bnContextQA: Benchmarking Long-Context Question Answering and Challenges in Bangla

Adnan Ahmad[1], Labiba Adiba[1], Namirah Rasul[1]*
Md Tahmid Rahman Laskar[2], Sabbir Ahmed[1]
[1]Islamic University of Technology, [2]York University
[1]{adnanahmad, labibaadiba, namirahrasul, sabbirahmed}@iut-dhaka.edu
[2]tahmid20@yorku.ca

Abstract

Large models have advanced in processing long input sequences, but their ability to consistently use information across extended contexts remains a challenge. Recent studies highlight a positional bias where models prioritize information at the beginning or end of the input while neglecting the middle, resulting in a U-shaped performance curve, but this was limited to English. Whether this bias is universal or shaped by language-specific factors remains unclear. In this work, we investigate positional bias in Bangla, a widely spoken but computationally underrepresented language. To support this, we introduce a novel Bangla benchmark dataset, 'bnContextQA', specifically designed for long-context comprehension. The dataset comprises of 350 long-context QA instances, each paired with 30 context paragraphs, allowing controlled evaluation of information retrieval at different positions. Using this dataset, we assess the performance of LLMs on Bangla across varying passage positions, providing insights into cross-linguistic positional effects. The `bnContextQA` dataset is publicly available at `https://github.com/labiba02/bnContextQA.git` to support future research on long-context understanding in Bangla and multilingual LLMs.

1 Introduction

Large language models are increasingly capable of processing long sequences, with context lengths extending to tens of thousands of tokens (Chang et al., 2024). This capability is crucial for real-world applications, including question answering, summarization, and retrieval-augmented generation (Zheng et al., 2025; Laskar et al., 2023, 2024). However, the assumption that models can robustly use all available context is being challenged by recent findings (Li et al., 2024a).

The Lost in the Middle study(Liu et al., 2023) highlights a striking limitation: LLMs tend to prioritize information located at the beginning (primacy bias) or end (recency bias) of their input, while struggling to retrieve and apply information positioned in the middle. This U-shaped performance curve calls into question the practical utility of extended context lengths, since critical details in real documents are not always conveniently placed.

Despite these findings, prior research has been restricted to English. Given the syntactic and morphological differences in languages like Bangla, it is vital to investigate if the same biases persist across multilingual contexts.

Bangla, being one of the most widely spoken languages in the world, remains underexplored in the evaluation of LLMs (Kabir et al., 2024; Mahfuz et al., 2025; Abrar et al., 2024). Unlike English, Bangla has complex morphology and flexible word order, which may interact differently with model architectures when processing long contexts. In this work, we address this gap by:

1. Constructing a Bangla long-context QA dataset with 350 questions and 30 passages per question.
2. Running baseline evaluations with two state-of-the-art generative QA models, GPT-4.1 (OpenAI) and Gemini 2.5 Flash Lite (Google)
3. Presenting early evidence of positional bias in Bangla.

Our preliminary results show that, similar to English, LLMs also struggle with middle-position evidence in Bangla. These findings motivate further work on long-context modeling, dataset expansion, and evaluation of Bangla LLMs.

2 Literature Review

Recent progress in the advanced language models has excelled across a broad spectrum of natural lan-

* Authors 1,2,3 contributed equally.

Proceedings of the Second Workshop on Bangla Language Processing (BLP-2025), pages 357–365

guage tasks, enabling them to adapt to diverse linguistic settings (Mahbub et al., 2023; Ahmed et al., 2024; Khan et al., 2023a,b; Arif et al., 2025). As their capabilities expand, researchers have increasingly turned their attention to how such models perform in long-context processing, where models must maintain coherence, track dispersed information, and reliably retrieve details embedded across lengthy sequences (Li et al., 2024a; Huang et al., 2024; Liu et al., 2025).

Prior works on long-context processing have primarily focused on English (Bai et al., 2024; Li et al., 2024b; Bai et al., 2025; Gao et al., 2025). A key study, Lost in the Middle: How Language Models Use Long Contexts (Liu et al., 2023), systematically examined model behavior on extended inputs, showing that accuracy drops for middle-position evidence. However, it relied solely on English datasets, leaving open whether these positional effects generalize to other languages.

Liu et al. (2023) evaluated positional sensitivity using two tasks:

- **Multi-document question answering**: Models received multiple documents with only one containing the correct answer, using the NaturalQuestions-Open (NQ-Open) dataset (Lee et al., 2019). The gold document's position was varied to measure robustness.
- **Key–value retrieval**: A synthetic benchmark where models selected the correct value from a JSON object, allowing position to be manipulated independently of natural language semantics.

Across both settings, performance followed a U-shaped curve, with strong primacy and recency effects and significantly weaker retrieval in the middle.

While these findings demonstrate positional bias, existing resources offer little insight into how this manifests in multilingual or low-resource settings. In Bangla, current reading-comprehension datasets rely on single-document passages and therefore cannot evaluate long-context reasoning or positional effects. The most notable dataset, BanglaRQA (Ekram et al., 2022), provides 3000 passages and 14,889 question–answer pairs with diverse question types and answer formats, but lacks multi-document inputs, distractors, and controlled evidence placement. This gap motivates the creation of bnContextQA, a benchmark designed specifically for long-context comprehension in Bangla. By pairing each question with 30 semantically related passages, including curated distractors, and precisely controlling the gold passage's position, our dataset enables systematic analysis of positional bias and extends long-context QA research beyond English.

3 bnContextQA

In this section, we explain the construction of our long-context Bangla QA dataset. We describe how passages were collected and curated from Bangla Wikipedia, how distractor passages were carefully designed to be topically similar yet unanswerable, and the preprocessing steps applied to ensure data quality. An example of a dataset instance and summary statistics are provided in Appendix A.1.

3.1 Data Acquisition

To study the effect of long-context input on Bangla LLMs, we constructed a Bangla long-context QA dataset simulating multi-document question answering. Existing Bangla QA datasets, such as Bengali-SQuAD (Tahsin Mayeesha et al., 2021), SQuAD_Bn (Bhattacharjee et al., 2022), and BanglaQA (Shahriar et al., 2023), mainly contain short passages, limiting systematic evaluation on extended contexts. Our dataset includes multiple passages per question, with one gold passage containing the answer and several semantically related distractors.

We used Bangla Wikipedia as the primary source for its broad coverage across domains. Passages were manually curated to ensure correctness, quality, and domain diversity, and to control semantic similarity for realistic distractors—beyond what automatic extraction can reliably achieve. This careful construction ensures the QA task requires genuine reasoning rather than superficial keyword matching, providing a robust benchmark for long-context comprehension in LLMs.

3.2 Dataset Structure

Each sample in our dataset is represented as a JSON object with the following components:

- **Question** (`question`): A natural language query in Bangla.
- **Language** (`language`): Fixed as "bn" to indicate Bengali.
- **Documents** (`documents`): A list of 30 passages, each containing a title, content, and

source. One passage contains the gold evidence for the correct answer. The remaining passages serve as distractors, deliberately chosen to share topical or lexical similarity with the gold passage, increasing task difficulty. Each passage is of 175 token on average.

- **Answer** (`answer`): The ground-truth answer derived from the relevant document.
- **Relevant Document Index**(`relevant_document_index`): The index pointing to the passage containing the gold evidence.
- **Context Length** (`context_length`): The number of passages provided per instance (fixed at 30 in our dataset).
- **Metadata** (`metadata`): Includes dataset source, retrieval method, and special notes.

3.3 Question Generation

To create queries for multi-document evaluation, we selected topics from Banglapedia and Wikibangla that contain many closely related articles, ensuring sufficient semantic overlap between gold passages and distractors. Using these topics as prompts, we generated candidate questions with ChatGPT and then manually filtered them to ensure short, unambiguous answers that were not easily guessable from the gold passage's wording or position. Throughout the process, prompts were crafted carefully so the model could not rely on prior knowledge but had to use the provided gold passage, ensuring alignment with the objectives of evaluating long-context reasoning.

3.4 Distractor Design

A key feature of our dataset is the careful construction of distractor passages. Instead of random text, distractors were selected from Bangla Wikipedia and Banglapedia to ensure topical and stylistic alignment with the gold passage.

To keep this process systematic, native Bangla speakers applied a structured manual filtering procedure. Candidate distractors were evaluated according to:

- **Topical relevance:** The distractor must fall within the same broad domain as the gold passage to maintain thematic coherence (e.g., historical sites, political events, scientific topics).
- **Lexical similarity:** Passages were selected to share key vocabulary, technical terms, or stylistic features with the gold passage, preventing models from relying on simple keyword matching.
- **Factual distinction:** Distractors were checked to ensure they contain no answer-bearing text or paraphrases that could accidentally reveal the correct answer.
- **Structural parity:** Distractors were matched to the gold passage in length, complexity, and informational density, preventing models from exploiting superficial cues such as unusually short, long, or structurally simple passages.

This human-guided design produces distractors that are plausible, challenging, and semantically aligned, resulting in a robust evaluation setting for long-context comprehension in Bangla LLMs.

3.5 Preprocessing and Cleaning

During passage collection, raw Wikipedia text often contained nuanced references, extraneous symbols, or English phrases that could bias results. To minimize such noise, annotators were instructed to remove redundant citations and bracketed references, normalize the Bangla script to a consistent Unicode form, eliminate repetitive English words unless essential to factual content (translating necessary terms into Bangla), and standardize passage lengths to ensure comparability. Additional metadata such as topic category, article identifiers, and the specified gold position was added to each item. This produced a dataset that is linguistically coherent, semantically consistent, and suitable for evaluating long-context reasoning in Bangla.

4 Experimental Details

4.1 Models

In this section, we describe the models that we evaluate on our proposed Bangla dataset. We conducted our experiments using Gemini-2.5-Flash-Lite (Google, 2024) and GPT-4.1-Nano (OpenAI, 2024) (details about the models are given in Appendix A.2). We selected them for their cost-effectiveness and accessibility. Gemini-2.5-Flash-Lite offers high-throughput processing at low cost while maintaining strong reasoning capabilities. GPT-4.1-Nano provides efficient performance with extensive context windows. These

attributes make both models particularly suitable for deployment in Bangla-speaking regions where computational resources are often constrained. By focusing on these models, our work evaluates whether affordable LLMs can still maintain competitive performance on long-context tasks in low-resource languages.

Although large generative models like TituLLM (Nahin et al., 2025) and BongLLaMA (Zehady et al., 2024) are trained on Bangla language and support extended contexts, we did not use them in this study. In our preliminary trials, we observed that these models tended to overgenerate or rely heavily on parametric knowledge rather than grounding their answers in the provided passages. Since our evaluation requires short, span-based answers to measure sensitivity to positional placement, such behavior makes them less suitable for this specific task.

4.2 Evaluation Method

Each model was evaluated across all context lengths and gold passage positions. Models predicted the answer span from one gold passage and multiple semantically related distractors, increasing task difficulty and requiring fine-grained reasoning.

To assess positional effects, the gold passage was placed at multiple locations across different context lengths: positions 1, 3, and 5 for length 5; 1, 3, 5, 7, and 10 for length 10; 1, 5, 10, 15, and 20 for length 20; and 1, 5, 10, 15, 20, 25, and 30 for length 30. These placements capture early, middle, and late positions, enabling analysis of primacy and recency effects as well as mid-sequence degradation. Combined with controlled context variation and carefully matched distractors, this setup provides a robust framework for evaluating Bangla QA models' long-context reasoning and extractive accuracy. Details on evaluation metrics and implementation are provided in Appendices A.3 and A.4.

5 Results

We evaluated two generative question answering models, `GPT-4.1-nano` and `Gemini 2.5-flash-lite` on our Bangla QA dataset with input contexts containing 10, 20, and 30 total passages, each containing one gold passage and several distractors.

The results in Table 1 demonstrate a clear positional effect on model performance for both GPT-4.1-nano and Gemini-2.5-Flash-Lite. For GPT-4.1-nano, the highest scores are observed when the relevant context appears at the beginning (Index 1), with 60.63% Exact Match (EM) and 67.40% F1, followed by a sharp decline in the middle positions (Indices 3 and 5) and a slight recovery at Index 10 (41.59% EM, 48.00% F1). Gemini-2.5-Flash-Lite shows a similar U-shaped trend but consistently achieves higher overall scores, with EM and F1 peaking at 61.59% and 76.32% at Index 1, gradually decreasing toward the middle, and partially recovering at Index 10 (54.29% EM, 71.30% F1). This pattern indicates that both models pay more attention to information at the beginning and end of long contexts while underperforming for passages in the middle, confirming that positional bias also manifests in Bangla QA tasks and mirrors the "lost in the middle" phenomenon observed in prior studies (Liu et al., 2023).

Table 1: Evaluation results of GPT-4.1-nano and Gemini-2.5-flash-lite on different indices, showing Exact Match (EM) and F1 Score.

Index	**GPT-4.1-nano**		**Gemini-2.5-Flash-Lite**	
	EM	F1	EM	F1
1	60.63%	67.40%	61.59%	76.32%
3	51.11%	57.36%	57.78%	73.13%
5	40.95%	48.23%	56.19%	72.29%
7	38.73%	43.86%	56.83%	72.16%
10	41.59%	48.00%	54.29%	71.30%

Similar positional patterns emerge across all context lengths for both models. For length 5, GPT-4.1-nano performs best at early positions (64.13% EM, 69.42% F1) but drops in the middle, while Gemini-2.5-Flash-Lite maintains higher and more stable scores (60.63–60.95% EM, 74.50–75.66% F1) with only a mild mid-sequence dip. For longer contexts (20 and 30), the U-shaped trend becomes more pronounced: GPT-4.1-nano starts relatively high but declines sharply in the middle (down to 34.92% EM, 40.27% F1 at length 20 and 30.16% EM, 34.09% F1 at length 30) before a slight end-of-sequence recovery. Gemini-2.5-Flash-Lite follows the same pattern but consistently outperforms GPT-4.1-nano, maintaining stronger EM/F1 scores even at mid-range positions (e.g., 54.60–55.56% EM and 64.49–70.93% F1). Overall, this confirms a robust U-shaped positional bias in Bangla long-context QA, consistent with findings from English benchmarks.

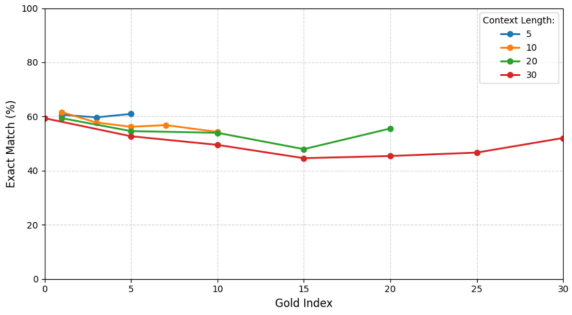

Figure 1: Performance comparison using Exact Match in Gemini-2.5-Flash-Lite

Figure 1 shows the performance of Gemini-2.5-Flash-Lite on different context lengths and gold passage positions using Exact Match. Detailed results for context lengths 5, 20, and 30 are provided in Appendix Tables 3, 4, and 5 along with performance graph of the models for visualization in Figure 3- 5.

Why Gemini Outperforms GPT-4.1-Nano: While both models show a similar U-shaped positional pattern, Gemini-2.5-Flash-Lite consistently achieves higher EM and F1 scores across all context lengths. Several factors may explain this gap. First, architectural differences may give Gemini stronger long-context retrieval, such as improved attention routing or memory-efficient mechanisms. Second, its tokenizer offers better subword coverage for Indo-Aryan languages, reducing Bangla word fragmentation and improving span extraction. Third, Gemini's broader multilingual and South Asian training corpus likely provides richer exposure to Bangla morphology, orthographic variation, and Wikipedia-style text. Together, these advantages help Gemini maintain more reliable attention over long Bangla contexts, especially in the middle regions where GPT-4.1-Nano degrades more sharply.

6 Conclusion

In this work, we introduced a Bangla long-context QA dataset with semantically challenging distractors and reported preliminary results on generative QA models, GPT-4.1-Nano and Gemini-2.5-Flash-Lite. Our early findings confirm positional biases similar to the "Lost in the Middle" phenomenon observed in English: models achieve higher accuracy when the gold passage appears at the beginning or end of the context, but struggle when it is placed in the middle. Among the tested models, Gemini-2.5-Flash-Lite consistently outperformed GPT-4.1-nano. These results represent an initial step rather than a complete solution. Our ongoing work focuses on two directions: (1) evaluating more Bangla LLMs while enforcing grounding in provided passages (like TigerLLM (Raihan and Zampieri, 2025)), and (2) expanding the dataset with more questions, diverse domains, and multi-hop reasoning. By pursuing these directions, we aim to provide a stronger benchmark and a more comprehensive understanding of how Bangla LLMs process extended contexts. Other task, like intrinsic bias measurements of Bangla (Sadhu et al., 2024), can be done for longer context and context length variation using our dataset.

Limitations

Despite providing a first step toward long-context QA in Bangla, our study has several limitations. The dataset is small, with each instance containing only a single gold passage, limiting multi-hop reasoning and domain coverage. Distractor passages, though carefully designed, may not fully capture real-world complexity, and evaluation of a few LLMs using span-based metrics (Exact Match and F1) may not reflect generative answer quality. Moreover, we did not focus on Bangla-specific linguistic aspects such as morphology and syntax, which remain important directions for future work. Overall, these results are preliminary, and further work is needed to expand the dataset, explore more reasoning scenarios, and test additional models.

Acknowledgments

The authors would like to express their sincere gratitude to the Islamic University of Technology (IUT) for providing financial support during the dataset curation phase. The authors also acknowledge the CUPE Research Grant Fund for supporting the OpenAI API model usage costs.

References

Ajwad Abrar, Farzana Tabassum, and Sabbir Ahmed. 2024. Performance evaluation of large language models in bangla consumer health query summarization. In *2024 27th International Conference on Computer and Information Technology (ICCIT)*, pages 2748–2753.

Tasnim Ahmed, Shahriar Ivan, Ahnaf Munir, and Sabbir Ahmed. 2024. Decoding depression: Analyzing

social network insights for depression severity assessment with transformers and explainable ai. *Natural Language Processing Journal*, 7:100079.

Nokimul Hasan Arif, Shadman Rabby, Md Hefzul Hossain Papon, and Sabbir Ahmed. 2025. Preemptive hallucination reduction: An input-level approach for multimodal language model. *Preprint*, arXiv:2505.24007.

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

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. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664, Vienna, Austria. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, and Rifat Shahriyar. 2022. Banglanlg: Benchmarks and resources for evaluating low-resource natural language generation in bangla. *CoRR, abs/2205.11081*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

Syed Mohammed Sartaj Ekram, Adham Arik Rahman, Md. Sajid Altaf, Mohammed Saidul Islam, Mehrab Mustafy Rahman, Md Mezbaur Rahman, Md Azam Hossain, and Abu Raihan Mostofa Kamal. 2022. BanglaRQA: A benchmark dataset for underresourced Bangla language reading comprehension-based question answering with diverse question-answer types. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2518–2532, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2025. How to train long-context language models (effectively). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7376–7399, Vienna, Austria. Association for Computational Linguistics.

Google. 2024. Gemini 2.5 flash: Our next-generation model for efficiency. `https://deepmind.google/technologies/gemini/pro/`. Accessed: 2024-11-19.

Yunpeng Huang, Jingwei Xu, Junyu Lai, Zixu Jiang, Taolue Chen, Zenan Li, Yuan Yao, Xiaoxing Ma, Lijuan Yang, Hao Chen, Shupeng Li, and Penghao Zhao. 2024. Advancing transformer architecture in long-context large language models: A comprehensive survey. *Preprint*, arXiv:2311.12351.

Mohsinul Kabir, Mohammed Saidul Islam, Md Tahmid Rahman Laskar, Mir Tafseer Nayeem, M Saiful Bari, and Enamul Hoque. 2024. BenLLM-eval: A comprehensive evaluation into the potentials and pitfalls of large language models on Bengali NLP. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2238–2252, Torino, Italia. ELRA and ICCL.

Alvi Khan, Fida Kamal, Mohammad Abrar Chowdhury, Tasnim Ahmed, Md Tahmid Rahman Laskar, and Sabbir Ahmed. 2023a. BanglaCHQ-summ: An abstractive summarization dataset for medical queries in Bangla conversational speech. In *Proceedings of the First Workshop on Bangla Language Processing (BLP-2023)*, pages 85–93, Singapore. Association for Computational Linguistics.

Alvi Khan, Fida Kamal, Nuzhat Nower, Tasnim Ahmed, Sabbir Ahmed, and Tareque Chowdhury. 2023b. NERvous about my health: Constructing a Bengali medical named entity recognition dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5768–5774, Singapore. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, Sawsan Alqahtani, M Saiful Bari, Mizanur Rahman, Mohammad Abdullah Matin Khan, Haidar Khan, Israt Jahan, Amran Bhuiyan, Chee Wei Tan, Md Rizwan Parvez, Enamul Hoque, Shafiq Joty, and Jimmy Huang. 2024. A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13785–13816, Miami, Florida, USA. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Huang. 2023. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 431–469, Toronto, Canada. Association for Computational Linguistics.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2024a. LooGLE: Can long-context language

models understand long contexts? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16304–16333, Bangkok, Thailand. Association for Computational Linguistics.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024b. Long-context llms struggle with long in-context learning. *Preprint*, arXiv:2404.02060.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, Yuanxing Zhang, Zhuo Chen, Hangyu Guo, Shilong Li, Ziqiang Liu, Yong Shan, Yifan Song, Jiayi Tian, Wenhao Wu, and 18 others. 2025. A comprehensive survey on long context language modeling. *Preprint*, arXiv:2503.17407.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Ridwan Mahbub, Ifrad Khan, Samiha Anuva, Md Shihab Shahriar, Md Tahmid Rahman Laskar, and Sabbir Ahmed. 2023. Unveiling the essence of poetry: Introducing a comprehensive dataset and benchmark for poem summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14878–14886, Singapore. Association for Computational Linguistics.

Tamzeed Mahfuz, Satak Kumar Dey, Ruwad Naswan, Hasnaen Adil, Khondker Salman Sayeed, and Haz Sameen Shahgir. 2025. Too late to train, too early to use? a study on necessity and viability of low-resource Bengali LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1183–1200, Abu Dhabi, UAE. Association for Computational Linguistics.

Shahriar Kabir Nahin, Rabindra Nath Nandi, Sagor Sarker, Quazi Sarwar Muhtaseem, Md Kowsher, Apu Chandraw Shill, Md Ibrahim, Mehadi Hasan Menon, Tareq Al Muntasir, and Firoj Alam. 2025. Titullms: A family of bangla llms with comprehensive benchmarking. *arXiv preprint arXiv:2502.11187*.

OpenAI. 2024. Gpt-4.1 series: Technical report. `https://openai.com`. Accessed: 2024-11-19.

Nishat Raihan and Marcos Zampieri. 2025. Tigerllm-a family of bangla large language models. *arXiv preprint arXiv:2503.10995*.

Jayanta Sadhu, Ayan Khan, Abhik Bhattacharjee, and Rifat Shahriyar. 2024. An empirical study on the characteristics of bias upon context length variation for bangla. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1501–1520.

Md Shihab Shahriar, Ahmad Al Fayad Chowdhury, Md Amimul Ehsan, and Abu Raihan Kamal. 2023. Question answer generation in bengali: Mitigating the scarcity of qa datasets in a low-resource language. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 430–441.

Tasmiah Tahsin Mayeesha, Abdullah Md Sarwar, and Rashedur M Rahman. 2021. Deep learning based question answering system in bengali. *Journal of Information and Telecommunication*, 5(2):145–178.

Abdullah Khan Zehady, Safi Al Mamun, Naymul Islam, and Santu Karmaker. 2024. Bongllama: Llama for bangla language. *arXiv preprint arXiv:2410.21200*.

Xu Zheng, Ziqiao Weng, Yuanhuiyi Lyu, Lutao Jiang, Haiwei Xue, Bin Ren, Danda Paudel, Nicu Sebe, Luc Van Gool, and Xuming Hu. 2025. Retrieval augmented generation and understanding in vision: A survey and new outlook. *Preprint*, arXiv:2503.18016.

A Appendix

A.1 Dataset Example and Statistics

Each question in our dataset is stored in JSON format, containing the question, a list of passages (with title, content, and source), the gold answer, and metadata. Figure 2 shows a sample instance (truncated for space).

To provide an overview of our dataset, Table 2 summarizes the key statistics, including the total number of items and passages, as well as the average passage length in terms of tokens and characters.

Table 2: Summary statistics of the dataset

Statistic	Value
Total items	350
Total passages	10,500
Avg. passage length (tokens)	175.62
Avg. passage length (characters)	1,130.41

A.2 Model Details

We evaluated two extractive QA models on our dataset.

Gemini-2.5-Flash-Lite: Gemini-2.5-Flash-Lite (Google, 2024) is part of Google's Gemini family of models, designed specifically for low latency, high throughput, and cost efficiency. Despite being a lightweight model, it supports up to 1 million tokens of context, making it

```
{
  "question": "কোন ফিফা বিশ্বকাপের আয়োজক দেশ সৌদি আরব?",
  "language": "bn",
  "documents": [
    {"title": "২০৩৪ ফিফা বিশ্বকাপ",
     "content": "২০৩৪ ফিফা বিশ্বকাপ হবে ২৫ তম ফিফা বিশ্বকাপ আসর, এই চতুর্বার্ষিক আন্তর্জাতিক ফুটবল চ্যাম্পিয়নশিপে ফিফা এর সদস্যভুক্ত জাতীয় দলগুলি পরস্পর... (truncated)",
     "source": "https://bn.wikipedia.org/wiki..."},

    {"title": "২০১০ ফিফা বিশ্বকাপ",
     "content": "২০১০ ফিফা বিশ্বকাপ হচ্ছে আন্তর্জাতিক ফুটবল প্রতিযোগিতা ফিফা বিশ্বকাপের ঊনিশতম আসর। ফিফা বিশ্বকাপ হচ্ছে বিশ্বের প্রধান ফুটবল প্রতিযোগিতা। এই... (truncated)",
     "source": "https://bn.wikipedia.org/wiki..."},

    {"title": "২০২২ ফিফা বিশ্বকাপ",
     "content": "২০২২ ফিফা বিশ্বকাপ হচ্ছে ফিফা দ্বারা আয়োজিত চতুর্বার্ষিক আন্তর্জাতিক ফুটবল প্রতিযোগিতা ফিফা বিশ্বকাপের ২২তম আসরের চূড়ান্ত পর্ব, যেখানে আন্তর্জাতিক... (truncated)",
     "source": "https://bn.wikipedia.org/wiki..." },
     .
     .
     .
    {"title": "২০০২ ফিফা বিশ্বকাপ",
     "content": "২০০২ ফিফা বিশ্বকাপ চতুর্বার্ষিক আন্তর্জাতিক ফুটবল প্রতিযোগিতা ফিফা বিশ্বকাপের ১৭তম আসরের চূড়ান্ত পর্ব ছিল, যেখানে আন্তর্জাতিক ফুটবল সংস্থা ফিফার... (truncated)",
     "source": "https://bn.wikipedia.org/wiki..." }
  ],
  "answer": "২০৩৪ ফিফা বিশ্বকাপ",
  "relevant_document_index": 0,
  "context_length": 30,
  "metadata": {
  "source_dataset": "Wikipedia_Bangla",
  "retrieval_method": "Manual",
  "notes": "Relevant document at the beginning to test primary bias."
  }
}
```


Figure 2: Example of an instance from our dataset in JSON format.

suitable for long-context tasks while maintaining affordability. Its release emphasizes stability and accessibility, with one of the lowest per-token costs among commercial LLMs, making it an appealing choice for researchers and practitioners working in resource-constrained environments.

GPT-4.1-Nano: GPT-4.1-Nano (OpenAI, 2024) is the smallest and most affordable member of OpenAI's GPT-4.1 family, introduced in 2025. Like Gemini-2.5-Flash-Lite, it supports a 1 million token context window, enabling it to handle extended inputs effectively. GPT-4.1-Nano is marketed as the fastest and cheapest variant in the GPT-4.1 lineup, optimized for deployment in cost-sensitive or large-scale applications. Despite its reduced size, it demonstrates strong reasoning and comprehension capabilities, striking a balance between performance and accessibility.

A.3 Evaluation Metrics

We evaluate QA model performance using two widely adopted metrics: Exact Match (EM) and F1 score. These metrics provide complementary perspectives on model accuracy.

Exact Match (EM): Exact Match measures the percentage of predictions that exactly match the reference answers. It is a strict metric: a prediction is counted as correct only if it exactly matches the gold answer after normalizing for punctuation, articles, and capitalization. EM is particularly useful for assessing models in scenarios where precise answers are required, such as extractive QA tasks. However, it does not reward partially correct answers or alternative phrasings, making it less informative for generative models that may produce valid but slightly different answers.

F1 Score: The F1 score captures the token-level overlap between the predicted and reference answers, allowing partial credit for answers that are mostly correct. It is the harmonic mean of precision and recall, defined as follows:

$$\text{Precision} = \frac{|\text{pred tokens} \cap \text{ref tokens}|}{|\text{pred tokens}|}$$

$$\text{Recall} = \frac{|\text{pred tokens} \cap \text{ref tokens}|}{|\text{ref tokens}|}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Precision measures the proportion of predicted tokens that are correct, while recall measures the proportion of reference tokens that are captured by the prediction. F1 balances both aspects, providing a finer-grained evaluation. This metric is especially suitable for generative QA models, which may produce answers that are semantically correct but do not exactly match the reference text.

A.4 Implementation Details

All experiments were implemented using the HuggingFace Transformers library with a PyTorch backend, which provided a flexible and reliable framework for working with pre-trained QA models. Inputs were tokenized using the respective model tokenizers, and the maximum input length was set to accommodate the chosen context sizes,

ensuring that the models could process all passages in each instance without truncation.

The models were run in inference mode on a GPU-enabled environment on Colab, which allowed us to efficiently handle the large number of passages and maintain reasonable processing times. Using Colab provided a convenient and reproducible platform, with consistent hardware and software configurations.

A.5 Evaluation Results

Tables 1–5 report the detailed Exact Match (EM) and F1 scores of GPT-4.1-nano and Gemini-2.5-Flash-Lite across different context lengths (5, 10, 20, and 30) and varying positions of the relevant passage within the input.

The results show a consistent positional bias: performance is highest when the relevant context is placed at the beginning of the input, declines significantly when the context is in the middle, and recovers slightly when it appears at the end. This U-shaped trend aligns with the "lost in the middle" phenomenon observed in prior long-context QA studies, suggesting that the effect also holds for Bangla question answering.

Table 3: Evaluation results of GPT-4.1-nano and Gemini-2.5-Flash-Lite with context length 5 at different indices, showing Exact Match (EM) and F1 Score.

Index	GPT-4.1-nano		Gemini-2.5-flash-lite	
	EM	F1	EM	F1
0	64.13%	69.42%	60.63%	75.66%
3	44.76%	51.52%	59.68%	74.50%
5	49.52%	55.04%	60.95%	75.45%

Table 4: Evaluation results of GPT-4.1-nano and Gemini-2.5-Flash-Lite with context length 20 at different indices, showing Exact Match (EM) and F1 Score.

Index	GPT-4.1-nano		Gemini-2.5-flash-lite	
	EM	F1	EM	F1
0	59.37%	67.07%	59.37%	74.17%
5	45.08%	52.13%	54.60%	69.27%
10	36.51%	43.07%	53.97%	68.49%
15	34.92%	40.27%	47.94%	64.49%
20	37.78%	42.91%	55.56%	70.93%

Figure 3- 5 shows the performance of both models on different context lengths and gold passage positions.

Table 5: Evaluation results of GPT-4.1-nano and Gemini-2.5-Flash-Lite with context length 30 at different indices, showing Exact Match (EM) and F1 Score.

Index	GPT-4.1-nano		Gemini-2.5-flash-lite	
	EM	F1	EM	F1
1	58.73%	64.96%	59.37%	74.41%
5	42.22%	47.78%	52.70%	67.31%
10	33.97%	39.62%	49.52%	65.74%
15	35.56%	41.56%	44.60%	59.45%
20	32.06%	36.05%	45.40%	60.09%
25	30.16%	34.09%	46.67%	60.84%
30	35.24%	39.69%	52.06%	67.13%

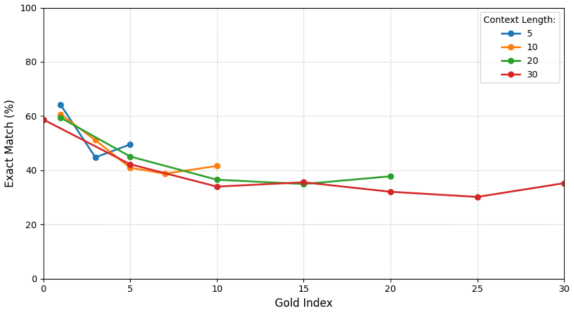

Figure 3: Performance comparison using Exact Match in GPT-4.1-nano

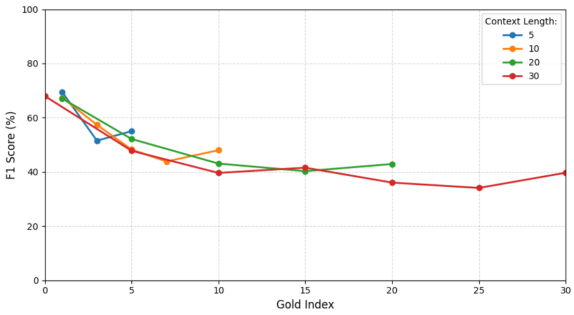

Figure 4: Performance comparison using F1 Score in GPT-4.1-nano

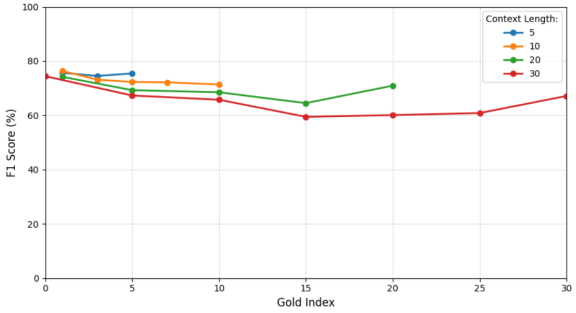

Figure 5: Performance comparison using F1 Score in Gemini-2.5-Flash-Lite