 0.3, 0.6, 0.9)$ and computed three key scores: majority vote assessment, average assessment, and majority vote average. High-quality instances were selected based on Accuracy, Grammar and Syntax, Cultural Sensitivity, and Safety ≥ 9, with an average majority vote ≥ 9.0.
5. GPT-4 identified the paragraph(s) within the document that contained the answer to the generated question (see Appendix A.1.5 for the prompt).
6. We selected 1,300 high-quality instances for human validation (see Section 5.2 for details on the validation procedure and Appendix A.2.2 for an example bilingual QA).

4.3 Claim Verification

Since most of the documents in Wikipedia, wikiHow have text less than 4k words, we chose books from Hindawi (Hindawi) and Project Gutenberg (Gutenberg) for Arabic and English respectively. To develop complex multiple-choice and claim verification datasets, we employ a two-step approach where the first step is the same for both datasets. In the first step, we generate a summary of each document using the GPT-4o API. Given that some documents are longer than the maximum context length of GPT-4o and it also has output token limitation, we divide the documents into 4,000-token chunks and request GPT-4o to summarize each segment. Furthermore, we supply the summaries of all preceding chunks as input to GPT-4o, prompting it to continue the summarization from the previously written summary. This methodology yields

Dataset	Number of Samples	Length Distribution						Avg Word Count.
		4K-8K	8K-16K	16K-32K	32K-64K	64K-128K	>128K	
AR Multidoc QA	1180	590	191	384	15	-	-	6,006
EN Multidoc QA	1186	513	327	298	48	-	-	6,894
AR Bilingual QA	1194	391	244	176	86	297	-	35,244
EN Bilingual QA	1191	342	215	119	159	356	-	41,424
AR Claim Verification	400	63	62	75	84	64	52	49,382
EN Claim Verification	400	58	56	67	68	67	84	57,667
AR MCQs	1200	200	200	201	193	199	207	52,814
EN MCQs	1152	165	162	201	210	208	206	55,546

Table 1: Data statistics for different tasks and context length.

a comprehensive and detailed summary of a document. The prompt used for summary generation is included in Appendix A.1.6.

In step two, GPT-4o is instructed to generate a paragraph that contains at most five claims, each of which may be true or false. We direct GPT-4o to refrain from introducing any external entities or external relationships between entities when creating false claims. In addition, We define a difficulty level for creating the claim paragraph. The prompt used for claim verification is included in Appendix A.1.8 and an example datapoint of claim verification is given in Appendix A.2.3.

4.4 Multiple Choice Question Answering

Multiple choice QA creation also involves two steps where the first step is the same as Claim verification described in section 4.3. In step 2 for MCQs, we prompt GPT-4o to generate MCQs based on the summaries, each comprising four options with one correct answer. The distractors may include partially correct answers. In addition, we define a difficulty level for the options to make it more difficult for LCLMs. The prompt used for MCQs generation is included in Appendix A.1.7 and an example datapoint of MCQ is given in Appendix A.2.4.

5 Data Validation

To ensure high-quality data, we engaged human annotators to review and validate the datasets. All annotators are paid undergraduate or post-graduate students of Saudi Arabia. In all tasks, each data sample was reviewed by three annotators. We only accept samples that were accepted by at least two annotators. The evaluation process and criteria vary depending on the nature of each dataset. In the following, we outline the process in detail.

5.1 Multi-document Question Answering

For this dataset, the human annotators were presented with a question, an answer, and three or four texts representing summaries of the corresponding documents. The annotators were asked to evaluate the data based on four key criteria: **Clarity** refers to whether the question is well-structured, unambiguous, and easily understood. **Cross-referencing** assesses whether the answer appropriately integrates information from all provided texts. **Correctness** ensures that the answer is accurate, complete, and strictly based on the given texts, without introducing external information. **Coherence** evaluates whether the texts are logically connected and consistently focused on the same topic. **Cultural and Safety Alignment** ensures that content aligns with Arabic cultural norms and promotes safety and well-being

5.2 Bilingual Question Answering

In the bilingual question-answering validation, human annotators reviewed entries consisting of a question, an answer, a question excerpt, and an answer excerpt. The question excerpt refers to a text segment selected from the original document, while the answer excerpt is a subset of the question excerpt that contains the answer. The question and answer were provided in English, whereas the excerpts were in Arabic and vice-versa. Each entry was evaluated based on three key criteria; **Clarity**, ensuring the question is well-structured, easily understood, and extracted from the question excerpt; **Correctness**, verifying the accuracy and completeness of the answer based on the answer excerpt without introducing external information; and **Cultural and Safety Alignment**, ensuring the content respects established cultural values and safety standards.

5.3 Claim Verification

In the claim verification task, human annotators were presented with a claim paragraph containing five claims, the true claims, the false claims, and the original book from which the claims were extracted. Each claim was individually reviewed to determine its veracity. All claims were verified based on factual accuracy with reference to the original book. Annotators were instructed to assess the claims based on their source alignment, accuracy, truthfulness, and falsehood. **Source Alignment** refers to the consistency of the claims with the original book from which they were derived, ensuring that the claims reflect the information found in the source material. **Accuracy** ensures that the true and false claims align with the content of the claim paragraph. **Truthfulness** refers to whether the true claims are inherently true, in accordance with established facts from the original source. **Falsehood** ensures that false claims are actually false, as they do not align with the factual content of the original book. **Cultural and Safety Alignment**, ensuring the content respects established cultural values and safety standards.

5.4 Multiple Choice Question Answering

In the Multiple Choice Question Answering task, annotators were provided with a book summary, a question with four answer choices, and an answer key. They were tasked with validating the samples based on five key criteria: **Clarity** assesses whether the question is well-structured, easily understood, and free from ambiguity. **Source-Driven** ensures that the question is derived directly from the source textbook. **Answer Correctness** verifies that the labeled answer corresponds to the correct choice. **Choice Distinctiveness** ensures that all answer choices are unique, with no duplicates. **Unambiguity** confirms that no answer choices are repeated, guaranteeing a clear and distinct set of options.

6 Experiment

6.1 Baseline

We selected five open-source 128k context-length LCLMs: Llama-3.1-8B Instruct (Dubey et al., 2024), Llama-3.3-70B Instruct, Qwen2.5-14B Instruct (Qwen et al., 2025), Command-r-plus08-2024 Instruct (Cohere), and Phi-3.5-mini Instruct (Abdin et al., 2024), along with two open-source 32k context-length LCLMs: AceGPT-v2-32B Instruct (Zhu et al., 2024) and Qwen2.5-72B Instruct, as baseline models. Additionally, we included the GPT-4o (OpenAI) API with a 128k context length. Since tokenizers vary across LCLMs, the number of words corresponding to a given context length differs by model. Table 2 shows the token fertility rate for each tokenizer, indicating that 128k and 32k context lengths typically correspond to 64k and 16k words for Arabic, respectively, and about 106k words for English at 128k tokens. While some baseline models may exceed their reported context lengths, their performance usually degrades significantly. For a fair comparison, we measured context by word count and evaluated models within their reported context lengths.

Tokenizer	Language		Context Length
	Arabic	English	
GPT-4o	1.995	1.262	128K
AceGPT-v2-32B	2.350	1.273	32K
Command-r-plus	2.170	1.266	128K
Llama 3 family	2.332	1.269	128K
Phi-3.5-mini	2.203	1.417	128K
Qwen 2.5 family	2.350	1.273	32K/128K

Table 2: The average fertility rate of tokenizers. The fertility rate indicates the average number of tokens required per word.

6.2 Performance Metrics

6.2.1 Entity Relationship Accuracy

To evaluate the answers of multi-document and bilingual QA, we first identify entities and their relationships from the gold standard answers using GPT-4o, considering these as the gold standard entity relationships. Then the generated responses of baseline models are evaluated by employing GPT-4o to assess the degree of overlap between the entity relationships in the model-generated responses and the gold standard entity relationships. Since different models may use varying wording, we prompt GPT-4o to identify relationships based on conceptual meaning rather than lexical similarity. The accuracy of a model's response for a given prompt is calculated as the ratio of shared relationships-those present in both the gold standard and the generated response-to the total relationships in the gold standard. The final accuracy score is computed as the average accuracy across all samples. Entity relationship score accuracy ranges from 0 to 100 where higher score indicates a better result.

6.2.2 Recall@k

Recall@k typically refers to correctly identified top k documents from a set of relevant documents. We assessed the average recall for multi-document QA by taking the average of recall@2, recall@3 and recall@4 which measures the models' ability to retrieve the correct documents.

6.2.3 Accuracy

Language accuracy in bilingual QA evaluates the percentage of responses provided in the correct language. In claim verification, accuracy by sentence is the percentage of an LCLM's ability to correctly identify whether individual statements are true or false when provided with a context and a single sentence as input. Conversely, accuracy by paragraph assesses the percentage of true or false statements within a paragraph that an LCLM can correctly identify when given the context and the entire claim paragraph as input. Finally, for multiple-choice questions (MCQs), accuracy represents the percentage of MCQ samples for which the baseline models generate the correct response.

7 Evaluation

7.1 Multi-document Question Answering

Table 3 summarizes the performance of LCLMs on multi-document QA tasks. As shown in the table, GPT-4o achieved the highest accuracy for entity relationship evaluation in both Arabic and English. Although the entity relationship accuracy of Command-r-plus is higher than four LCLMs, it failed to retrieve the correct document IDs, resulting in lower average recall showing its limitation to accurately trace the source of the retrieved information.

The low average entity relationship accuracy and recall can be attributed to the significant performance decline observed in most models as the number of words increases (see Appendix A.3), highlighting the limited capability of LCLMs when handling increased context lengths or larger numbers of documents. For Arabic, the standard deviation of entity relationship accuracy across word counts ranges from 2.29% to 11.49% across different models, with accuracy generally decreasing as word count increases. A similar trend is observed in English, where standard deviations for entity relationship accuracy across word counts range from 4.72% to 19.36%. Notably, the standard deviations for recall across word count bins are typically much higher than those for entity relationship accuracy, further emphasizing the overall limitations of LCLMs in document tracing, particularly as context length increases.

7.2 Bilingual Question Answering

Table 3 presents the performance of LCLMs on the bilingual QA task. In the table, "Arabic" indicates a scenario where the question is in Arabic, the context is in English, and the answer must be provided in Arabic. Conversely, "English" represents the opposite scenario, where the question is in English, the context is in Arabic, and the answer must be in English. From the table, we observe that Qwen2.5-14B- Instruct-1M model obtained the highest correct language accuracy both for Arabic and English, GPT-4o has the highest entity relationship accuracy. Unlike in multi-document QA, the standard deviations of entity relationship accuracy across different context lengths for Arabic are more stable across most LCLMs, ranging from 0.84% to 8.64% (see Appendix A.4.1). However, Llama-3.3-70B exhibit relatively higher standard deviations ($\geq$ 24%). Both Llama models and Qwen2.5-14B-Instruct, entity relationship accuracy gradually decline as the word count increases.

For English, most LCLMs experience a decline in entity relationship accuracy as the word count increases, with some exceptions in specific word count ranges (see Appendix A.4.2). For instance, GPT-4o shows a decrease in accuracy between 8k and 16k word count range before gradually increasing. The overall standard deviation for varying word counts across LCLMs ranges from 4.56% (Llama-3.3-70B) to 25.21% (Llama-3.1-8B-Instruct), representing significant variability in performance among context length.

7.3 Claim Verification

Table 3 presents the performance of LCLMs on the claim verification task at both the sentence and paragraph levels. From the table, we observe that the accuracy of some LCLMs decreases significantly when claims are presented as paragraphs. However, the opposite scenario is also observed when accuracy by paragraph is higher than the accuracy by sentence. Similar to other evaluation tasks, claim verification demonstrates that LCLMs generally perform better in English compared to Arabic.

The standard deviation across word count bins for LCLMs for Arabic and English are very close to each other ranging from 1.91% to 7.11% (see Appendix A.5). The accuracy of Qwen2.5-14B-Instruct-1M and Phi-3.5-mini-Instruct declines for Arabic as context length increases. Although GPT-

	Multidocument QA				**Bilingual QA**				**Claim Verification**				**MCQ**	
	Arabic		**English**		**Arabic**		**English**		**Arabic**		**English**		**Arabic**	**English**
Model	**Entity Rel Acc**	**Avg Recall**	**Entity Rel Acc**	**Avg Recall**	**Lang Acc**	**Entity Rel Acc**	**Lang Acc**	**Entity Rel Acc**	**Acc. by Sent**	**Acc. by Para**	**Acc. by Sent**	**Acc. by Para**	**Acc.**	**Acc.**
128k Context Length														
GPT-4o	**70.55**	**72.30**	**77.81**	40.53	95.35	**73.47**	97.60	**78.79**	**68.02**	**64.19**	**81.43**	**75.43**	**79.42**	**87.36**
Llama-3.1-8b	48.19	15.95	62.37	41.51	76.94	54.53	97.81	45.10	52.09	54.83	56.08	63.51	49.64	81.25
Llama-3.3-70B	50.76	38.58	61.40	**54.59**	94.96	50.86	99.88	57.31	52.58	59.34	76.17	70.17	71.02	84.49
Qwen2.5-14B	68.88	41.81	70.01	31.47	**97.16**	71.86	**100**	64.36	51.82	35.17	81.29	64.27	74.09	76.95
Command-r -plus-08-2024	50.97	4.82	68.32	5.78	83.03	51.59	84.25	49.61	53.04	55.15	71.93	64.41	64.31	74.32
Phi-3.5-mini	44.51	13.83	68.00	30.44	89.66	43.01	76.83	31.96	52.06	26.04	42.86	64.27	3.60	78.82
32k Context Length														
AceGPT-v2 -32B	42.64	0.42	58.12	7.02	**85.89**	40.57	69.95	33.71	**53.00**	56.28	**76.86**	45.98	52.54	75.64
Qwen2.5-72B	**66.39**	**75.09**	**68.76**	**64.43**	80.85	**77.61**	**93.32**	**68.82**	52.81	**61.92**	76.77	**69.63**	**70.12**	**79.81**

Table 3: Performance of LCLMs for four tasks. Bold value indicates the best performing model.

4o achieves the highest accuracy across all bins for English, it experiences a performance drop with increasing context length.

7.4 Multiple Choice Questions

From table 3, we observe that the general trend of better performance in English compared to Arabic persists in the MCQ task. However, some models, such as Llama-3.1-8B and Phi-3.5-mini, exhibit a substantial disparity in performance between Arabic and English. Overall, all models demonstrated higher accuracy in English MCQ tasks compared to the other three evaluation tasks. In contrast, the results for Arabic MCQs indicate that certain models are significantly undertrained in Arabic compared to English, highlighting a gap in their multilingual capabilities.

The performance across different word count bins showed that some LCLMs exhibit sudden jumps or drops in accuracy for both Arabic and English. Additionally, there is no consistent trend of performance improvement or decline as the word count bins increase, suggesting that the models' behavior varies unpredictably with changes in context length (See Appendix A.6).

7.5 Human Evaluation on Entity Relationship Accuracy

Human evaluators assessed whether entity relationships in gold-standard answers were present in baseline models' responses using 50 randomly selected multi-document QA samples per model. The top-performing models were GPT-4o, Qwen2.5-14B, Command-r-plus-08-2024, and Phi-3.5-mini. Despite differences between entity relationship accuracy calculations and human evaluations, the results showed a strong correlation between the best-performing models for multi-document QA.

8 Conclusion

Our work introduces a new benchmark dataset for long-context English and Arabic, designed to evaluate LCLMs' capabilities in deep reasoning, information extraction, and tracing. This dataset is particularly significant for evaluating long-context Arabic tasks, as, to the best of our knowledge, no dedicated Arabic benchmark currently exists for such evaluations. Since the dataset is human-validated, it ensures high quality and serves as a valuable resource for advancing progress toward Artificial General Intelligence (AGI) in both Arabic and English. The evaluation results across four distinct tasks demonstrate that, although the initial data was generated using GPT-4o, our data creation methodology introduces sufficient complexity to challenge even GPT-4o, preventing it from achieving exceptionally high scores. Notably, in certain tasks in the benchmark, other models outperformed GPT-4o. Moreover, although we evaluated most of the baseline models up to 64k words for Arabic (approximately 128k tokens), `LC-Eval` is capable of evaluating context lengths of up to 256k tokens. This is because it includes data points more than 128k words, and the token fertility rate of Arabic is ≥ 2 for all the baseline models we evaluated. Overall, all LCLMs performed better in English than in Arabic, underscoring the necessity of a benchmark dataset for Arabic to identify and address areas where LCLMs require improvement. Finally, our entity relationship accuracy method for open-ended questions considers semantic meaning, offering a more robust evaluation than existing methods.

9 Limitation

We recognize the following limitations in our work:

1. **Created by GPT-4o**: Since the initial dataset was created using GPT-4o and subsequently human-validated, this may result in a higher evaluation score for GPT-4o compared to other LCLMs, potentially introducing a bias in its favor.

2. **Benchmark Size**: The benchmark size for different word range bins is not large enough to eliminate the effects of randomness in LCLM performance. Future work should focus on increasing the number of samples in each bin to ensure more robust and reliable evaluations.

3. **No Validation on Summaries**: The content of the summaries used to generate the multiple-choice questions (MCQs) and claim paragraphs was not validated. This lack of validation may introduce inaccuracies or inconsistencies in the generated evaluation data.

4. **Domain Distribution**: While the dataset includes multiple domains, it lacks a sufficient number of datapoints for each individual domain. As a result, high performance in a specific task does not necessarily indicate that the LCLM performs well across all domains. Future efforts should aim to improve the domain balance within the dataset.

5. **Human Evaluation on Entity Relationships**: If the human evaluation process for entity relationship existence aligned exactly with the method used to calculate entity relationship accuracy, it would provide a more direct comparison between the human-evaluated approach and the LLM-as-a-judge approach.

6. **No Penalization for Repetition**: We occasionally observe that LCLMs repeat previously generated tokens. Since entity relationship accuracy focuses solely on identifying matching relationships between the gold-standard answer and the generated responses, it does not penalize repetition. As a result, an LCLM can achieve 100% entity relationship accuracy while still repeating its output.

Ethical Considerations

We affirm that all authors of this work are aware of and fully adhere to the ACL Code of Ethics. In developing the datasets presented in this paper, we employed GPT-4o while ensuring alignment with ethical principles. To uphold quality and cultural integrity, all datasets were meticulously validated by human annotators to ensure accuracy and the absence of content conflicting with safety standards or Arab cultural norms. Furthermore, all annotators were fairly compensated based on mutually agreed-upon wage standards and working hours, with all employment arrangements strictly adhering to local regulations.

References

Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, et al. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2023. L-eval: Instituting standardized evaluation for long context language models. *Preprint*, arXiv:2307.11088.

Anthropic. Introducing Claude 3.5 Sonnet — anthropic.com. https://www.anthropic.com/news/claude-3-5-sonnet. [Accessed 29-01-2025].

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. *Preprint*, arXiv:2308.14508.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *Preprint*, arXiv:2412.15204.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan Al-Rashed, Faisal A. Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Al-rubaian, Ali Alammari, Zaki Alawami, Abdulmohsen

Al-Thubaity, Ahmed Abdelali, Jeril Kuriakose, Abdalghani Abujabal, Nora Al-Twairesh, Areeb Alowisheq, and Haidar Khan. 2024. Allam: Large language models for arabic and english. *Preprint*, arXiv:2407.15390.

Egor Bogomolov, Aleksandra Eliseeva, Timur Galimzyanov, Evgeniy Glukhov, Anton Shapkin, Maria Tigina, Yaroslav Golubev, Alexander Kovrigin, Arie van Deursen, Maliheh Izadi, and Timofey Bryksin. 2024. Long code arena: a set of benchmarks for long-context code models. *Preprint*, arXiv:2406.11612.

Cohere. Cohere's Command R+ Model (Details and Application) — Cohere — docs.cohere.com. https://docs.cohere.com/v2/docs/command-r-plus. [Accessed 29-01-2025].

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 166–175, New York, NY, USA. Association for Computing Machinery.

Project Gutenberg. Project Gutenberg — gutenberg.org. https://www.gutenberg.org/. [Accessed 29-01-2025].

Amey Hengle, Prasoon Bajpai, Soham Dan, and Tanmoy Chakraborty. 2024. Multilingual needle in a haystack: Investigating long-context behavior of multilingual large language models. *Preprint*, arXiv:2408.10151.

Hindawi. hindawi.org. https://www.hindawi.org/. [Accessed 29-01-2025].

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? *Preprint*, arXiv:2404.06654.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Song Dingjie, Zhihong Chen, Mosen Alharthi, Bang An, Juncai He, Ziche Liu, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2024. AceGPT, localizing large language models in Arabic. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8139–8163, Mexico City, Mexico. Association for Computational Linguistics.

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. *Preprint*, arXiv:2406.16264.

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? *Preprint*, arXiv:2406.13121.

OpenAI. Models. https://platform.openai.com/docs/models. [Accessed 29-01-2025].

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, Alham Fikri Aji, Zhiqiang Shen, Zhengzhong Liu, Natalia Vassilieva, Joel Hestness, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, Hector Xuguang Ren, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *Preprint*, arXiv:2308.16149.

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

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel Bowman. 2022. Squality: Building a long-document summarization dataset the hard way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1156.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. 2024. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5627–5646, Miami, Florida, USA. Association for Computational Linguistics.

WorldData.info. 2024. Arabic - Worldwide distribution — worlddata.info. https://www.worlddata.info/languages/arabic.php. [Accessed 13-02-2025].

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Jianqing Zhu, Huang Huang, Zhihang Lin, Juhao Liang, Zhengyang Tang, Khalid Almubarak, Mosen Alharthi, Bang An, Juncai He, Xiangbo Wu, Fei Yu, Junying Chen, Zhuoheng Ma, Yuhao Du, Yan Hu, He Zhang, Emad A. Alghamdi, Lian Zhang, Ruoyu Sun, Haizhou Li, Benyou Wang, and Jinchao Xu. 2024. Second language (arabic) acquisition of llms via progressive vocabulary expansion. *arXiv preprint arXiv:2412.12310*.

A Appendix

A.1 Prompts

A.1.1 Multi-document QA Generation

You are a helpful AI assistant tasked with formulating questions and providing detailed, informative answers based on a given text and its most similar texts. I will provide you with a main text and a set of selected similar texts. Your responsibilities are as follows: Generate a question based on the main text and all the selected similar texts. Provide a detailed and informative answer based on all provided texts.
Follow these criteria carefully:

Question Requirements:
- The question must be in Arabic.
- Start the question with REQ:
- The question must be clear and ask about explicit information derived from the provided texts only.
- The question must seek the combined knowledge from the main text and all the selected similar texts.
- The question should encourage a detailed, long, and informative answer.
- Avoid yes/no or overly general questions.
- Handle edge cases (e.g., sparse content in the middle sections) by formulating questions that draw out deeper implications or relationships.

Answer Requirements:
- The answer must be in Arabic.
- Start the answer with RES:
- The answer must be long, detailed, and based **entirely** on the main text and all the selected similar texts.
- Avoid including content from external sources.
- Ensure the answer is long, comprehensive, and strictly relevant to the question.
- Use Markdown formatting sparingly, only to enhance clarity (e.g., for headings or lists).
- Avoid unnecessary formatting for answer text.
- Avoid any external information or overlap with unrelated content.
- Do not acknowledge the provided texts explicitly in the answer.
- Handle edge cases (e.g., sparse content in the middle sections) by drawing out deeper implications or relationships.

Main Text: {line[”text”]}
Selected Similar Texts: {similar_texts_dict}

A.1.2 Multi-document Generated QA Evaluation

I have provided a main text, a set of selected similar texts, a question, and an answer generated based on these texts.
The information in the provided texts can be assumed to be correct. I need you to evaluate the quality of the question and answer together based on the following criteria, and assign a score from 0 to 10 for each criterion, with 0 being very poor and 10 being excellent:

- Accuracy: Do the question and answer accurately reflect the information and ideas in the main text and the selected similar texts?

```
- Completeness: Do the question and answer fully address all aspects of the task,
combining relevant information from the main text and the selected similar texts?
Is any important information missing?

- Clarity: Are the question and answer clear, well-structured, and free of
ambiguity? Is the language used easy to understand?
- Relevance: Do the question and answer stay focused on the provided texts?
Are there any off-topic or irrelevant parts?

- Grammar and Syntax: Are the grammar and syntax of the question and answer
correct? Are there any spelling mistakes or awkward phrasing? Is the sentence
structure smooth and appropriate for the context?

- Consistency: Are the question and answer consistent with the provided texts
and with each other? Do they maintain a coherent narrative throughout?

- Tone and Style: Is the tone appropriate for the context? Do the question
and answer maintain a consistent style and voice throughout?

- Cultural Sensitivity: Do the question and answer respect cultural norms,
particularly with regard to laws, Islam, and Saudi culture? Is there any
content that could be considered offensive or inappropriate?

- Safety: Do the question and answer promote safety and well-being? Is there
any content that could be considered harmful, unsafe, or promoting dangerous
behavior?

After evaluating, provide the scores for each criterion as a dictionary in
the format {{'Criterion': score}}. Ensure that:
- No new line characters (`\\n`) or code block formatting (e.g., ``` or
triple backticks) are used in the dictionary or its representation.

- The evaluation is precise and considers each criterion based solely on the
provided texts, question, and answer.

Main Text: {line[”answer_text”][”main_text”]}
Selected Similar Texts: {line[”answer_text”][”selected_similar_texts”]}
Question: {line[”question”]}
Answer: {line[”answer”]}
```

A.1.3 Bilingual QA Generation

```
You are a helpful AI assistant tasked with formulating open-ended questions and
providing detailed, informative answers based on a given text.
Task Overview:
1. Generate a question based on the main text.
2. Provide a detailed and informative answer based solely on the main text.

Question Requirements:
- The question must be in English.
- Start the question with REQ: .
- The question should be a ”{question_type}” type of question.
- It must focus on information from the **second and third quarters** of the text,
```

```
avoiding content from the beginning or end.
- The question should encourage a thoughtful, detailed, and informative response.
- Avoid yes/no or overly general questions.
- Do not reference or explicitly mention the main text in the question.

Answer Requirements:
- The answer must be in English
- start the answer with RES:.
- It should be accurate, detailed, and based **entirely** on the main text.
- Avoid including content from external sources.
- Ensure the answer is long, comprehensive, and strictly relevant to the uestion.
- Do not reference or acknowledge the main text in the answer.
- Avoid unnecessary formatting for plain text.
Main Text: {[line[”qa_chunk”]]}

%question_type is one of the following: [”How”, ”Why”, ”What”, Where”,”Who”, ”Whos”,
”Whom”, ”When”, ”Which”, ”Describe”, ”Explain”]
```

A.1.4 Bilingual QA Evaluation

```
I have provided the Arabic text and question, and an answer in English generated
based on the text. The information in the provided text can be assumed to be correct.

I need you to evaluate the quality of the answer based on the following criteria, and
assign a score from 0 to 10 for each criterion, with 0 being very poor and 10 being excellent:

Accuracy: Does the answer accurately reflect the information and ideas in the text?
Does the question and answer in English

Completeness: Does the answer fully address all aspects of the question? Is any important
information missing?

Clarity: Is the answer clear and easy to understand? Is the language well-structured and
free of ambiguity?

Relevance: Does the answer stay focused on answering the question based on the text? Are
there any off-topic or irrelevant parts?

Grammar and Syntax: Is the grammar correct? Are there any spelling mistakes or awkward
phrasing? Is the sentence structure smooth and appropriate for the context?

Consistency: Is the answer consistent with the provided text and with itself? Does it
maintain a coherent narrative throughout?

Tone and Style: Is the tone appropriate for the context? Does the answer maintain a
consistent style and voice throughout?

Cultural Sensitivity: Does the answer respect cultural norms, particularly with regard to
laws, Islam, and Saudi culture? Is there any content that could be considered offensive
or inappropriate?

Safety: Does the answer promote safety and well-being? Is there any content that could be
considered harmful, unsafe, or promoting dangerous behavior?
```

````
After evaluating, provide the scores for each criterion as a dictionary in the
format {{'Criterion': score}}.

Do not use new line ”\n” or ”```” in the dictionary or any identification of the data type shape.

Text: {[line[”qa_chunk”]]}
Question: {line[”question”]}
Answer: {line[”answer”]}
````

A.1.5 Bilingual QA segment identification

```
You are a helpful AI assistant tasked with pinpointing the exact paragraphs in a
long Arabic text that support a given English answer. I will provide:
An English question (derived from the Arabic text), An English answer
(based on that same text) and A long Arabic text (the source for both the
question and the answer).

Your task is to identify all paragraphs in the Arabic text where this answer
is found or supported. When presenting these paragraphs, you must provide them
exactly as they appear—with no edits, changes, or additions to the original text.

English question: {[line[”question”]]}
English answer: {[line[”answer”]]}
Arabic text: {[line[”qa_chunk”]]}
```

A.1.6 Summary Generation

```
You are an excellent writing assistant. I will give you a chunk to summarize.
I will also provide you with the text I wrote for the previous (n-1) chunk.
Please help me continue writing the summarization to the next chunk based on
the chunk to summarize, and the already written text.
Make sure the summarization is detailed and contains key information.
If already written text is empty, it will be indicated by ”” and summarize the
chunk to summarize as the first chunk.

Requirements for summarization:
    1. Cover all main points
    2. Keep information on elements that may be important for future chunks
    3. Create a comprehensive summary that can be built upon
    5. The summary should not be in bullet points or numbers but in paragraphs
    6. The summary should be 5%-10% of the provided chunk to summarize.
   7. Exclude any irrelevant information to the summary, such as chapter information,
    Patent information, chapter name, headlines, author name and copyrights.
    8. If already_written_text is empty, start with introductory paragraph.


chunk to summarize: {chunk_to_summarize}

Already written text:  {already_written_text}
```

```
Please integrate the already written text to the new summary, and now continue
writing the summary for the next chunk.


Make sure that the chunk to summarize is coherent with the already written text. If
already written text is empty, do not add anything before the summary.
Do not make any changes to the already written text and continue the
summarization as if it is the continuation of already written text.
Include already written text in the summary.

Output in the following json format:
{
    Summary: <<Summary>>
}

Replace <<Summary>> with the generated summary.
```

A.1.7 MCQ Generation

```
You are an expert in generating High-Difficulty Multiple-Choice Questions (MCQs).
Based on the passage provided below, you need to generate a well-formed MCQ.
Please follow the format exactly as described:

Requirements:
1. **Question Design**:
- Each question must integrate information from **multiple, non-adjacent parts**
of the passage.
- Questions should emphasize **critical analysis**, requiring the reader to
interpret relationships, infer meaning, or synthesize ideas.
- Questions should have a **difficulty level of 90-100** on a scale of 0-100.
- **90-94**: Challenging, requiring detailed understanding and connection of ideas.
- **95-97**: Very challenging, demanding integration of complex concepts and
nuanced reasoning.
- **98-100**: Extremely challenging, involving deep interpretation and synthesis
of intricate details.

2. **Answer Options**:
- Provide **4 options per question**, with a single correct answer.
- Distractors (incorrect choices) can be partially correct and very close to
the correct answer, making the question more difficult.
- Each option’s **difficulty level** should be between **95-100**.


3. **Output Format**:
- Use **JSON** to structure the output.
- For each question, include the following fields:
- `question`: The text of the question.
- `difficulty_level_of_the_question`: The difficulty level of the
question (90-100).
- `choices`: An array of 4 plausible answer options, each written
clearly and precisely.
- `correct_answer`: The number corresponding to the correct choice (1-4).
- `difficulty_level_of_the_choices`: The difficulty level of the answer
```

```
options (95-100).

4. **Balance of Difficulty**:
- At least **2 questions** must have a difficulty level of **98-100**.
- At least **2 questions** must have a difficulty level of **95-97**.


Here is the passage to generate the MCQ from:

{chunk_to_process}

**Expected output format:**

{
    "question": "<question_text>",
    "difficulty_level_of_the_question": <difficulty_level_of_the_question>,
    "choices": [
        "Answer 1: <answer_1>",
        "Answer 2: <answer_2>",
        "Answer 3: <answer_3>",
        "Answer 4: <answer_4>"
    ],
    "correct_answer": <correct_answer_index>,
    "difficulty_level_of_the_answers": <difficulty_level_of_the_answers>
}

Make sure the output is structured exactly as shown above.
The question should be based on the passage, and the answers
should be plausible but distinct.
```

A.1.8 Claim Verification

```
You are an excellent claim writer.
Your task is to create a 5 sentence paragraph with
claims from a given passage. In the paragraph, some claims could be true
and some could be false. The paragraph should have coherence and be
challenging for even an expert reader to judge the truth of each claim.

**Requirements for creating the paragraph from the passage:**
1. The paragraph should contain at most 5 sentences. Sentences should
not be unnecessarily long.
2. Each sentence should contain a claim that can be either true or false.
3. The false claim should contain partially true statement to make it
more difficult to ideintify.
4. Do not include any external entities or external relationships in
any of the claims.
5. The difficulty level of each claim should be between 97-100 out
of 1-100. 97-100 signifies extremely difficult.

**Output Format:**
- A paragraph containing the claims.
- A breakdown specifying which claims are true and which are false.
- A corrected version of the paragraph where all claims are accurate.
```

~~~
**Passage:**
[Insert passage here]

**Output Example:**
```json
{
  "claims_paragraph": <created claim paragraph>,
  "true_claims": [
    "<true claim 1>",
    "<true claim 2>"
  ],
  "false_claims": [
    "<false claim 1>",
    "<false claim 2>",
  ],
  "corrected_paragraph": <corrected Paragraph>"
}
```

Replace <...> in the Output example json with generated true claims,
false claims and corrected paragraph.
~~~

A.1.9 Entity Relationships

```
You will be given a text: {input_text}

Your task is to identify all entities in each line and their relationships.
Include people, organizations, locations, dates, numerical values, and any
other relevant entities. Relationship means how these entities are
connected to each other.

Instructions:
1. Identify all entities for each sentence.
2. Map all relationships between connected entities for each sentence.
3. Express the relationship between entities with at most 3 words.
4. Break multiple relationships into smaller relationships.
5. When identifying relationships, consider only two entities at a time.
6. Avoid duplicates and ensure each entity and relationship pair appears only once.


JSON Output:

Map the relationships for each text in the following JSON format strictly:

json
{
     [
    {"entity_1": "Entity A", "relationship": "Relation", "entity_2": "Entity B"},
    {"entity_1": "Entity C", "relationship": "Relation", "entity_2": "Entity D"},
     ...

      ]
```

```
}
Strictly follow this JSON structure.
Do not generate any additional text outside of JSON.
Do not leave any entities or their relationships unrecorded.
```

A.2 Evaluation Samples

A.2.1 Multi-document QA Example

[1] distractor document-1
.....
[16] China reported 41 new coronavirus (COVID-19) cases on Tuesday. China's National Health Commission stated that the total number of COVID-19 cases reached 95,851, while the total deaths remain at 4,636. Beijing, September 22, 2021.
.....
[35] China reported 200 new coronavirus (COVID-19) cases. China's National Health Commission stated that the total number of COVID-19 cases reached 101,277, while the total deaths remain at 4,636. Beijing, December 27, 2021.
.....
[37] China reported 207 new coronavirus (COVID-19) cases. China's National Health Commission stated that the total number of COVID-19 cases reached 101,890, while the total deaths remain at 4,636. Beijing, December 30, 2021.
.....
[n] distractor document-n

Question:
How has the number of COVID-19 cases and deaths in China evolved from September 2021 to the end of December 2021?

A.2.2 Bilingual QA Example

(Long text about architectural inspired design in Arabic Language here)
....
Question:
Where can you find architectural elements inspired by Islamic design in Oxford and Cambridge?

A.2.3 Claim Verification Example

(Long English document here)
....

Claims paragraph:

Contributors who are most receptive to suggestions are always the ones who can be trusted to work independently. Editors strive to minimize restrictions on contributors once they are confident in their abilities because writers perform best when passionate about their work. Modern magazines have shifted towards relying more on new and unknown contributors, providing a platform for aspiring writers. The tradition of editing has remained unchanged over time, with editors being the first and often the most critical reviewers of a contributor's work. Contributors should focus on producing true and beautiful work, as editors appreciate quality submissions and are more likely to support consistent contributors.

A.2.4 Multiple Choice Question Example

(Long English document here)
....
Question:
How did the introduction of Arabic numerals and algebra by oriental scholars in Europe impact the curriculum, according to the text?

Options:
A) It led to the inclusion of practical subjects like financial training in the curriculum.
B) It revolutionized mathematical calculations, making arithmetic and algebra more practical and accessible.
C) It resulted in the early introduction of geometry in lower grades to develop spatial understanding.
D) It caused the curriculum to heavily emphasize traditional literary subjects over practical applications.

A.3 Detailed Evalution of Multi-document QA

A.3.1 Arabic

Model	Metric	4k-8k	8k-16k	16k-32k	32k-64k	Avg	Std
128k Context Length							
GPT-4o	**RougeL**	49.38	49.7	47.09	45.17	47.835	2.123
	Avg Recall	77.40	76.98	69.76	65.07	72.30	5.96
	Entity Rel Acc	74.91	67.54	69.93	69.82	70.55	3.10
Llama-3.1-8B	**RougeL**	17.9	15.94	14.32	11.9	15.015	2.54
	Avg Recall	40.28	14.26	7.39	1.88	15.95	16.98
	Entity Rel Acc	59.47	48.61	43.92	40.76	48.19	8.18
Llama-3.3-70B	**RougeL**	17.9	16.68	15.28	10.75	15.15	3.12
	Avg Recall	69.83	47.37	37.08	0.04	38.58	29.10
	Entity Rel Acc	56.79	57.98	57.52	30.75	50.76	13.34
Qwen2.5-14B	**RougeL**	36.82	35.38	36.51	37.1	36.45	0.75
	Avg Recall	68.58	49.04	32.37	17.26	41.81	22.06
	Entity Rel Acc	71.73	69.75	67.22	66.83	68.8825	2.29
Command-r -plus-08-2024	**RougeL**	24.58	25.19	22.74	17.29	22.45	3.59
	Avg Recall	18.08	0.43	0.01	0.78	4.82	8.84
	Entity Rel Acc	59.63	51.63	51.7	40.94	50.975	7.67
Phi-3.5-mini	**RougeL**	25.17	27.48	28.4	-	27.01	1.66
	Avg Recall	36.18	5.24	0.08	-	13.83	19.52
	Entity Rel Acc	46.06	45.65	41.83	-	44.51	2.33
32k Context Length							
AceGPT-v2 -32B	**RougeL**	13.8	9.86	-	-	11.83	2.78
	Avg Recall	0.84	0	-	-	0.42	0.59
	Entity Rel Acc	50.76	34.52	-	-	42.64	11.48
Qwen2.5-72B	**RougeL**	18.41	26.4	-	-	22.405	5.64
	Avg Recall	83.27	66.91	-	-	75.09	11.57
	Entity Rel Acc	65.93	66.85	-	-	66.39	0.65

Table 4: Performance of Arabic language in multi-document QA.

A.3.2 English

Model	Metric	4k-8k	8k-16k	16k-32k	32k-64k	64k-128k	Avg	Std
128k Context Length								
GPT-4o	**RougeL**	38.92	36.61	35.84	35.69	34.69	36.35	1.59
	Avg Recall	56.37	43.79	42.25	36.83	23.41	40.53	11.95
	Entity Rel Acc	85.1	75.54	69.48	81.44	77.52	77.81	5.93
Llama-3.1-8B	**RougeL**	21.96	21.77	21.17	19.91	19.27	20.82	1.17
	Avg Recall	81.71	68.52	45.38	11.61	0.36	41.52	35.17
	Entity Rel Acc	79.23	63.5	55.03	63.9	50.23	62.38	11.05
Llama-3.3-70B	**RougeL**	21.24	20.99	20.68	20.12	19.26	20.46	0.78
	Avg Recall	84.02	72.16	62.39	42.92	11.45	54.59	28.42
	Entity Rel Acc	75.77	60.94	57.5	54.4	58.4	61.40	8.36
Qwen2.5-14B	**RougeL**	29.05	28.05	28.96	29.07	29.6	28.95	0.56
	Avg Recall	62.32	41.20	29.71	17.74	6.37	31.47	21.60
	Entity Rel Acc	82.47	66.1	59.88	69.07	72.53	70.01	8.37
Command-r-plus-08-2024	**RougeL**	28.03	24.98	23.03	22.47	20.97	23.89	2.72
	Avg Recall	16.26	7.17	4.52	0.64	0.04	5.72	6.57
	Entity Rel Acc	80.07	70.64	66.64	66.39	57.89	68.32	8.04
Phi-3.5-mini	**RougeL**	24.72	23.84	23.85	24.46	22.68	23.91	0.70
	Avg Recall	61.43	52.17	29.43	7.28	1.89	30.44	26.38
	Entity Rel Acc	74.74	68.92	68.92	66.7	60.76	68.00	5.03
32k Context Length								
AceGPT-v2-32B	**RougleL**	19.92	19.17	18.62	-	-	19.23	0.65
	Avg Recall	13.58	4.62	2.86	-	-	7.02	5.74
	Entity Rel Acc	63.58	55.4	55.39	-	-	58.12	4.72
Qwen2.5-72B	**RougeL**	31.48	29.56	28.15	-	-	29.73	1.67
	Avg Recall	74.99	70.48	47.84	-	-	64.43	14.55
	Entity Rel Acc	85.07	73.87	47.36	-	-	68.76	19.36

Table 5: Performance for English in multi-document QA.

A.4 Bilingual Question Answer

A.4.1 Arabic

Model	Metric	4k-8k	8k-16k	16k-32k	32k-64k	Avg	Std
128k Context Length							
GPT-4o	**RougeL**	41.79	36.57	35.67	35.44	37.37	2.98
	Lang Acc	99.03	100	83.4	99	95.35	7.98
	Entity Rel Acc	76.85	73.25	69.15	74.65	73.47	3.24
Llama-3.1-8B	**RougeL**	9.84	9.33	9.26	8.06	9.12	0.75
	Lang Acc	74.55	75.19	73.63	84.4	76.94	5.01
	Entity Rel Acc	64.31	52.33	57.65	43.85	54.54	8.64
Llama-3.3-70B	**RougeL**	19.59	19.83	12.23	13.54	16.29	3.97
	Lang Acc	95.94	95.66	92.72	95.55	94.97	1.50
	Entity Rel Acc	65.52	61.23	62.42	14.28	50.86	24.45
Qwen2.5-14B	**RougeL**	36.72	43.19	34.96	44.53	39.85	4.71
	Lang Acc	96.72	98.8	95.37	97.78	97.16	1.46
	Entity Rel Acc	77.11	71.32	69.85	69.17	71.86	3.61
Command-r-plus-08-2024	**RougeL**	20.98	18.4	10.46	9.63	14.86	5.67
	Lang Acc	86.63	77.78	83.63	84.1	83.04	3.74
	Entity Rel Acc	52.12	53.12	49.45	51.67	51.59	1.55
Phi-3.5-mini	**RougeL**	16.09	20.64	21.46	-	19.39	2.89
	Lang Acc	88.52	90.2	94.59	-	91.10	3.13
	Entity Rel Acc	44.8	31.43	45.13	-	40.45	7.81
32k Context Length							
AceGPT-v2-32B	**RougeL**	6.46	6.11	5.33	-	5.96	0.57
	Lang Acc	92.75	87.16	77.78	-	85.89	7.56
	Entity Rel Acc	41.5	39.86	40.35	-	40.57	0.84
Qwen2.5-72B	**RougeL**	19.62	31.49	11.64	-	20.91	9.98
	Lang Acc	91.69	91.63	59.25	-	80.85	18.71
	Entity Rel Acc	78.86	79.31	74.66	-	77.61	2.56

Table 6: Performance of Arabic language in bilingual QA.

A.4.2 English

Model	Metric	4k-8k	8k-16k	16k-32k	32k-64k	Avg	Std
128k Context Length							
GPT-4o	**RougeL**	41.57	47.8	39.21	44.69	43.32	3.73
	Lang Acc	99.4	99.2	96.33	95.5	97.61	1.99
	Entity Rel Acc	74.08	67.13	83.61	90.36	78.79	10.25
Llama-3.1-8B	**RougeL**	17.23	15.7	13.29	9.31	13.88	3.45
	Lang Acc	95.44	95.83	100	100	97.81	2.52
	Entity Rel Acc	65.08	63.37	40.87	11.11	45.11	25.21
Llama-3.3-70B	**RougeL**	32.55	28.48	29.93	30.81	30.44	1.70
	Lang Acc	99.55	100	100	100	99.89	0.23
	Entity Rel Acc	61.04	60.34	56.82	51.05	57.31	4.56
Qwen2.5-14B	**RougeL**	48.6	44.1	44.51	39.69	44.26	3.64
	Lang Acc	100	100	100	100	100	0
	Entity Rel Acc	68.83	69.55	63.51	55.56	64.36	6.45
Command-r-plus-08-2024	**RougeL**	30.87	17.84	11.51	8.36	17.14	9.96
	Lang Acc	81.25	85.33	82.34	88.1	83.035	3.74
	Entity Rel Acc	71.86	63.4	40.98	22.2	84.25	22.44
Phi-3.5-mini	**RougeL**	29.36	25.37	25.18	-	26.63	2.36
	Lang Acc	71.42	86.36	72.71	-	76.83	8.27
	Entity Rel Acc	43.29	41.5	11.11	-	31.96	18.08
32k Context Length							
AceGPT-v2-32B	**RougeL**	17.23	16.21	10.58	-	14.67	3.58
	Lang Acc	83.33	45.57	80.95	-	69.95	21.14
	Entity Rel Acc	42.14	33.61	25.38	-	33.71	8.38
Qwen2.5-72B	**RougeL**	38.78	29.26	-	-	34.02	6.73
	Lang Acc	99.5	87.14	-	-	93.32	8.73
	Entity Rel Acc	70.34	67.31	-	-	68.82	2.14

Table 7: Performance of English language in bilingual QA.

A.5 Claim Verification

A.5.1 Arabic

Model	Metric	4k-8k	8k-16k	16k-32k	32k-64k	Avg	Std
128k Context Length							
GPT-4o	**Acc by Paragraph**	61.28	62.22	67.82	65.46	64.19	3.00
	Acc by Sentence	63.85	67.6	69.28	71.38	68.02	3.18
Llama-3.1-8B	**Acc by Paragraph**	54.23	59.14	56.13	49.85	54.83	3.89
	Acc by Sentence	56.25	50.72	53.67	47.75	52.09	3.67
Llama-3.3-70B	**Acc by Paragraph**	58.36	58	63.49	57.52	59.83	3.89
	Acc by Sentence	56.52	48.99	52.25	52.58	52.09	3.67
Qwen2.5-14B	**Acc by Paragraph**	36.3	42.74	36.09	25.56	35.17	7.11
	Acc by Sentence	56.52	51.35	53.37	50	52.81	2.83
Command-r-plus-08-2024	**Acc by Paragraph**	52.42	56.42	59.55	52.21	55.15	3.51
	Acc by Sentence	56.25	50	53.25	52.69	53.04	2.56
Phi-3.5-mini	**Acc by Paragraph**	33.67	26.67	17.78	-	26.04	7.96
	Acc by Sentence	55.18	51.01	50	-	52.06	2.74
32k Context Length							
AceGPT-v2-32B	**Acc by Paragraph**	54.87	57.69	-	-	56.28	1.99
	Acc by Sentence	54.05	51.95	-	-	53	1.48
Qwen2.5-72B	**Acc by Paragraph**	59.31	64.31	-	-	61.81	3.53
	Acc by Sentence	56.52	51.35	-	-	53.93	3.65

Table 8: Performance of Arabic language in Claim Verification Task.

A.5.2 English

Model	Metric	4k-8k	8k-16k	16k-32k	32k-64k	64k-128k	Avg	Std
128k Context Length								
GPT-4o	**Acc by Paragraph**	77.74	76.06	68.54	68.17	86.67	75.43	7.61
	Acc by Sentence	82.87	82.98	82.33	79.11	79.86	81.43	1.81
Llama-3.1-8B	**Acc by Paragraph**	69.82	64.27	60.39	62	61.11	63.51	3.81
	Acc by Sentence	65.05	54.51	55.8	48.15	56.92	56.08	6.05
Llama-3.3-70B	**Acc by Paragraph**	71.92	72.66	65.57	67.39	73.33	70.17	3.46
	Acc by Sentence	79.89	79.37	76.97	69.62	75	76.17	4.15
Qwen2.5-14B	**Acc by Paragraph**	67.04	63.8	61.67	64.4	64.44	64.27	1.91
	Acc by Sentence	81.43	79.43	82.74	83.75	79.11	81.29	2.02
Command-r -plus-08-2024	**Acc by Paragraph**	68.67	65.87	62.16	62.73	62.67	64.27	3.43
	Acc by Sentence	72.53	72.02	76.05	66.67	72.41	71.93	3.36
Phi-3.5-mini	**Acc by Paragraph**	70.36	63.47	62.16	62.73	62.67	64.27	3.43
	Acc by Sentence	41.28	41.28	40.66	35.37	49.12	42.86	5.65
32k Context Length								
AceGPT-v2 -32B	**Acc by Paragraph**	41.01	45.05	51.88	-	-	45.98	5.49
	Acc by Sentence	78.24	74.74	77.61	-	-	76.86	1.86
Qwen2.5-72B	**Acc by Paragraph**	74.11	72.11	63.13	-	-	69.78	5.84
	Acc by Sentence	77.17	74.45	75.43	-	-	75.68	1.37

Table 9: Performance of English language in Claim Verification Task.

A.6 Multiple Choice Question

A.6.1 Arabic

Model	**Metric**	**4k-8k**	**8k-16k**	**16k-32k**	**32k-64k**	**Avg**	**Std**
128k Context Length							
GPT-4o	**Accuracy**	76.04	79.04	78.05	84.54	79.42	3.63
Llama-3.1-8B		50	47.79	56.70	44.09	49.64	5.30
Llama-3.3-70B		72.91	73.16	74.39	63.63	71.02	4.96
Qwen2.5-14B		73.95	72.42	73.17	76.81	74.09	1.92
Command-r-plus-08-2024		65.62	70.95	61.58	59.09	64.31	5.18
Phi-3.5-mini		0.62	0.14	0.67	0	3.60	3.37
32k Context Length							
AceGPT-v2-32B	**Accuracy**	57.29	47.79	-	-	52.54	6.71
Qwen2.5-72B		71.87	68.38	-	-	70.12	2.46

Table 10: Performance of Arabic MCQs Task.

A.6.2 English

Model	**Metric**	**4k-8k**	**8k-16k**	**16k-32k**	**32k-64k**	**64k-128k**	**Avg**	**Std**
128k Context Length								
GPT-4o	**Accuracy**	83.51	91.82	86.76	89	32.25	76.67	25.01
Llama-3.1-8B		75.53	86.36	81.86	78	34.67	71.28	20.87
Llama-3.3-70B		78.72	90.45	84.31	84.5	34.27	74.45	22.84
Qwen2.5-14B		75	81.36	74.50	72	65.72	73.71	5.64
Command-r-plus-08-2024		73.93	79.54	65.68	68	25.40	62.51	21.43
Phi-3.5-mini		69.68	85.90	80.88	77	22.98	67.29	25.46
32k Context Length								
AceGPT-v2-32B	**Accuracy**	74.46	76.81	36.76	-	-	47.51	22.47
Qwen2.5-72B		75.53	84.09	40.19	-	-	66.60	23.26

Table 11: Performance of English MCQs Task.