TigerLLM - A Family of Bangla Large Language Models

Nishat Raihan
George Mason University
Fairfax, VA, USA
mraihan2@gmu.edu

Marcos Zampieri
George Mason University
Fairfax, VA, USA
mzampier@gmu.edu

Abstract

The development of Large Language Models (LLMs) remains heavily skewed towards English and a few other high-resource languages. This linguistic disparity is particularly evident for Bangla - the 5^{th} most spoken language. A few initiatives attempted to create open-source Bangla LLMs with performance still behind high-resource languages and limited reproducibility. To address this gap, we introduce TigerLLM - a family of Bangla LLMs. Our results demonstrate that these models surpass all open-source alternatives and also outperform larger proprietary models like GPT3.5 across standard benchmarks, establishing TigerLLM as the new baseline for future Bangla language modeling.

1 Introduction

LLMs have fundamentally transformed NLP by achieving exceptional performance across a broad range of tasks (Brown et al., 2020; Chowdhery et al., 2022; Raihan et al., 2025c). While these models exhibit unprecedented capabilities in language understanding, generation, reasoning, and specialized applications, their advancements predominantly benefit high-resource languages (Alam et al., 2024). This inequality is particularly noticeable for Bangla. Despite having about 237 million native speakers,[1] Bangla remains quite underserved in modern NLP advancements.

This under-representation stems primarily from the limitation of high-quality training data. While proprietary models like GPT-4 (Brown et al., 2023) and Claude-3.5 (Bai et al., 2024) demonstrate reasonable Bangla capabilities, open-source alternatives consistently underperform. Recent multilingual models such as Gemma-2 (Gemma et al., 2024) and LLaMA 3.1 (Dubey et al., 2024), despite leveraging diverse training corpora and advanced tokenization systems like TikTokenizer (Corso et al., 2024), also fail to deliver satisfactory performance for Bangla.

1.1 Limitations of Bangla LLM Initiatives

Training Recent attempts at developing Bangla LLMs (see Table 1) through continual pretraining (titu-Gemma) and model distillation approaches (Zehady et al., 2024) have yielded low and non-reproducible results (see Table 2), often performing worse than their base models. The absence of technical documentation and academic publications further compounds this issue by making result reproduction impossible. Our investigation into these models' performances reveals the need for improvement in the training process. While the unavailability of pretraining corpora limits our analysis of that phase, the finetuning approach demonstrates consistent problematic patterns.

Data Most Bangla LLM initiatives rely on translated versions of synthetic datasets like Alpaca-Instruct (Taori et al., 2023) and OpenOrca (Mitra et al., 2023), which are generated through model distillation (Hinton et al., 2015). This approach suffers from two fundamental limitations: (1) the datasets are generated by early GPT-3.5 (Brown et al., 2020) releases, a model with limited Bangla support, resulting in suboptimal instruction quality, and (2) these English datasets are translated to Bangla using machine translation systems like Google Translate with limited quality checks, further degrading the training data quality. These cascading compromises in training data ultimately result in poor model performance.

1.2 Contributions

To address the recurring challenges in Bangla LLM development, we introduce three fundamental contributions:

1. The **Bangla-TextBook** corpus, comprising 10 million tokens of carefully curated educational

[1] ethnologue.com/language/ben/

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 887–896

	Base-LLM	Size	pt	corpora	ft	ft-dataset	Paper/Report?	Reproducibility?
titu-Gemma	Gemma-2	2B	4.4B	✗	✗	✗	✗	✗
titu-LLaMA	LLaMA-3.1	3B	37B	✗	✗	✗	✗	✗
Bangla-LLaMA	LLaMA-3.2	3B	✓	✗	172K	Orca-translated	✓	✗
G2B	Gemma-2	9B	✗	✗	145K	Alpaca-translated	✗	✗
Bangla-LLaMA	LLaMA-2	13B	✓	✗	145K	Alpaca-translated	✗	✗
TigerLLM	LLaMA-3.2	1B	10M	Bangla-TextBook	100K	Bangla-Instruct	✓	✓
TigerLLM	Gemma-2	9B	10M	Bangla-TextBook	100K	Bangla-Instruct	✓	✓

Table 1: Comparative analysis of Bangla LLM initiatives and their methodological approaches. The pretraining (*pt*) and finetuning (*ft*) columns indicate corpus size in tokens and instruction count respectively.

content across multiple domains, prioritizing content quality over scale.

2. A high-quality **Bangla-Instruct** dataset of 100 thousand instruction-response pairs, generated through self-instruct (Wang et al., 2023) and model distillation using state-of-the-art teacher models (GPT-4o and Claude-3.5-Sonnet).
3. The **Tiger-LLM** family (1B and 9B parameters), featuring models pretrained and finetuned on our high-quality datasets, achieving 30-55% performance improvements over existing benchmarks.

All components are open-sourced to establish robust foundations for future Bangla language modeling research.[2]

2 Related Work

Early transformer-based *encoder-only* pre-trained language models such as BERT (Devlin et al., 2019) concentrate on high-resource languages like English. Subsequent work adapts them to mid- and low-resource contexts through continued pre-training and task-specific finetuning. In Bangla, for instance, Sami et al. (2022) present BANGLABERT, demonstrating that a dedicated monolingual encoder markedly improves downstream classification and QA relative to multilingual baselines.

The shift to *decoder-only* models has produced large multilingual models — e.g. BLOOM (Le Scao et al., 2022), LLAMA3 (Dubey et al., 2024), and AYA (Üstün et al., 2024)—that cover dozens of under-represented languages. Yet empirical analyses reveal that these models still perform best when prompted in high-resource languages, with significant degradation for languages such as Bangla or Swahili (Raihan et al., 2025a; Jin et al., 2024).

As discussed in the previous section, dedicated Bangla decoder models remain scarce and fragmented. GPT2-Bangla (Bhattacharjee et al., 2023) continues GPT-2 pre-training on a 4GB Bangla corpus, while Bong-LLAMA (Zehady et al., 2024) and the *titu-Gemma*[3] checkpoint attempt instruction tuning on translated datasets. These efforts often lack rigorous evaluation protocols, transparent data curation, or reproducible training pipelines—as reflected in the inconsistent results summarized in Table 1. Consequently, a clear methodological gap persists in developing open, reproducible decoder-only LLMs that natively support Bangla and other low-resource languages.

3 Bangla-TextBook Corpus

Previous Bangla LLMs rely predominantly on corpora sourced from OSCAR (Ortiz Suárez et al.) and Common Crawl (Bhattacharjee et al., 2022; Zehady et al., 2024), despite quality control challenges. While alternative Bangla corpora have emerged (Bhattacharyya et al., 2023), the absence of curated educational content remains a critical gap. This emphasis on data quality is particularly significant given recent findings by Gunasekar et al. (2023) and Raihan et al. (2025b), which demonstrate that LLMs achieve superior performance through high-quality training data, even with reduced volume.

To bridge this gap, we present the Bangla-TextBook corpus, constructed exclusively from high-quality **open-source** educational materials published by the National Curriculum and Textbook Board of Bangladesh. We collect texts from 163 textbooks for Grades 6-12, resulting in a total of 9,897,623 tokens and 697,903 sentences.

[2] https://github.com/mraihan-gmu/TigerLLM/tree/main/

[3] https://huggingface.co/hishab/titulm-gemma-2-2b-v1.1

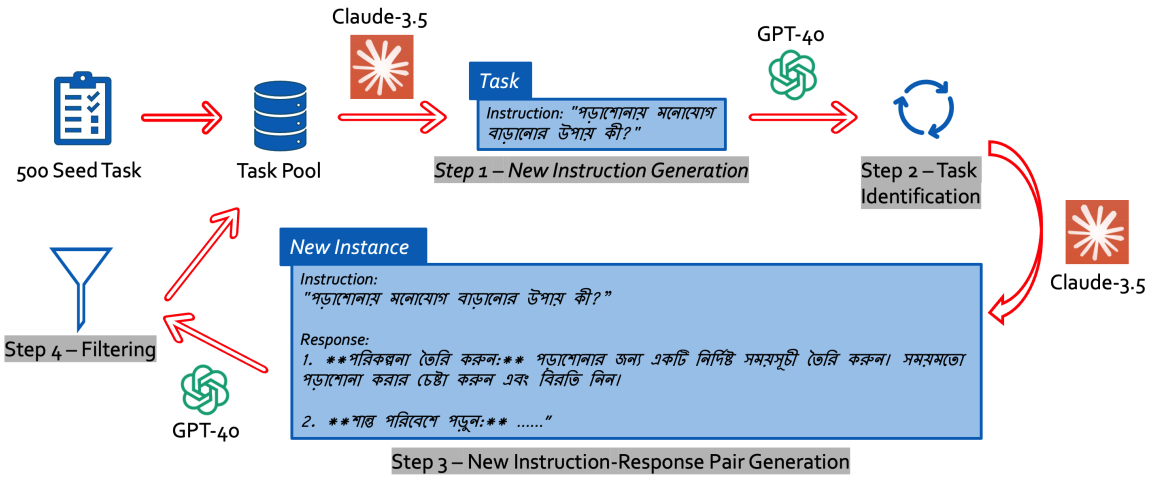

Figure 1: The Bangla-Instruct generation pipeline. With 500 seed tasks, we employ a multi-step process using GPT-4o and Claude-3.5-Sonnet as teacher models to generate instruction-response pairs in Bangla.

4 Bangla-Instruct

To address the limitations described in Section 1.1, we introduce Bangla-Instruct, a collection of 100,000 native Bangla instruction-response pairs bootstrapped using self-instruct (Wang et al., 2023). While instruction datasets like Alpaca (Taori et al., 2023) and OpenOrca (Mitra et al., 2023) utilized GPT3 and GPT3.5 respectively, we significantly improve upon their approach by employing GPT-4 and Claude-3.5-Sonnet as our teacher models, leveraging their superior instruction-following capabilities.

Our dataset creation begins with 500 diverse seed tasks carefully curated by a team of 50 undergraduate and graduate students from leading Bangladeshi universities (Appendix A.1). These volunteers, spanning various academic disciplines and geographical regions of Bangladesh, ensure our seed tasks capture authentic linguistic patterns and cultural contexts. Each seed task undergoes multiple rounds of peer review to maintain quality and cultural sensitivity. Further information on quality control is presented in Appendix (Appendix A.3).

Our generation pipeline consists of four primary steps, each designed to maintain data quality and cultural authenticity (see Figure 1).

(1) Seed & Instruction Generation: We begin with a human-curated seed pool $\mathcal{T}_s = \{t_1, \ldots, t_{500}\}$ drawn from *50* volunteers representing five academic disciplines across Bangladesh (see Appendix A.1). At every generation round i, we sample $k = 8$ seed tasks and prompt CLAUDE to create a candidate batch of instructions $\mathcal{I}_n$, expanding coverage of the ten seed categories $c_{1\ldots10}$ listed in Appendix A.2 while preserving authentic linguistic patterns.

(2) Task Typing: Each instruction $i \in \mathcal{I}_n$ is classified by GPT-4o into $\tau(i) \in \{\textit{open-ended}, \textit{classification}, \textit{generation}\}$, providing the expected answer style and the minimum-length threshold $l_{\min}(\tau)$ used in subsequent filtering.

(3) Response Drafting: Conditioned on $(i, \tau(i))$, CLAUDE produces a comprehensive response r_i. We retain the highest-scoring draft according to an internal coherence metric $c(i, r)$.

(4) Multi-stage Filtering: GPT-4o applies the four-criteria filter $\mathcal{F}$—Language ($\mathcal{L}$), Cultural ($\mathcal{C}$), Quality ($\mathcal{Q}$), and Novelty ($\mathcal{N}$) (see Appendix A.3). On average, ~63% of (i, r) pairs pass $\mathcal{F}$, yielding a balanced complexity mix (40% basic, 40% intermediate, 20% advanced). Valid pairs are appended to $\mathcal{T}_s$, and the loop continues until 100K high-quality instruction–response pairs are reached.

By coupling two complementary LLMs with strict verification and a human-seeded, domain-balanced task pool, our pipeline mitigates error propagation and preserves cultural nuance—addressing shortcomings observed in earlier Bengali instruction datasets (see Appendix A for full statistics).

5 TigerLLM

As candidate base models, we consider 3 families of multilingual LLMs - LLaMA 3.2 (1B, 3B) (Dubey et al., 2024), Gemma-2 (2B, 9B) (Gemma et al., 2024) and Pangea (7B) (Yue et al., 2024).

Evolution of TigerLLM Figure 2 depicts the final selection of the models and a high-level overview of the process.

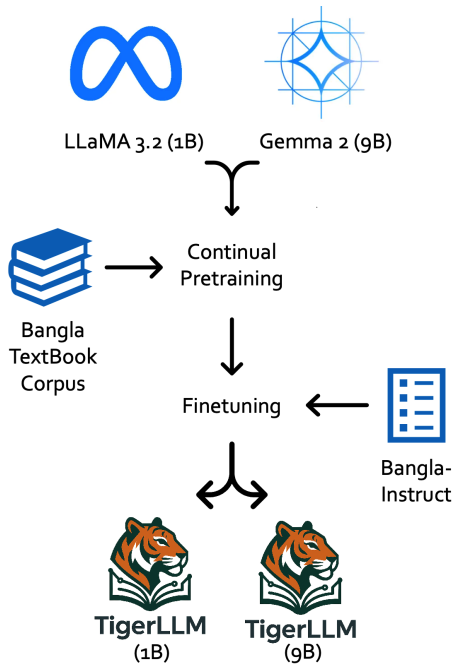

Figure 2: Evolution of TigerLLM.

Upon the selection phase, we finalize two pretrained language models—LLaMA 3.2 (1B) and Gemma 2 (9B)—chosen for their robust foundational capacities. These models then undergo continual pretraining (see Figure 3) on a specialized **Bangla-TextBook** corpus, which infuses them with a richer understanding of the Bangla language, including its context-specific nuances, stylistic variations, and domain-specific terminology.

Pretraining We utilize a computing cluster with 8 NVIDIA A100 GPUs (40GB each), 512GB RAM, and 2TB storage. The distributed training setup enables efficient parallel processing, completing the pretraining in approximately 120 hours on this high-performance configuration with gradient checkpointing enabled.

Continual Pretraining We use the `Bangla-TextBook` corpus for the models to learn culture and language-specific nuances and gather sufficient and reliable knowledge from a set of high-quality texts. The pretraining phase has been carried out multiple times with empirical choices of hyper-parameters.

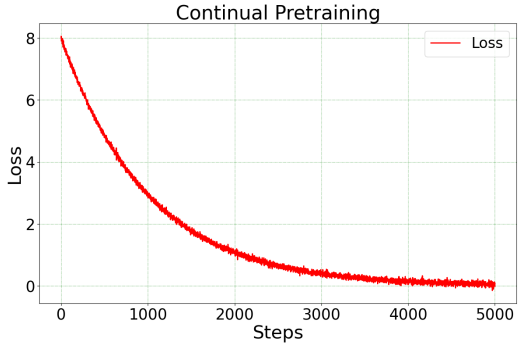

Figure 3: Continual Pretraining - Loss per Steps.

Finetuning We conduct finetuning on a single NVIDIA A100 (40GB) through Google Colab[4], supported by 80GB RAM and 256GB storage. The process completes in approximately 96 hours, proving sufficient for model adaptation and task-specific optimization with minimal computational overhead.

Model Distillation Following this continual pretraining step, the models are finetuned on a carefully curated **Bangla-Instruct** dataset (Figure 4). LoRA (Hu et al., 2021) is not used, we implement full finetuning for better learning. To speed up the training process, we utilize Flash Attention (Dao et al., 2022), we set key parameters: 2048 token maximum sequence length, batch size of 8, 4 gradient accumulation steps, and 3 epochs. Learning rate (5×10^{-5}), weight decay (0.02), and 10% warm-up steps ensure stable convergence. Table 5 in Appendix B lists complete hyperparameters.

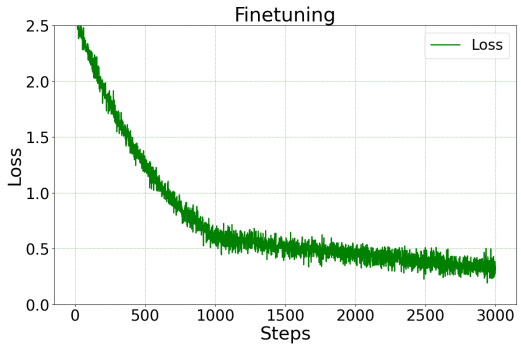

Figure 4: Finetuning - Loss per Steps.

By blending the foundational strengths of LLaMA and Gemma with specialized Bangla corpora and instruction-oriented finetuning, the final TigerLLM models emerge as optimized solutions capable of delivering high-quality, instruction-following re-

[4] `colab.research.google.com`

	MMLU-bn	PangBench-bn	BanglaQuaD	mHumanEval-bn	BEnQA	BanglaRQA
	understanding	*multitasking*	*question answering*	*coding*	*knowledge*	*reasoning*
GPT3.5	0.55	0.55	0.50	0.56	0.50	0.49
Gemini-Flash1.5	0.66	0.57	0.62	0.58	0.56	0.61
GPT4o-mini	0.67	0.62	0.65	0.56	0.60	0.60
LLaMA3.2 (11B)	0.22	0.19	0.21	0.15	0.18	0.20
Gemma 2 (27B)	0.35	0.51	0.43	**0.64**	0.50	0.56
Pangea (7B)	0.18	0.15	0.17	0.10	0.14	0.16
Titu-LLM	0.06	0.19	0.08	0.02	0.17	0.21
Bong-LLaMA	0.05	0.12	0.08	0.02	0.15	0.13
Bangla-LLaMA	0.02	0.08	0.05	0.10	0.11	0.09
Bangla-Gemma	0.18	0.15	0.12	0.10	0.22	0.19
TigerLLM (1B)	0.61	0.55	0.68	0.61	0.59	0.62
TigerLLM (9B)	**0.72**	**0.68**	**0.70**	0.63	**0.65**	**0.68**

Table 2: Performance comparison of TigerLLM with other models on various Bangla-specific benchmarks. All values are reported as **% in Pass@1**, where higher scores indicate better performance.

sponses tailored to Bangla-language tasks.

6 Evaluation

Bangla LLM Benchmarks Although there has been limited research on Bangla LLMs, several benchmarks have been established to assess their performance. We focus on five benchmarks specifically curated to evaluate Bangla LLMs across a diverse set tasks. For multitask understanding, we use the Bangla subset of MMLU-Pro (Wang et al., 2024) and PangBench (Yue et al., 2024). For question answering, we consider BanglaQuaD (Rony et al., 2024), while for general knowledge, we use BEnQA (Shafayat et al., 2024). For reasoning tasks, we refer to BanglaRQA (Ekram et al., 2022).

As shown in the survey of Raihan et al. (2024), most coding benchmarks like HumanEval (Chen et al., 2021) do not support Bangla, so we utilize the Bangla subset of mHumanEval (Raihan et al., 2025a).

Results We present the results obtained by the two TigerLLM models compared to a variety of strong LLM baselines in Table 2. The performance comparison of various models on Bangla-specific benchmarks reveals a common trend. The fine-tuned models generally perform worse than their base counterparts across most tasks. In particular, the results reported by the authors are not reproducible, as mentioned in Section 1.1. However, TigerLLM is the only finetuned model, consistently outperforming both its base and fine-tuned variants across all tasks. Even the 1B variant does better than most models, falling short to only its 9B counterpart, further validating our emphasis on high-quality data (Section 4).

Takeaways TigerLLM demonstrates that carefully curated, high-quality datasets can yield superior performance even with smaller model sizes. Our results show that the 1B parameter model outperforms larger alternatives across multiple benchmarks, emphasizing the importance of data quality over quantity. The success of our Bangla-TextBook corpus and Bangla-Instruct dataset establishes a new paradigm for low-resource language model development.

7 Conclusion and Future Work

This paper introduces TigerLLM, a family of state-of-the-art Bangla language models that outperforms existing alternatives across six benchmarks. TigerLLM's success stems from two key innovations: (1) the high-quality Bangla-TextBook corpus derived from educational materials and (2) the carefully curated Bangla-Instruct dataset generated using advanced teacher models.

The three resources introduced here (corpus, instruction dataset, and models) establish a robust foundation for future Bangla language modeling research. Together, they will contribute to speeding up advances in Bangla language modeling.

In future work we will conduct a deeper qualitative analysis of the model's behavior, broaden the corpus to cover a wider array of domains, scale the model to larger parameter counts without compromising quality, and devise richer evaluation metrics tailored specifically to Bangla tasks.

Limitations

While TigerLLM delivers state-of-the-art performance, several limitations warrant acknowledgment. First, our Bangla-TextBook corpus, though carefully curated, is limited to educational materials from grades 6-12, potentially missing broader linguistic patterns present in other domains. The 10 million token size, while sufficient for our current models, may constrain scaling to larger architectures. Additionally, our Bangla-Instruct dataset, despite its quality-focused generation process, covers only a subset of possible instruction types and may not fully capture the complexity of real-world Bangla language use cases.

Furthermore, our models are currently limited to 1B and 9B parameters, primarily due to computational constraints and our emphasis on thorough experimentation with smaller computationally efficient architectures. While this approach enabled rapid iteration and quality-focused development, it may not fully exploit the potential benefits of larger model scales.

Ethical Considerations

Our work prioritizes ethical considerations throughout the development process. The Bangla-TextBook corpus uses open-source publicly available educational materials from the National Curriculum and Textbook Board of Bangladesh. The volunteer-driven seed task creation process incorporated diverse perspectives while maintaining cultural sensitivity and avoiding harmful biases.

We implemented rigorous filtering mechanisms to ensure cultural appropriateness, gender neutrality, and religious sensitivity in our instruction dataset. The multi-stage review process, involving both automated checks and human verification, helps prevent the propagation of harmful stereotypes or biases. Additionally, our open-source approach promotes transparency and enables community oversight of model behavior.

We strongly recommend that users implement appropriate safeguards when deploying TigerLLM in production environments, particularly for applications involving sensitive information or critical decision-making.

References

Firoj Alam, Shammur Absar Chowdhury, Sabri Boughorbel, and Maram Hasanain. 2024. Llms for low resource languages in multilingual, multimodal and dialectal settings. In *Proceedings of EACL*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Tyler Conerly, et al. 2024. Claude 3.5 sonnet technical report.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, and Md Saiful Islam. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the ACL (NAACL-2022)*.

Abhik Bhattacharjee, Tahmid Hasan, and Md Saiful Islam. 2023. Banglagpt: A gpt-2 language model continued pre-training for bangla.

Pramit Bhattacharyya, Joydeep Mondal, Subhadip Maji, and Arnab Bhattacharya. 2023. Vacaspati: A diverse corpus of bangla literature. In *Proceedings of the 13th International Joint Conference on Natural Language Processing*.

Tom Brown, Ben Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. 2023. Gpt-4 technical report.

Tom Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. 2021. Evaluating large language models trained on code. *arXiv preprint*, arXiv:2107.03374.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Francesco Corso, Francesco Pierri, and Gianmarco De Francisci Morales. 2024. What we can learn from tiktok through its research api. In *Proceedings of WebSci*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Proceedings of NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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 ACL (EMNLP-2022)*.

Team Gemma, Morgane Riviere, Shreya Pathak, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, et al. 2021. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

Yiqiao Jin, Mohit Chandra, Gaurav Verma, Yibo Hu, Munmun De Choudhury, and Srijan Kumar. 2024. Better to ask in english: Cross-lingual evaluation of large language models for healthcare queries. In *Proceedings of the ACM Web Conference 2024*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, et al. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Nishat Raihan, Antonios Anastasopoulos, and Marcos Zampieri. 2025a. mHumanEval - a multilingual benchmark to evaluate large language models for code generation. In *Proceedings of NAACL*.

Nishat Raihan, Christian Newman, and Marcos Zampieri. 2024. Code llms: A taxonomy-based survey. In *Proceedings of IEEE BigData*.

Nishat Raihan, Joanna C. S. Santos, and Marcos Zampieri. 2025b. MojoBench: Language modeling and benchmarks for mojo. In *Findings of the ACL (NAACL-2025)*.

Nishat Raihan, Mohammed Latif Siddiq, Joanna CS Santos, and Marcos Zampieri. 2025c. Large language models in computer science education: A systematic literature review. In *Proceedings of SIGCSE*.

Md. Rashad Al Hasan Rony, Sudipto Kumar Shaha, Rakib Al Hasan, Sumon Kanti Dey, Amzad Hossain Rafi, Ashraf Hasan Sirajee, and Jens Lehmann. 2024. Banglaquad: A bengali open-domain question answering dataset.

Abdullah As Sami, Nusrat Jahan Prottasha, Mohammad Shamsul Arefin, Pranab Kumar Dhar, and Takeshi Koshiba. 2022. Bangla-bert: transformer-based efficient model for transfer learning and language understanding. *IEEE Access*.

Sheikh Shafayat, H M Quamran Hasan, Minhajur Rahman Chowdhury Mahim, Rifki Afina Putri, James Thorne, and Alice Oh. 2024. Benqa: A question answering and reasoning benchmark for bengali and english.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, and Yann Dubois. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, and Noah A Smith. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of ACL*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Xiang Yue, Yueqi Song, Akari Asai, Seungone Kim, et al. 2024. Pangea: A fully open multilingual multimodal llm for 39 languages. *arXiv preprint arXiv:2410.16153*.

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

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, and et al. 2024. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*.

A `Bangla-Instruct` Curation

A.1 Volunteer Information

The seed tasks were created by 50 undergraduate and graduate students from various universities across Bangladesh, ensuring geographical and academic diversity:

- 15 students from Computer Science and Engineering.
- 10 students from Bengali Literature.
- 10 students from Business Administration.
- 8 students from Science and Engineering.
- 7 students from Social Sciences.

Each volunteer contributed 10 diverse instructions, resulting in our initial pool of 500 seed tasks. The distribution ensured coverage across multiple domains while preserving authentic Bengali linguistic patterns and cultural contexts.

A.2 The Seed Dataset

Our seed dataset comprises 10 distinct categories, carefully chosen to cover a broad spectrum of tasks relevant to Bengali language and culture:

1. **Cultural Knowledge and Heritage** (c_1): Tasks focusing on Bengali traditions, festivals, folk tales, and historical events. These include explaining cultural practices, describing traditional ceremonies, and discussing historical significance of various customs.

2. **Academic Writing** (c_2): Structured writing tasks ranging from essay outlines to full academic compositions. Topics cover various academic disciplines while maintaining Bengali writing conventions and scholarly standards.

3. **Mathematical Problem Solving** (c_3): Tasks involving mathematical concepts explained in Bengali, including algebra, geometry, and arithmetic. Special attention is given to Bengali mathematical terminology and local problem-solving contexts.

4. **Programming and Technical** (c_4): Programming problems described in Bengali with solutions in standard programming languages. Includes algorithm explanation, code documentation, and technical concept elaboration in Bengali.

5. **Creative Writing** (c_5): Open-ended creative tasks including story writing, poetry composition, and descriptive passages. Emphasizes Bengali literary devices, metaphors, and cultural storytelling elements.

6. **Scientific Explanation** (c_6): Tasks requiring clear explanation of scientific concepts in Bengali, focusing on making complex ideas accessible while maintaining technical accuracy. Covers physics, chemistry, biology, and environmental science.

7. **Business and Economics** (c_7): Professional writing tasks including business case analyses, market reports, and economic concept explanations. Incorporates local business contexts and Bengali business terminology.

8. **Social Issues Analysis** (c_8): Critical analysis tasks addressing contemporary social issues in Bangladesh and Bengali society. Includes problem identification, cause analysis, and solution proposition.

9. **Data Analysis and Statistics** (c_9): Tasks involving interpretation and analysis of data presented in Bengali, including statistical concepts explanation, data visualization description, and numerical analysis.

10. **Language and Translation** (c_{10}): Tasks focused on Bengali language mastery, including idiom explanation, translation between Bengali and English, and linguistic analysis of Bengali texts.

Each category accounts for approximately 10% of the seed dataset (50 ± 5 tasks per category), ensuring balanced representation across domains. The tasks within each category vary in complexity level: 40% basic, 40% intermediate, and 20% advanced, based on linguistic complexity and cognitive demand.

A.3 Filtering Methodology

Our filtering process $\mathcal{F} : (\mathcal{I}, \mathcal{R}) \to \{0, 1\}$ implements the following criteria:

1. **Language Adherence** ($\mathcal{L}$)
   - Bengali Word Ratio: $\frac{|\text{Bengali Words}|}{|\text{Total Words}|} \geq 0.95$
   - Unicode Consistency: $\forall c \in \text{text}, c \in$ Bengali-UTF8
   - Grammar Check: Using GPT-4o's Bengali grammar scoring function $g(x) \geq 0.8$

2. **Cultural Sensitivity** ($\mathcal{C}$)
    - Religious Neutrality: $r(x) \in [-0.1, 0.1]$ on our bias scale
    - Regional Inclusivity: No specific region/-dialect preference
    - Gender Representation: Balanced pronouns and roles
    - Political Neutrality: Avoidance of partisan content

3. **Content Quality** ($\mathcal{Q}$)
    - Minimum Length: $l(x) \geq l_{min}(\tau)$ where τ is task type
    - Coherence Score: $c(i, r) \geq 0.8$ between instruction i and response r
    - Factual Accuracy: Verified against Bengali Wikipedia
    - Format Adherence: Proper paragraph breaks, lists, or code blocks

4. **Novelty Verification** ($\mathcal{N}$)
    - Similarity Threshold: $\forall j \in \mathcal{D}, \text{sim}(i, j) \leq 0.7$
    - Lexical Diversity: Minimum Type-Token Ratio of 0.4
    - Response Uniqueness: No duplicate responses within same category
    - Task Format Variation: Ensure uniform distribution across formats

A pair (i, r) is accepted if and only if:

$$\mathcal{F}(i, r) = \mathbb{1}[\mathcal{L}(i, r) \wedge \mathcal{C}(i, r) \wedge \mathcal{Q}(i, r) \wedge \mathcal{N}(i, r)] = 1$$

This rigorous filtering ensures the quality and diversity of our final dataset while maintaining Bengali linguistic and cultural authenticity.

B Experimentation Details

B.1 Pretraining HyperParameters

Hyperparameter	**Value**
Per device train batch size	64
Gradient accumulation steps	16
Number of training epochs	4
Learning rate	5×10^{-6}
FP16	False
BF16	True
Dataloader num workers	8
Gradient checkpointing	True
Logging steps	1000
DDP find unused parameters	False
Max gradient norm	1.0
Warmup steps	1000
Evaluation strategy	steps
Evaluation steps	1,000
Save strategy	steps
Save steps	1,000
Save total limit	3
Load best model at end	True
Metric for best model	loss
Greater is better	False

Table 3: Final set of hyperparameters, chosen empirically after several iterations of trial and error, for pretraining on the `Bangla-TextBook` corpus.

B.2 Finetuning Hyperparameters

Parameter	Value
Max Sequence Length	2048
Batch Size (Train/Eval)	16
Gradient Accumulation Steps	4
Number of Epochs	3
Learning Rate	1e-5
Weight Decay	0.02
Warmup Steps	10%
Optimizer	AdamW (8-bit)
LR Scheduler	Cosine
Precision	BF16
Evaluation Strategy	Steps
Evaluation Steps	50
Save Strategy	Steps
Save Steps	Varies
Seed	42

Table 4: Final set of hyperparameters, chosen empirically after several iterations of trial and error, for finetuning TigerLLM (1B).

Parameter	Value
Max Sequence Length	2048
Batch Size (Train/Eval)	32
Gradient Accumulation Steps	8
Number of Epochs	3
Learning Rate	1e-6
Weight Decay	0.04
Warmup Steps	15%
Optimizer	AdamW (8-bit)
LR Scheduler	Cosine
Precision	BF16
Evaluation Strategy	Steps
Evaluation Steps	250
Save Strategy	Steps
Save Steps	Varies
Seed	42

Table 5: Final set of hyperparameters, chosen empirically after several iterations of trial and error, for finetuning TigerLLM (9B).