Evaluating LLMs' Multilingual Capabilities for Bengali: Benchmark Creation and Performance Analysis

Anonymous ACL submission

Abstract

Bengali is an underrepresented language in NLP research. However, it remains a challenge due to its unique linguistic structure and computational constraints. In this work, we systematically investigate the challenges that hinder Bengali NLP performance by focusing on the absence of standardized evaluation benchmarks. We then evaluated 10 recent open source Large Language Models (LLMs) in 8 of the translated datasets and performed a comprehensive error analysis to pinpoint their primary failure modes. Our findings reveal consistent performance gaps for Bengali compared to English, particularly for smaller models and specific model families like Mistral. We also identified promising robustness in certain architectures, such as DeepSeek, that maintain more stable performance across languages. We find that excessive tokenization per row often introduces noise and degrades model accuracy, while concise per word tokenization improves score outcomes. These findings highlight critical areas where current models fall short and underscore the need for improved dataset quality and evaluation methodologies tailored to multilingual contexts. This work will catalyze further research on NLP for underrepresented languages, helping to democratize access to advanced language technologies worldwide.

1 Introduction

Large Language Models (LLMs) have transformed text generation enabling applications in machine translation, text summarization and conversational agents. These models such as GPT-2 and GPT-3 leverage vast amounts of data and deep neural architectures to generate human-like text with fluency (Witteveen and Andrews, 2019). Controlled text generation approaches have also been explored to refine outputs and guide language models toward desirable properties (Yu et al., 2021). Although these research developments have been substantial, text generation in under-resourced languages like Bengali remains a challenge.

Recent efforts have sought to extend LLM capabilities to Bengali, a language spoken by over 230 million people. While general-purpose LLMs perform well in high-resource languages like English and Chinese, Bengali NLP faces limitations due to its linguistic complexity and scarcity of large-scale datasets (Kabir et al., 2023). To address this, dedicated Bengali LLMs such as BanglaBERT (Bhattacharjee et al., 2021), BanglaGPT (Salim et al., 2023) have been developed. More recent Bengali-focused models like TituLLM (Nahin et al., 2025) and TigerLLM (Raihan and Zampieri, 2025) have also emerged, demonstrating promising results in various Bengali NLP tasks. These models aim to enhance performance in Bengali NLP tasks such as text classification, sentiment analysis and machine translation.

However, the development of robust Bengali LLMs is still faced by different challenges. First, the lack of large-scale, high-quality Bengali text corpora limits pretraining and fine-tuning efforts (Shahriar and Barbosa, 2024). While resources like the Sangraha corpus (Khan et al., 2024) developed by AI4Bharat offer numerous data across 22 Indian languages including Bengali, the quality and quantity of Bengali tokens remain limited compared to high-resource languages like English. The Sangraha corpus consists of about 251 billion tokens across all languages, but Bengali's allocation is significantly smaller at about 30 billion tokens. In contrast, English has access to around 2 trillion tokens in large-scale multilingual corpora such as the Common Corpus (Langlais et al., 2025). This huge difference in token availability poses a major challenge in achieving comparable model performance in Bengali NLP. Second, the Bengali language's rich morphology and complex writing system introduce significant tokenization challenges. Unlike English, which uses the Latin script with largely independent characters, Bengali employs an alphasyllabary script where base characters are frequently modified by diacritics and conjunct forms that alter pronunciation and meaning (Alam et al., 2021). These modifications can occur on either side of a base character, forming intricate multi-character grapheme clusters that do not align well with standard tokenization schemes used in LLMs. As a result, traditional subword tokenization methods such as Byte Pair Encoding (BPE) or WordPiece struggle to segment Bengali text effectively, leading to highly fragmented or inconsistent tokens (Shahriar and Barbosa, 2024). This increased token complexity means that models require more training data to learn meaningful inter-token relationships in Bengali than in English. Failure to capture these linguistic nuances not only increases computational overhead but also degrades model performance on downstream tasks. Third, Bengali NLP research suffers from the absence of standardized evaluation datasets, making it difficult to benchmark model performance effectively (Kabir et al., 2023).

This lack of evaluation datasets motivates the need for well-defined benchmark datasets for Bengali LLMs. Without standardized datasets, it is hard to compare models or track improvements in NLP research. While some efforts have been made to curate evaluation datasets (Shafayat et al., 2024) progress is still slow due to the extensive annotation and validation required.

Efforts to develop LLMs for underrepresented languages have explored various methodologies. The Khayyam Challenge (Ghahroodi et al., 2024) curated a large-scale Persian dataset using original non-translated content ensuring language-specific nuances are preserved. Similarly, Cohere's Aya model (Üstün et al., 2024) employed instruction tuning across multiple low-resource languages to enhance linguistic adaptability. AI4Bharat's Sangraha dataset tackled data scarcity by aggregating and refining multilingual corpora . In contrast, Turkish LLM research (Acikgoz et al., 2024) experimented with two approaches: adapting English-trained models via transfer learning and pretraining from scratch. While these efforts have proven effective their applicability to Bengali remains uncertain due to unique linguistic characteristics and uniqueness in Bengali.

Although substantial progress has been made in developing NLP resources for Bengali, there remain opportunities to accelerate advancement further. Typically, when creating initial benchmarks for lower-resourced languages, researchers bootstrap by translating existing English datasets into the target language, as demonstrated in prior works for Persian and Turkish. However, this initial step has not yet been widely adopted for Bengali, largely due to practical constraints, including the substantial manual validation effort required to correct machine translation errors, associated time investments, and overall costs. Because current machine translation systems often introduce inaccuracies and lose linguistic nuance, manual intervention becomes necessary to refine and validate the translated data. In this study, we directly address these challenges by systematically translating major English benchmark datasets into Bengali and did a performance analysis on them.

Motivated by these challenges, this research aims to bridge the existing gaps in Bengali NLP by constructing high-quality evaluation datasets. To address these limitations, this work contributes in a few key areas.

- We publicly release a comprehensive suite of high-quality Bengali benchmark datasets, along with the accompanying translation pipeline and codebase to facilitate reproducible research and future advancements in Bengali NLP evaluation.
- We describe the methodology used to translate and curate high-quality datasets.
- We conduct inference experiments and analyze results to assess model effectiveness of open source multilingual models.
- We analyze tokenization behavior across Bengali and English benchmarks, revealing that Bengali inputs produce significantly larger token counts per instance and per word with dataset remaining consistent across both languages.
- We identify the impact of tokenization granularity on performance, showing that higher tokens per row often correlate with lower model scores (due to noise) while more compact per-word tokenization tends to improve accuracy.
- We examine language-specific encoding efficiencies, demonstrating that English tokens carry higher average bytes per token compared to Bengali with implications for model resource requirements.

In Section 2, we describe the datasets that were translated, outline the translation methodologies, and explain the rationale behind the choice of translation models. In Section 3, we detail the experimental procedures, including the datasets selected for inference, the evaluation metrics used, and the results obtained. Section 4 presents an analysis of the results, summarizes key findings, and outlines directions for future work. Finally, in Section 5, we discuss the challenges encountered during translation and highlight the limitations of our approach.

2 Methodology

The translation pipeline for converting English NLP benchmarks begins with dataset selection and blind review using multiple models. GPT-4o-mini was chosen for translation, supported by prompt

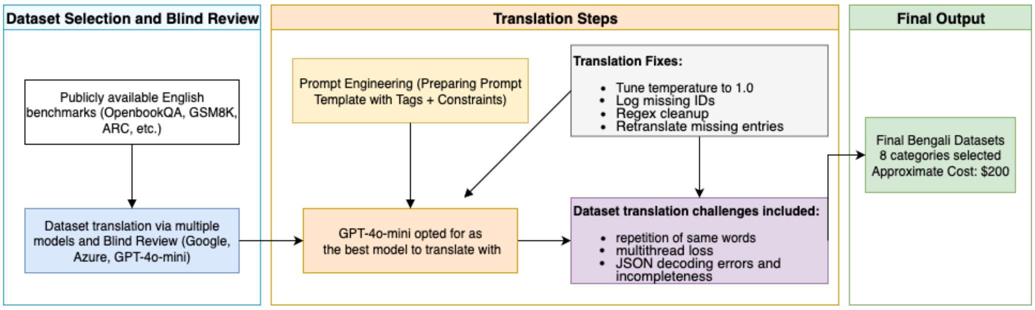

Dataset Selection and Blind Review
Publicly available English benchmarks (OpenbookQA, GSM8K, ARC, etc.)
Dataset translation via multiple models and Blind Review (Google, Azure, GPT-4o-mini)
Translation Steps
Prompt Engineering (Preparing Prompt Template with Tags + Constraints)
GPT-4o-mini opted for as the best model to translate with
Translation Fixes:
• Tune temperature to 1.0
• Log missing IDs
• Regex cleanup
• Retranslate missing entries
Dataset translation challenges included:
• repetition of same words
• multithread loss
• JSON decoding errors and incompleteness
Final Output
Final Bengali Datasets
8 categories selected
Approximate Cost: $200

Figure 1: Methodology Overview

engineering. The post-processing steps addressed translation errors and formatting issues. The final output includes 8 cleaned Bengali datasets completed at a cost of approximately $200.

2.1 Dataset Selection

To select appropriate datasets, we refer to the methodology used in the white paper by LLaMA, identifying commonly used datasets that align with our research objectives. This approach allowed us to ensure the inclusion of high-quality, diverse and representative text corpora for Bengali language modeling. A summary of the dataset statistics is attached.

2.2 Translation

For the translation process, we utilized OpenAI's gpt-4o-mini-2024-07-18 model to translate the selected datasets from English to Bengali while preserving linguistic accuracy and contextual integrity.

The model was instructed through comprehensive prompting to properly translate the dataset and not change the underlying meaning of the original text. Special attention was given to preserving the integrity of ground truth values to prevent any corruption. Temperature values ranging from 0.0 to 1.0 were used to control the translation quality and creativity. As the model sometimes responds with elaborate and redundant answers, special care for that was taken during the prompting process. An example of the prompting template is shown in Table 2.

2.3 Translation Decisions

In our study, we performed a blind review of translations generated by three different services: Google Translate, Azure's Translation Endpoint and OpenAI's gpt-4o-mini-2024-07-18. Each translation was assessed by human reviewers without revealing its source. Based on the reviewers' feedback, we determined that gpt-4o-mini-2024-07-18 produced the most accurate and coherent translations among the three.

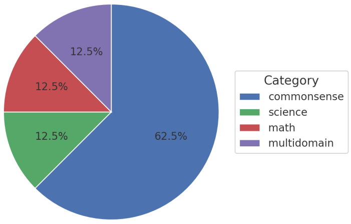


Figure 2: Dataset Distribution

2.4 Translation Challenges

During the translation process, we encountered several issues:

- **Repetitive Translations**: Some words were being repeated excessively, leading to unnatural sentence structures. To mitigate this, we increased the temperature parameter to 1 while keeping other parameters constant, which helped introduce variability and improve translation quality.

- **Missing Entries Due to Multithreading**: Some dataset entries were skipped due to parallel processing errors. We resolved this issue by analyzing logs and re-processing the missing translations to ensure dataset completeness.

- **Decoding Errors**: Some dataset entries had decoding errors due to the JSON not being parsed properly . These errors include missing comma(,)delimiters, unclosed quotation marks(“”), mismatched key-value pairs, missing “bangla translation” tags, unescaped json quotes etc. This was resolved by updating the corresponding regex and escaping response strings as necessary.

- **Incomplete Translations**: Some translated dataset entries contained incomplete sentences, missing answer-key values and missing options. Such sentences had to be retranslated to fix the issue.

Dataset Name	Train	Dev.	Test	Task Type	Dataset Type
OpenbookQA	4957	500	500	MCQ	Multi-step reasoning, commonsense
ARC	3370	869	3548	MCQ	Grade-school science
BigBenchHard	-	-	Var.	MCQ	Logical reasoning
Alpaca Eval	-	-	10465	Instruction	Benchmark
Anthropic	86372	-	35006	-	Safety, helpfulness
Apps	5000	-	5000	-	Coding
BFCL	-	-	250	-	Function calling
BoolQ	9427	3270	-	-	Reading comprehension
CommonSenseQA	9741	1221	1140	MCQ	Commonsense reasoning
Dolly	-	-	7295	Instruction	Varied NLP tasks
GSM8k	7473	-	1319	Numbers	Grade-school math
Hellaswag	39905	10042	10003	-	Commonsense reasoning
HumanEval	-	-	164	-	Code generation
MATH	8599	-	4999	Exact Match	Math reasoning
MMLU	98487	1528	13869	MCQ	College-level reasoning
MMLU-Pro	-	70	12032	-	College-level reasoning
MR-GSM8k	-	-	12024	Exact Match	Math reasoning
PIQA	16113	-	3084	MCQ	Commonsense reasoning
SIQA	33410	1954	-	MCQ	Social IQ
TruthfulQA	-	-	1634	MCQ	Truthfulness assessment
Winogrande	19482	1267	1767	MCQ	Pronoun resolution

Table 1: Summary of Dataset Statistics

2.5 Translation Results

Twenty major LLM benchmark datasets were translated into Bengali. From these, eight datasets were selected, spanning the Commonsense, Science, Math, and Multidomain categories. The total cost of translation amounted to approximately $200.

3 Experimental Details

We selected eight benchmark datasets spanning four high-level categories for our evaluations. In the **Commonsense** category, we included HELLASWAG, WINOGRANDE, COMMONSENSEQA, BOOLQ and OPENBOOKQA. For **Science**, we used ARC. In the **Math** category, we chose GSM8K-MAIN and for **Multidomain**, we selected MMLU. Each dataset was translated into Bengali according to our methodology and our experiments measure model performance on these translated versions.

3.1 Chosen Models

For our research, we selected all available open-source multilingual LLaMA models to ensure broad generalization and comprehensive evaluation. The specific models used in our experiments include:

3.2 Evaluation Metrics

The evaluation process was done without finetuning the Llama family of models and running inference on the corresponding datasets. To assess the performance of the models, the following evaluation metrics were employed:

- **Accuracy**: Measures the proportion of correctly answered questions out of the total number of questions. Formally,

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} \nVdash(\text{response}_i = \text{answer}_i)}{n}$$

where $\nVdash(\cdot)$ is the indicator function (1 if the condition is true, and 0 otherwise).

- **Response Error Rate (RER) and Response Adherence Rate (RAR).** The *Response Error Rate (RER)* measures the fraction of model-generated responses that fail to conform to any of the valid answer formats specified for a given input. More precisely, it captures the rate at which the model's response does not begin with any of the acceptable prefixes. The complement of this metric, *Response Adherence Rate (RAR)*, represents the proportion of responses that correctly begin with a valid option. These metrics are particularly useful for structured or categorical tasks where responses are expected to adhere to a predefined format, such as "yes" or "no" in binary classification tasks.

Formally, let n be the total number of examples, resp_i denote the model's response for example i, and P_i be the set of valid prefixes (e.g., class labels or canonical answer forms) for that example. Define an indicator variable:

$$e_i = \nVdash\left(\forall\, p \in P_i : \neg\left(\text{resp}_i \text{ starts with } p\right)\right),$$

where $\nVdash(\cdot)$ is the indicator function, which returns 1 if the condition is true and 0 otherwise.

Role	Content
System	You are a professional translator tasked with accurately translating text from English to Bengali. Your primary goal is to provide precise and culturally appropriate translations, regardless of the content's nature.
User	Translate the following English text into Bengali and ensure the output is valid JSON with all strings enclosed in double quotes: <english_text> {{ ”input”: {input}, ”target”: {target} }} </english_text> Guidelines: 1. Translate accurately, maintaining meaning, tone, and context. 2. Handle idiomatic expressions appropriately. 3. Preserve specialized terminology or proper nouns. 4. Translate sensitive content accurately without censorship. 5. Do not translate JSON keys, only values. 6. Ensure valid JSON output with double-quoted strings. Output within <bangla_translation> tags. Notes in <translator_notes> tags.

Table 2: Prompting Structure for English to Bengali Translation

Model Family	Size	Multilingual	Bengali in Pretraining	Reference
LLaMA 3.1	8B	Limited	✗(Token overlap only)	(Grattafiori et al., 2024)
LLaMA 3.1	70B	Limited	✗(Token overlap only)	(Grattafiori et al., 2024)
LLaMA 3.2	3B	Limited	✗	(Grattafiori et al., 2024)
LLaMA 3.3	70B	Limited	✗	(Grattafiori et al., 2024)
Qwen 2.5	7B	Yes	✓	(Qwen et al., 2025)
Qwen 2.5	72B	Yes	✓	(Qwen et al., 2025)
Mistral	7B	No	✗	(Jiang et al., 2023)
Mistral Small	24B	No	✗	(Mistral AI Team, 2025)
DeepSeek-R1	14B	Yes	✓	(Guo et al., 2025)
DeepSeek-R1	70B	Yes	✓	(Guo et al., 2025)

Table 3: Benchmark models evaluated on Bengali data. We used chat or instruct-tuned version of each model. Bengali coverage is based on available documentation or token overlap estimates.

The RER is then given by:

$$\text{RER} = \frac{1}{n}\sum_{i=1}^{n} e_i.$$

Accordingly, the RAR is defined as:

$$\text{RAR} = 1 - \text{RER} = \frac{1}{n}\sum_{i=1}^{n}(1 - e_i).$$

In the case of the BoolQ dataset, which is a binary question answering task with “yes” or “no” as valid answers, we evaluate RER by checking whether each model response exactly matches one of these expected labels. To ensure consistency, responses are first normalized through a label mapping function (e.g., mapping “Yes” to “yes”) and converted to lowercase. The error condition is met if the response does not match any of the valid labels associated with the input. The final RER is computed as the proportion of such mismatches across all examples, and RAR is derived as its complement. This evaluation framework ensures that the model not only answers correctly but also adheres strictly to the expected response format.

	EN									BN								
Model	OBQA	CSQA	ARC-E	ARC-C	BoolQ	GSM8K-M	Winogrande	HellaSwag	MMLU	OBQA	CSQA	ARC-E	ARC-C	BoolQ	GSM8K-M	Winogrande	HellaSwag	MMLU
llama3.1:8b	0.790	0.735	0.888	0.788	0.809	0.111	0.616	0.753	0.647	0.172	0.423	0.529	0.433	0.671	0.387	0.519	0.193	0.282
llama3.1:70b	0.938	0.816	0.969	0.934	0.882	0.923	0.805	0.882	0.814	0.790	0.650	0.922	0.846	0.822	0.811	0.648	0.624	0.650
llama3.2:3b	0.730	0.701	0.832	0.720	0.669	0.586	0.535	0.583	0.567	0.330	0.287	0.349	0.323	0.446	0.145	0.485	0.257	0.280
llama3.3:70b	0.896	0.771	0.941	0.916	0.885	0.931	0.804	0.709	0.802	0.764	0.643	0.918	0.835	0.833	0.827	0.635	0.659	0.652
qwen2.5:7b	0.874	0.817	0.917	0.881	0.786	0.882	0.679	0.819	0.690	0.516	0.464	0.654	0.538	0.572	0.106	0.516	0.435	0.414
qwen2.5:72b	0.960	0.849	0.969	0.943	0.893	0.909	0.809	0.905	0.817	0.336	0.609	0.835	0.779	0.848	0.824	0.355	0.222	0.587
mistral:7b	0.048	0.014	0.056	0.038	0.719	0.416	0.011	0.162	0.068	0.006	0.048	0.019	0.019	0.594	0.011	0.240	0.046	0.026
mistral:24b	0.900	0.811	0.937	0.911	0.817	0.785	0.773	0.810	0.754	0.538	0.577	0.842	0.743	0.780	0.754	0.570	0.401	0.527
deepseek-r1:14b	0.774	0.645	0.733	0.723	0.872	0.859	0.766	0.811	0.571	0.500	0.457	0.568	0.500	0.792	0.557	0.552	0.339	0.367
deepseek-r1:70b	0.316	0.432	0.238	0.231	0.839	0.923	0.647	0.273	0.226	0.490	0.611	0.755	0.749	0.872	0.764	0.565	0.381	0.552

Table 4: Accuracy performance comparison of models across datasets for English (EN) and Bengali (BN).

	EN									BN								
Model	OBQA	CSQA	ARC-E	ARC-C	BoolQ	GSM8K-M	Winogrande	HellaSwag	MMLU	OBQA	CSQA	ARC-E	ARC-C	BoolQ	GSM8K-M	Winogrande	HellaSwag	MMLU
llama3.1:8b	0.000	0.001	0.040	0.015	0.000	0.121	0.000	0.000	0.000	0.658	0.016	0.061	0.051	0.000	0.258	0.005	0.407	0.257
llama3.1:70b	0.010	0.008	0.013	0.005	0.000	0.022	0.002	0.000	0.001	0.012	0.003	0.002	0.001	0.002	0.065	0.000	0.001	0.005
llama3.2:3b	0.000	0.003	0.041	0.019	0.000	0.073	0.000	0.000	0.018	0.018	0.002	0.001	0.000	0.000	0.465	0.041	0.018	0.032
llama3.3:70b	0.042	0.065	0.043	0.026	0.000	0.022	0.000	0.223	0.009	0.010	0.016	0.004	0.003	0.000	0.058	0.001	0.035	0.009
qwen2.5:7b	0.008	0.000	0.041	0.018	0.000	0.020	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.039	0.000	0.000	0.000
qwen2.5:72b	0.008	0.000	0.019	0.009	0.001	0.045	0.000	0.000	0.000	0.582	0.102	0.116	0.110	0.000	0.033	0.463	0.670	0.144
mistral:7b	0.914	0.975	0.938	0.950	0.023	0.211	0.979	0.722	0.886	0.986	0.788	0.930	0.929	0.024	0.561	0.542	0.815	0.913
mistral:24b	0.000	0.002	0.039	0.020	0.000	0.039	0.000	0.125	0.000	0.172	0.003	0.000	0.000	0.000	0.051	0.000	0.367	0.000
deepseek-r1:14b	0.190	0.192	0.250	0.235	0.011	0.039	0.056	0.007	0.282	0.244	0.162	0.314	0.344	0.015	0.158	0.046	0.277	0.348
deepseek-r1:70b	0.672	0.494	0.761	0.757	0.067	0.028	0.249	0.669	0.747	0.416	0.118	0.186	0.156	0.007	0.077	0.184	0.399	0.256

Table 5: RER performance comparison of models across datasets for English (EN) and Bengali (BN).

- **LLM-Judge** : Uses a separate LLM-based “judge” system to determine whether a model's answer conveys the same meaning as the correct ground truth, even if the wording differs. We define this as the fraction of answers for which the judge returns a “Correct” verdict:

$$\text{LLM-Judge} = \frac{\sum_{i=1}^{n} \mathbb{1}(\text{verdict}_i = \text{"Correct"})}{n}.$$

The judge is implemented via a few-shot learning approach with GPT models to provide consistent, human-like assessments.

	EN									BN								
Model	OBQA	CSQA	ARC-E	ARC-C	BoolQ	GSM8K-M	Winogrande	HellaSwag	MMLU	OBQA	CSQA	ARC-E	ARC-C	BoolQ	GSM8K-M	Winogrande	HellaSwag	MMLU
llama3.1:8b	0.790	0.735	0.928	0.801	0.809	0.122	0.777	0.409	0.648	0.474	0.381	0.562	0.451	0.671	0.477	0.710	0.377	0.360
llama3.1:70b	0.940	0.820	0.982	0.939	0.882	0.945	0.856	0.412	0.814	0.796	0.593	0.923	0.846	0.823	0.862	0.768	0.470	0.652
llama3.2:3b	0.730	0.703	0.872	0.738	0.669	0.600	0.680	0.352	0.572	0.340	0.265	0.350	0.323	0.446	0.243	0.642	0.282	0.290
llama3.3:70b	0.934	0.822	0.983	0.941	0.885	0.953	0.860	0.417	0.807	0.768	0.585	0.918	0.835	0.833	0.876	0.750	0.484	0.653
qwen2.5:7b	0.876	0.817	0.957	0.899	0.786	0.896	0.789	0.423	0.690	0.516	0.429	0.654	0.538	0.572	0.110	0.726	0.451	0.414
qwen2.5:72b	0.966	0.849	0.988	0.951	0.893	0.952	0.870	0.421	0.817	0.790	0.606	0.938	0.868	0.848	0.861	0.772	0.547	0.677
mistral:7b	0.678	0.654	0.865	0.704	0.727	0.508	0.585	0.387	0.535	0.252	0.195	0.283	0.239	0.605	0.030	0.697	0.372	0.282
mistral:24b	0.900	0.811	0.977	0.930	0.817	0.792	0.869	0.404	0.754	0.670	0.528	0.842	0.743	0.780	0.775	0.751	0.514	0.527
deepseek-r1:14b	0.958	0.810	0.980	0.948	0.893	0.895	0.976	0.898	0.816	0.692	0.808	0.830	0.758	0.847	0.634	0.895	0.832	0.606
deepseek-r1:70b	0.954	0.857	0.987	0.957	0.906	0.951	0.991	0.933	0.883	0.868	0.866	0.933	0.883	0.909	0.831	0.946	0.934	0.737

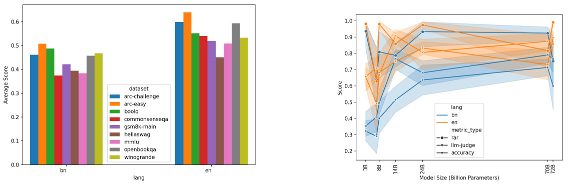
Table 6: LLM Judge performance comparison of models across datasets for English (EN) and Bengali (BN).

These metrics provide a comprehensive overview of the model's effectiveness in understanding and responding to commonsense questions across both English and Bengali languages.

3.3 Result Analysis

In Fig. 3a, we present the average scores grouped by dataset and language. As expected, performance in Bengali is generally lower than in English.

Fig. 3b shows how Accuracy, LLM-Judge, and RAR metrics vary with model size. Smaller models tend to underperform, especially in Bengali,



(a) Average of Accuracy, LLM-Judge, and RAR scores across datasets grouped by language.

(b) Variation of metric scores across model sizes in different languages.

Figure 3: Language-wise score trends and the effect of model size.

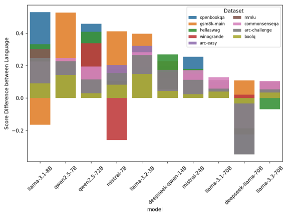


Figure 4: The models sorted by average of the score difference observed between English and Bengali across datasets.

with noticeable drops in accuracy and LLM-Judge scores.

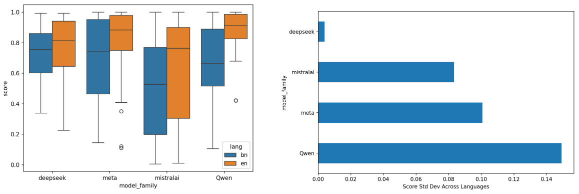
In Fig. 5a, we observe the distribution of scores across various model families. Mistral models consistently underperform across both languages.

Fig. 5b illustrates the standard deviation of average scores across languages. A lower deviation indicates greater robustness. In particular, the DeepSeek model family demonstrates high robustness across languages.

Fig. 4 illustrates the sorted score differences between English and Bengali prompts. Earlier LLaMA models show greater performance drops, likely due to limited Bengali representation in their pretraining data. Interestingly, the Qwen 72B model also appears among the lower-performing group, alongside smaller models (3B–8B). The language gap is most pronounced in tasks involving math (GSM-8K) and commonsense reasoning (Hellaswag, OpenbookQA). In contrast, larger models tend to show more consistent performance across both languages. Moreover, in select scenarios—particularly on DeepSeek and Mistral architectures—Bengali prompts unexpectedly outperform English ones; this may stem from the more structured and context-rich translations, which better align with the models' tokenization and leverage additional semantic cues present in the Bengali prompts.

3.4 Tokenization

We now proceed to evaluate and compare various tokenizers on our translated Bengali datasets. We



(a) LLM-Judge score distributions across different model architecture families.

(b) Standard deviation of average scores across languages for each model family.

Figure 5: Architecture-wise performance and robustness across languages.

report the computed values of the metrics for each tokenizer under consideration. These results highlight the tradeoffs between encoding granularity and byte-efficiency in the context of Bengali text. Finally, we analyze how these differences in tokenization affect downstream model performance.

To simplify notation, we use the following abbreviations for tokenization metrics: average tokens per row (ATPR), average tokens per word (ATPW), average bytes per token (ABPT) (Dagan et al., 2024), and average normalized sequence length (ANSL) (Dagan et al., 2024).

3.4.1 Average Token Count

In order to compare the efficiency of different tokenizers across dataset, we compute the mean number of tokens generated. Formally, given a dataset $\mathcal{D} = \{x_i\}_{i=1}^{N}$ of N text entries (rows), let $T(x)$ denote the number of tokens assigned to text x by a given tokenizer. We then define two related metrics.

3.4.2 Average Token Count Per Row

$$\mathrm{ATPR} = \frac{1}{N}\sum_{i=1}^{N} T(x_i)\,.$$

Here each x_i is the full concatenation of "System Prompt" and "Prompt" from one response CSV row, and $T(x_i)$ is the length of its tokenized sequence.

3.4.3 Average Token Count Per Word

Let $\mathrm{Words}(x_i)$ be the number of whitespace-separated words in x_i. We define

$$\mathrm{ATPW} = \frac{1}{N}\sum_{i=1}^{N} \frac{T(x_i)}{\mathrm{Words}(x_i)}\,.$$

This normalizes each row's token count by its word count, giving a per-word encoding cost.

This metric captures the average amount of raw text (in bytes) that each token represents. Because tokens correspond to subword units, a lower ABPT means each token encodes more of the original text, indicating a more byte-efficient tokenizer.

Conversely, a higher value implies finer granularity more tokens for the same byte length potentially increasing downstream compute costs.

3.4.4 Bytes Per Token

Let $\mathcal{D} = \{D_i\}_{i=1}^N$ be a corpus of N text examples. For each D_i, let $B_i = |D_i|_{\text{bytes}}$ denote its UTF-8 byte length, and $\ell_i^{(\lambda)} = |T_\lambda(D_i)|$ its token count under tokenizer T_λ. The per-example bytes-per-token is

$$r_i^{(\lambda)} = \frac{B_i}{\ell_i^{(\lambda)}},$$

and the average over the corpus is

$$\text{ABPT}(\lambda) = \frac{1}{N}\sum_{i=1}^{N} \frac{B_i}{\ell_i^{(\lambda)}}.$$

This metric reflects the average number of bytes each token spans. Lower ABPT indicates coarser, more byte-efficient tokenization, while higher values suggest finer granularity and potentially greater compute cost.

3.4.5 Average Normalized Sequence Length

Let $\ell_i^{(\beta)} = |T_\beta(D_i)|$ be the token count under the baseline tokenizer T_β. Define the per-example normalized length

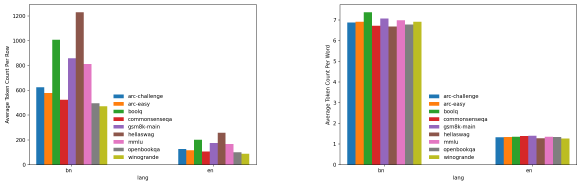
$$n_i^{(\lambda)} = \frac{\ell_i^{(\lambda)}}{\ell_i^{(\beta)}}.$$

Its dataset-wide average is

$$\text{ANSL}(\lambda) = \frac{1}{N}\sum_{i=1}^{N} \frac{\ell_i^{(\lambda)}}{\ell_i^{(\beta)}}.$$

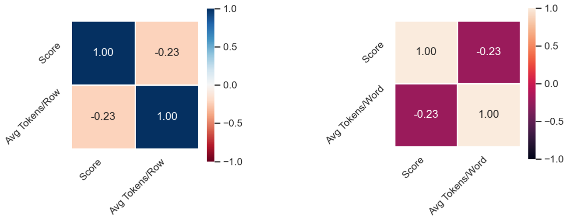
This ratio measures how the tokenizer's sequence length compares to that of a fixed baseline. A value below 1 indicates that T_λ produces shorter token sequences than the baseline—reducing model input length and inference latency—while a value above 1 signals longer, more fragmented encodings that may increase computational overhead.

The bar plots in Figure 6 illustrate the tokenization performance varies across different datasets. At a glance we can see that the Token counts in Bengali are significantly larger than English. In Figure 6a, the average token count per row reveals that boolq and hellaswag lead with over 1000 tokens, suggesting greater complexity or verbosity, particularly in the Bengali dataset. Their English counterparts also rank high but show lower and less varied token counts. The order of datasets with the highest average token counts remains consistent across both bn and en versions, underscoring a persistent trend in tokenization behavior. Figure 6b presents the average token count per word, revealing a more balanced distribution, with bn and lang datasets ranging between 2-7 tokens per word, while en consistently shows the lowest counts, suggesting more efficient tokenization for English. These findings highlight the challenges of tokenizing Bengali text, potentially due to linguistic complexity, compared to English.



(a) ATPR (b) ATPW

Figure 6: Comparison of tokenization efficiency metrics across datasets.

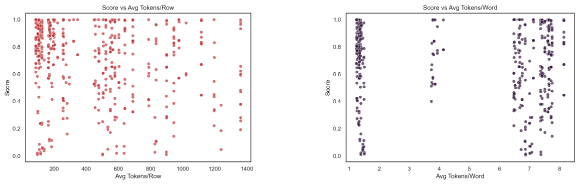


(a) ATPR (b) ATPW

Figure 7: Correlation of token efficiency metrics with LLM-Judge Score.

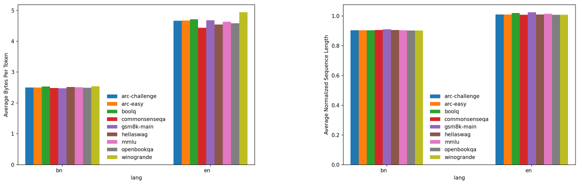
The heatmaps in Figure 7 provide valuable insights into the impact of tokenization on performance metrics. Figure 7a suggests that models with higher token counts per row tend to correlate with lower scores, potentially indicating that capturing more contextual information also introduces more noise. In contrast, Figure 7b reveals that lower token counts per word are associated with lower scores, hinting at the advantage of concise tokenization in maintaining semantic integrity. These findings underscore the need for a balanced tokenization approach, tailoring strategies to dataset characteristics to optimize model performance effectively.

The scatter plots in Figure 8 provide insights into the relationship between tokenization metrics and scores. Figure 8a shows that scores tend to stabilize or slightly decline as the average token count per row increases beyond a certain threshold, suggesting a potential saturation point where additional tokens may not significantly boost performance. Figure 8b indicates that scores are generally higher with lower average token counts per word, implying that more efficient tokenization at the word level could enhance model accuracy. These findings suggest that an optimal tokenization strategy might involve limiting excessive tok-



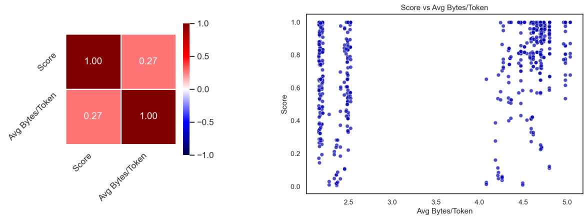
(a) ATPR (b) ATPW

Figure 8: Scatter plot of tokenization efficiency metrics against LLM-Judge Score.



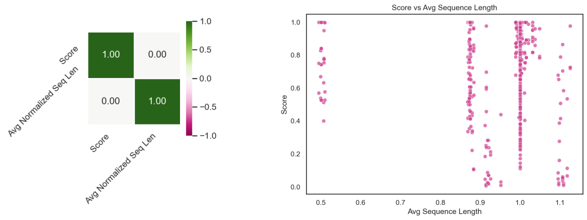
(a) ABPT (b) ANSL

Figure 9: Comparison of tokenization efficiency metrics across datasets and languages (Bengali & English) reflecting variations in encoding efficiency.



(a) Correlation (b) Scatter plot

Figure 10: Effect of tokenization efficiency measured by ABPT on LLM-Judge scores showing how byte-level tokenization impacts on model evaluation quality.



(a) Correlation (b) Scatter plot

Figure 11: Influence of tokenization length normalization, measured by ANSL on LLM-Judge scores demonstrating how relative sequence length affects evaluation outcomes.

enization per row while prioritizing concise word-level representation to maximize score outcomes.

The bar plot in Figure 9a reveals that English(en) datasets consistently show higher average bytes per token, suggesting that English tokenization may involve more complex or larger representations, potentially due to richer vocabulary or encoding schemes. In contrast, Bengali(bn) datasets exhibit lower and more uniform byte counts, indicating a more compact tokenization process, which could reflect simpler linguistic structures or optimized encoding for these datasets. These findings imply that tokenization efficiency varies by language, with English requiring more storage per token, possibly impacting model resource demands.

4 Conclusion

In this work, we conducted a systematic evaluation of recent large language models on Bengali, an underrepresented language in NLP research. By translating and adapting major LLM benchmark datasets, we provided a comprehensive assessment of model performance across multiple metrics, languages, and dataset categories. Our findings reveal consistent performance gaps for Bengali compared to English, particularly for smaller models and specific model families like Mistral. We also identified promising robustness in certain architectures, such as DeepSeek, that maintain more stable performance across languages.

Despite the challenges posed by machine-translated datasets and variability in model outputs, our study highlights critical areas where current models fall short and underscores the need for improved dataset quality and evaluation methodologies tailored to multilingual contexts. We hope that by open-sourcing our datasets and code, this work will catalyze further research on NLP for low-resource languages, helping to democratize access to advanced language technologies worldwide.

Moreover, our detailed tokenization analysis shows that Bengali inputs show substantially higher token counts per instance and per word compared to English, when datasets are kept consistent across languages. We find that excessive tokens per row often introduce noise and degrade model accuracy, while concise per-word tokenization improves score outcomes. Additionally, English tokens carry higher average bytes per token than Bengali, highlighting language-specific resource implications for model deployment.

Future efforts should focus on addressing the limitations noted here, including manual dataset validation, more flexible evaluation criteria to accommodate diverse model output, and improved automatic judging techniques to ensure reliable and fair evaluation.

5 Limitations

While our study offers valuable insights into multilingual model performance, it is not without limitations.

First, the Bengali datasets used in our evaluation were translated from English using automatic machine translation methods. These translations were not manually validated, which may introduce linguistic inaccuracies, ambiguities, or cultural mismatches that could affect model performance unfairly.

Second, model outputs can vary significantly in formatting and phrasing across different model families. While we attempt to evaluate correctness using automated methods such as exact match for accuracy, these strict rules may penalize valid answers that do not conform to a narrow format, especially in generative tasks. This limits the reliability of accuracy-based metrics across diverse models.

Lastly, our use of LLM-as-a-judge assumes that the judgment provided by a reference LLM is accurate. However, LLMs themselves can make mistakes, show bias, or misinterpret nuanced cases. This introduces an additional layer of uncertainty in the evaluation pipeline.

We acknowledge these limitations and consider them important areas for future work, including manual validation, improved normalization across outputs, and more robust automatic evaluation methods.

References

Emre Can Acikgoz, Mete Erdogan, and Deniz Yuret. 2024. Bridging the bosphorus: Advancing turkish large language models through strategies for low-resource language adaptation and benchmarking. *ArXiv*, abs/2405.04685.

Samiul Alam, Tahsin Reasat, Asif Shahriyar Sushmit, Sadi Mohammad Siddique, Fuad Rahman, Mahady Hasan, and Ahmed Imtiaz Humayun. 2021. A large multi-target dataset of common bengali handwritten graphemes. In *International Conference on Document Analysis and Recognition*, pages 383–398. Springer.

Abhik Bhattacharjee, Tahmid Hasan, Kazi Samin Mubasshir, M. Sohel Rahman, Anindya Iqbal, and Rifat Shahriyar. 2021. Banglabert: Combating embedding barrier for low-resource language understanding. *ArXiv*, abs/2101.00204.

Gautier Dagan, Gabriel Synnaeve, and Baptiste Rozière. 2024. Getting the most out of your tokenizer for pre-training and domain adaptation. *Preprint*, arXiv:2402.01035.

Omid Ghahroodi, Marzia Nouri, Mohammad Vali Sanian, Alireza Sahebi, Doratossadat Dastgheib, Ehsaneddin Asgari, Mahdieh Soleymani Baghshah, and Mohammad Hossein Rohban. 2024. Khayyam challenge (persianMMLU): Is your LLM truly wise to the persian language? In *First Conference on Language Modeling*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

M. Golam Kabir, Mohammed Saidul Islam, Md Tahmid Rahman Laskar, Mir Tafseer Nayeem, M Saiful Bari, and Enamul Hoque. 2023. Benllm-eval: A comprehensive evaluation into the potentials and pitfalls of large language models on bengali nlp. *ArXiv*, abs/2309.13173.

Mohammed Khan, Priyam Mehta, Ananth Sankar, Umashankar Kumaravelan, Sumanth Doddapaneni, Suriyaprasaad B, Varun G, Sparsh Jain, Anoop Kunchukuttan, Pratyush Kumar, Raj Dabre, and Mitesh Khapra. 2024. Indicllmsuite: A blueprint for creating pre-training and fine-tuning datasets for indian languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 15831–15879. Association for Computational Linguistics.

Pierre-Carl Langlais, Carlos Rosas Hinostroza, Mattia Nee, Catherine Arnett, Pavel Chizhov, Eliot Krzystof Jones, Irène Girard, David Mach, Anastasia Stasenko, and Ivan P. Yamshchikov. 2025. Common corpus: The largest collection of ethical data for llm pre-training. *Preprint*, arXiv:2506.01732.

Mistral AI Team. 2025. Mistral-Small-24B-Instruct-2501. Hugging Face Model Card, Apache-2.0 licensed. https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501 (accessed 2025-07-20).

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. *Preprint*, arXiv:2502.11187.

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, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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

Md. Shahidul Salim, Hasan Murad, Dola Das, and Faisal Ahmed. 2023. Banglagpt: A generative pretrained transformer-based model for bangla language. *2023 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, pages 56–59.

H M Quamran Hasan Sheikh Shafayat, Minhajur Rahman, Chowdhury Mahim, Rifki Afina, James Putri, Alice Thorne, Oh, Kabir Ahuja, Rishav Hada, Millicent Ochieng, Prachi Jain, Harshita Diddee, Samuel C. Maina, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, Jonathan H. Choi, Kristin E Hickman, and 40 others. 2024. Benqa: A question answering and reasoning benchmark for bengali and english. *ArXiv*, abs/2403.10900.

Arif Shahriar and Denilson Barbosa. 2024. Improving bengali and hindi large language models. In *International Conference on Language Resources and Evaluation*.

Sam Witteveen and Martin Andrews. 2019. Paraphrasing with large language models. In *Conference on Empirical Methods in Natural Language Processing*.

Dian Yu, Kenji Sagae, and Zhou Yu. 2021. Attribute alignment: Controlling text generation from pre-trained language models. In *Conference on Empirical Methods in Natural Language Processing*.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. *Preprint*, arXiv:2402.07827.