Script Fragmentation and Format: What Drives the English-Bengali Performance Gap in Open LLMs?

Shimanto Bhowmik
Rochester Institute of Technology
sb6778@g.rit.edu

Tawsif Tashwar Dipto
Islamic University of Technology
tawsiftashwar@iut-dhaka.edu

Md Sazzad Islam
Stanford University
sazzad14@stanford.edu

Sheryl Hsu
Stanford University
sherylh@stanford.edu

Tahsin Reasat
Bengali.AI
reasat@bengali.ai

Abstract

Bengali is spoken by more than 230 million people, yet no standardized instrument evaluates large language models (LLMs) on Bengali across the task categories used to benchmark frontier models. We release 8 English benchmarks translated into Bengali with a single consistent pipeline and use them to evaluate 10 open LLMs from 4 families on paired English and Bengali inputs. *Script fragmentation* is what subword tokenizers do to Bengali's alphasyllabary, whose written units are grapheme clusters spanning several Unicode code points: they cut the script into pieces smaller than a character, at a cost set by the vocabulary rather than the script itself. *Format* belongs to the evaluation, the answer shape that exact-match scoring demands regardless of whether the model knew the answer. Beyond confirming a substantial gap (macro LLM-judge score 0.79 in English versus 0.63 in Bengali), we show that part of it is a measurement artifact: exact-match accuracy conflates correctness with format adherence and because format failure is asymmetric across languages it distorts the gap for some models three to five fold and even reverses its sign for one reasoning-tuned model. On the fragmentation side, Bengali costs roughly five times more tokens per word than English, the 10 models share only 4 vocabularies and that cost varies twofold across them: the same Bengali word averages 7.9 tokens under the Llama 3 tokenizer but 4.0 under Mistral's Tekken. Under three of the four vocabularies the average Bengali token covers fewer bytes than a single code point. Bengali is the constant here and the vocabularies are not and so what Bengali text costs is set by tokenizer design rather than by the script. Fertility and sequence length correlate only weakly with scores ($r = -0.23$), so we present this as a cost and segmentation concern rather than a driver of the gap. The datasets, pipeline and evaluation code are released.

1 Introduction

Open large language models are now routinely compared on a standard set of English benchmarks covering commonsense reasoning, science questions, mathematics and broad knowledge (Grattafiori et al., 2024; Yang et al., 2025). For Bengali, one of the most widely spoken languages in the world, no comparable evaluation suite exists. This is not because Bengali lacks data or models: it has dedicated pretrained models such as BanglaBERT (Bhattacharjee et al., 2022), BanglaGPT (Salim et al., 2023), TituLLM (Nahin et al., 2025) and TigerLLM (Raihan and Zampieri, 2025) and roughly 30 billion tokens of curated pretraining text in the Sangraha corpus (Khan et al., 2024). What Bengali lacks is an *instrument*: a set of evaluation benchmarks that spans the same task categories as the English suites, is constructed consistently enough that scores are comparable across tasks and comes with an analysis of where its measurements can be trusted. Existing Bengali evaluations cover individual slices (science QA in BEnQA (Shafayat et al., 2024), the task selections of BenLLMEval (Kabir et al., 2023)) and translated collections such as Okapi (Lai et al., 2023) and Indic-Evals (Sarvam AI, 2024) cover several individual benchmarks, but no resource supports a controlled English–Bengali comparison across the full commonsense, science, math and multidomain spectrum. We argue that Bengali is best described not as data-poor but as *evaluation-poor* and this paper is about building and testing the missing instrument.

Proceedings of the Fifth Workshop on Multimodal Machine Learning in Low-resource Languages, pages 1–17
November 9, 2026 ©2026 Association for Computational Linguistics

The first of the two concepts is *script fragmentation*, and it starts from Bengali's writing system. Bengali is written in an alphasyllabary in which diacritics attach on either side of a base character and consonants combine into conjunct forms, producing multi-character grapheme clusters that constitute the writing system's natural units (Alam et al., 2021). In the word *chandra* (moon), for example, the cluster *ndra* is one written character built from five code points: three consonants joined by two virama signs. Bengali handwriting recognition works at this level, recovering grapheme clusters rather than individual code points (Alam et al., 2021); subword tokenization faces the text-side instance of the same segmentation problem. Byte-pair vocabularies trained predominantly on Latin-script text split Bengali into units smaller than its characters (Rust et al., 2021; Petrov et al., 2023); we measure this with standard tokenizer statistics (§4.3). The cut runs deep: under three of the 4 vocabularies the evaluated models share, the average Bengali token spans 2.2 - 2.6 bytes, less than a single three-byte Bengali code point, while the average English token spans 4.4 - 4.9 bytes. The depth is not fixed by the script: it varies twofold across those vocabularies, so fragmentation is a tokenizer choice. A benchmark suite with paired English and Bengali versions of identical content is precisely the setting in which this cost can be quantified, which is what we do here.

Format, the second concept, is a property of the evaluation rather than the language. Benchmark prompts instruct models to answer in a fixed shape (`Correct Answer: [option]` in our setup) and exact-match scoring accepts nothing else: a model that replies "the correct answer is (b)" or reasons before stating the answer is scored wrong despite knowing it. The response error rate (§4.2) measures this failure and §6.1 shows it strikes Bengali and English unequally.

Concretely, we translate 20 widely used English benchmarks into Bengali using a single consistent GPT-4o-mini pipeline, evaluate 8 of them with 10 open instruction-tuned models from the LLaMA, Qwen, Mistral and DeepSeek-R1 families in the range of 3B to 72B parameters, on the paired English and Bengali versions under three metrics: exact-match accuracy, an LLM-judge score and a response-format error rate. The headline gap is real: macro-averaged over models and tasks, English scores exceed Bengali scores by 0.17 - 0.18 on the 0 - 1 score scale under both metrics (LLM-judge: 0.79 vs. 0.63). The paired design also lets us decompose the gap; the decomposition is where the findings lie.

Our contributions are:

- **A uniformly constructed Bengali evaluation suite.** 8 benchmarks covering commonsense, science, math and multidomain knowledge, all translated with one pipeline and released with the pipeline and evaluation code. The suite complements existing Bengali resources: it adds WinoGrande, CommonsenseQA and OpenBookQA, for which we are not aware of public Bengali versions and, unlike prior translated collections assembled from heterogeneous sources, all datasets share one pipeline, prompt format and validation procedure (§3.4).
- **A paired English–Bengali evaluation of 10 open LLMs** across 4 families and three metrics, quantifying how the gap varies with model scale and family (§5).
- **Evidence that exact-match scoring conflates format adherence with capability.** When format-error rates are below 5% in both languages, exact-match and judge-based estimates of the cross-lingual gap agree within about 0.05; when format errors exceed 10%, exact match misstates the gap by 0.09 - 0.24 and can invert its sign. Reasoning-tuned models are penalized hardest. This extends known results on format sensitivity (Sclar et al., 2024; Zheng et al., 2024; Alzahrani et al., 2024) to the cross-lingual setting, where the artifact is asymmetric and therefore biases gap estimates rather than merely adding noise (§6.1).
- **A script-fragmentation characterization.** The script is fixed; the tokenizer decides what its text costs. Bengali fertility averages 6.7–7.4 tokens per word against $\sim$1.3 for English, but that pooled figure hides a twofold spread across the 4 vocabularies the 10 models share (7.9 for Llama 3, 4.0 for Tekken); per-instance sequence lengths inflate by up to 5.5$\times$, and three of the 4 vocabularies produce tokens spanning less than a single Bengali character while Tekken clears that boundary. We report a weak negative association between fragmentation and scores ($r = -0.23$) and are explicit that our evidence does not support a causal claim (§6.2).

2 Related Work

Bengali and Indic evaluation resources. Dedicated Bengali models include BanglaBERT (Bhattacharjee et al., 2022), BanglaGPT (Salim et al., 2023), TituLLM (Nahin et al., 2025) and TigerLLM (Raihan and Zampieri, 2025). On the evaluation side, BenLLMEval assembled a first LLM evaluation for Bengali (Kabir et al., 2023), BEnQA provides parallel English–Bengali science questions from Bangladeshi curricula (Shafayat et al., 2024) and the IndicLLMSuite provides pretraining and fine-tuning data across 22 Indian languages (Khan et al., 2024). Closest to our artifact, Okapi (Lai et al., 2023) released ChatGPT-translated versions of ARC, HellaSwag and MMLU in 26 languages including Bengali, Sarvam AI's Indic-Evals collection (Sarvam AI, 2024) releases Indic translations, including Bengali, of MMLU, BoolQ, ARC-Challenge and GSM8K and Global-MMLU (Singh et al., 2025) includes a Bengali split with partly human-verified translations. Section 3.4 details how our suite relates to these resources.

Bootstrapping benchmarks by translation. Translating English benchmarks is an established first step for languages without native evaluation suites: Turkish LLM development relied on translated benchmarks (Acikgoz et al., 2024) and the Aya project used translation alongside human templating across 100+ languages (Üstün et al., 2024). The approach has known costs, translationese, cultural mismatch and translation errors, that native construction, as in the Khayyam Challenge (Ghahroodi et al., 2024), avoids. We take the translation route with open eyes: it is the only way to obtain *paired* English–Bengali versions of identical items, which our measurement analysis requires (§3.5).

Tokenization and script fragmentation. Fertility and related statistics have long been used to compare tokenizers across languages (Rust et al., 2021) and tokenizer disparities translate into real cost and context-length penalties for non-Latin scripts (Petrov et al., 2023; Ahia et al., 2023). Tokenizer choice measurably affects downstream performance (Ali et al., 2024; Dagan et al., 2024) and subword fragmentation degrades robustness (Chai et al., 2024). For Indic scripts specifically, recent work proposes morphologically and orthographically grounded segmentation (Brahma et al., 2025; Vemula et al., 2025; Libovický and Helcl, 2024) and byte-level modeling for Bangla (Bhattacharyya and Bhattacharya, 2025). The visual counterpart of the segmentation problem is central to Bengali handwritten grapheme recognition (Alam et al., 2021). Our contribution is descriptive: paired statistics locating the fragmentation below the character level.

Evaluation robustness and format sensitivity. LLM scores are highly sensitive to prompt formatting (Sclar et al., 2024), option ordering and selection biases in MCQ (Zheng et al., 2024) and small benchmark perturbations (Alzahrani et al., 2024); prompt-sensitivity benchmarks quantify this variance systematically (Razavi et al., 2025; Polat et al., 2024). Unreliable answer extraction is a recognized failure mode of evaluation harnesses (Yu et al., 2025) and LLM-as-judge protocols are the standard mitigation, with known biases of their own (Zheng et al., 2023). We build on this literature rather than proposing new metrics: in a cross-lingual setting these known artifacts become *asymmetric* between languages, turning benign noise into systematic bias on gap estimates.

3 The Bengali Benchmark Suite

3.1 Benchmark Selection

Starting from the suites used to report results for open model families (Grattafiori et al., 2024), we translated 20 English benchmarks into Bengali (inventory in Appendix A) and selected 8 spanning 4 task categories with well-defined answers that support automatic scoring: **commonsense and reading comprehension** (HellaSwag, WinoGrande, CommonsenseQA, OpenBookQA, BoolQ), **science** (ARC, contributing ARC-Easy and ARC-Challenge splits), **math** (GSM8K) and **multidomain knowledge** (MMLU). Because ARC contributes two evaluation splits, result tables report 9 columns for the 8 benchmarks.

3.2 Translation Pipeline

All datasets were translated with `gpt-4o-mini-2024-07-18`. The model was prompted as a professional translator, instructed to preserve meaning, tone, idioms, proper nouns and, critically, the ground-truth answer values and to emit valid JSON so that translations could be machine-validated (Table 8). Automated checks validated JSON structure, key completeness and answer-key integrity for every item; entries that failed validation or were dropped by the

batched pipeline were detected by log analysis and retranslated until the datasets were complete. Translating all 20 benchmarks cost approximately $200, evidence that this bootstrap is affordable for other underserved languages.

3.3 Translator Selection

Before committing to a translation engine, we compared Google Translate, Azure Translator and `gpt-4o-mini-2024-07-18` in a blind review: reviewers(native Bengali speakers) saw translations of the same source items without knowing which system produced them and indicated which they preferred. GPT-4o-mini was preferred for accuracy and coherence and was used for the full pipeline. We did not log the number of reviewers, use a formal rubric or compute inter-rater agreement, so we treat this as an informal preference test that guided an engineering choice, not as a validation of translation quality (§3.5).

3.4 Position Relative to Existing Resources

Bengali versions of some of these benchmarks already exist: Okapi (Lai et al., 2023) includes ChatGPT-translated Bengali ARC, HellaSwag and MMLU, the Indic-Evals collection (Sarvam AI, 2024) provides Indic translations including Bengali of MMLU, BoolQ, ARC-Challenge and GSM8K, Global-MMLU (Singh et al., 2025) covers Bengali MMLU with partial human verification and BEnQA (Shafayat et al., 2024) provides parallel science QA. Our suite is complementary in three specific ways. First, it adds Bengali versions of WinoGrande, CommonsenseQA and OpenBookQA, for which we are not aware of public Bengali translations. Second, every dataset is produced by one pipeline with one prompt, one engine and one validation procedure, so differences between task scores reflect the tasks rather than heterogeneous translation provenance; this consistency is what makes the cross-task and cross-language analyses in §6 interpretable. Third, because every Bengali item is paired with its English source under identical formatting, the suite supports the controlled gap decomposition and paired tokenization statistics that are this paper's main analytical contributions.

3.5 Translation Quality

The suite is machine-translated and has not undergone item-level human validation; we state this up front rather than in passing. The automated pipeline guarantees structural integrity (well-formed items, intact answer keys, complete datasets) but not semantic fidelity. We therefore frame the suite as a *first-pass instrument with characterized error modes*: suitable for the relative, paired comparisons performed in this paper, where translation noise applies to all models equally and improvable through the released pipeline. Human validation on a stratified sample is the next step; the Limitations section returns to this point.

4 Experimental Setup

4.1 Models

We evaluate 10 open instruction-tuned models drawn from 4 families chosen to span three axes: parameter scale (3B - 72B), documented multilingual support and documented Bengali coverage in pretraining (Table 7, Appendix D). The LLaMA models represent widely deployed open models without documented Bengali pretraining, Qwen represents a multilingual family with documented Bengali exposure, Mistral documents no Bengali support (Mistral Small is multilingual but lists no Bengali) and the DeepSeek-R1 distillations, built on Qwen (14B) and Llama (70B) bases, represent reasoning-tuned models, a category that matters for measurement (§6.1). All models are evaluated zero-shot, without fine-tuning, using fixed prompt templates that instruct the model to answer in the format `Correct Answer: [option]`; full templates and inference details are in Appendix D.

4.2 Task Metrics

We score every model–dataset–language cell with three complementary metrics.

Exact-match accuracy counts a response as correct if it begins with the expected answer string (after normalization such as lowercasing and label mapping). This strictest, most common protocol is deliberately retained because its failure modes are part of what we study.

Response Error Rate (RER) is the fraction of responses that do not begin with *any* valid answer option for the item:

$$\mathrm{RER} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left(\forall\, p \in P_i : \neg\, \mathrm{pre}(r_i, p)\right), \quad (1)$$

where P_i is the set of valid (normalized) answer prefixes for item i and $\mathrm{pre}(r_i, p)$ tests whether response r_i begins with p. Its complement is the Response Adherence Rate,

$$\mathrm{RAR} = 1 - \mathrm{RER}. \quad (2)$$

RER isolates *format* failure from *knowledge* failure: a response with high RER cannot score under exact match regardless of whether the model knew the answer. This makes RER the diagnostic for the answer-extraction failures documented in evaluation-harness studies (Yu et al., 2025).

LLM-judge score is the fraction of responses that a few-shot GPT-based judge rules semantically equivalent to the ground truth, independent of surface format (Zheng et al., 2023). The judge is our estimate of capability with format effects removed; its own reliability limits are discussed in the Limitations section.

4.3 Tokenization Metrics

We characterize each model's tokenizer on the paired corpora using three standard statistics, computed over every prompt (system prompt plus item) in each dataset and language. Although we evaluate 10 models, they share only 4 distinct vocabularies: the Llama 3 tokenizer (the 4 LLaMA models and the DeepSeek-R1 Llama distill), the Qwen 2 tokenizer (both Qwen models and the DeepSeek-R1 Qwen distill) and the two Mistral vocabularies (Mistral 7B's 32k SentencePiece and Mistral Small's Tekken). All tokenizer statistics therefore characterize 4 vocabularies carried by 10 models. **Fertility** is the mean number of tokens per whitespace-separated word (Rust et al., 2021). **Tokens per instance** is the mean tokenized length of a full prompt, the quantity that determines context usage and inference cost (Ahia et al., 2023). **Bytes per token** is the mean UTF-8 byte length of the text divided by its token count (Dagan et al., 2024); higher values mean each token spans more text (coarser, more efficient tokenization), lower values mean finer fragmentation. Earlier drafts used ad hoc names for these quantities; we use the established names for comparability with the tokenization literature.

5 Results

5.1 The Cross-Lingual Gap

Table 1 summarizes performance macro-averaged over the 9 evaluation splits; Tables 4–6 give per-split scores. Averaged over all models, English exceeds Bengali by 0.180 under exact match and 0.169 under the LLM judge, so a substantial capability gap survives even after format effects are removed. The gap narrows sharply with scale within each family (row marginals of Figure 1): the

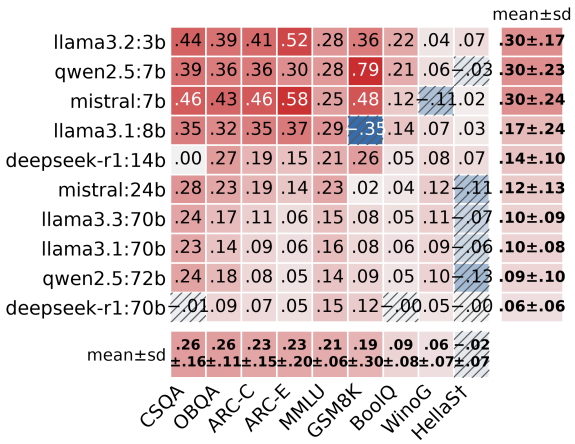


Figure 1: Per-model, per-split English−Bengali LLM-judge gap; rows and columns sorted by mean gap, marginals give mean±sd, hatched cells are negative (Bengali above English) and color encodes the printed value. †The judge is unreliable on HellaSwag (§6.1). The exact-match counterpart is Figure 7 (Appendix C).

LLaMA judge gap falls from 0.304 (3B) to 0.095–0.100 (70B), Qwen from 0.303 to 0.089 and Mistral from 0.299 to 0.125. The best Bengali performer overall is deepseek-r1:70b (judge 0.879), followed by the large Qwen and LLaMA models. Documented Bengali pretraining coverage (Table 7) shows no clear effect at small scale: qwen2.5:7b (documented coverage) and llama3.1:8b (none) appear to differ (judge gaps 0.303 vs. 0.173), but the difference vanishes once GSM8K, a split carrying known harness anomalies (see Limitations), is excluded entirely (0.242 vs. 0.239), so this pool supports no conclusion about coverage either way.

5.2 Where the Gap Concentrates

Figure 1 maps each model's judge gap by split and two regimes emerge: knowledge-heavy multiple choice carries the gap (CSQA, OBQA, ARC and MMLU average 0.21–0.26) while binary BoolQ and WinoGrande stay at or below 0.09. The column marginals separate two further stories: MMLU's gap is nearly model-independent (0.21±0.06), suggesting a property of the language pair rather than of any model, whereas GSM8K's similar mean hides extreme model dependence (0.19±0.30), spanning qwen2.5:7b's 0.79 collapse to llama3.1:8b's −0.35. The row marginals show mean and variance falling together with scale: a large gap is also an erratic gap. The exact-match view of the same matrix (Figure 7, Appendix C) is instead dominated by the format artifacts of §6.1.

	Exact-match accuracy			LLM-judge			RER	
Model	EN	BN	Δ	EN	BN	Δ	EN	BN
llama3.1:8b	0.682	0.401	0.281	0.669	0.496	0.173	0.020	**0.190**
llama3.1:70b	0.885	0.751	0.133	0.843	0.748	0.095	0.007	0.010
llama3.2:3b	0.658	0.322	0.336	0.657	0.353	0.304	0.017	0.064
llama3.3:70b	0.851	0.752	0.099	0.845	0.745	0.100	0.048	0.015
qwen2.5:7b	0.816	0.468	0.348	0.793	0.490	0.303	0.010	0.004
qwen2.5:72b	0.895	0.599	0.295	0.856	0.767	0.089	0.009	**0.247**
mistral:7b	0.170	0.112	0.058	0.627	0.328	0.299	**0.733**	**0.721**
mistral:24b	0.833	0.637	0.196	0.806	0.681	0.125	0.025	0.066
deepseek-r1:14b	0.750	0.515	0.236	0.908	0.767	0.141	**0.140**	**0.212**
deepseek-r1:70b	0.458	0.638	−0.179	0.935	0.879	0.057	**0.494**	**0.200**
Mean	0.700	0.520	0.180	0.794	0.625	0.169	0.150	0.173
Mean (excl. HellaSwag)	0.703	0.540	0.163	0.830	0.638	0.192	0.147	0.157

Table 1: Macro-averages over the 9 evaluation splits, per model and language. Δ = EN − BN. **Bold** marks RER above 0.1; for those five models the two gap estimates (underlined) diverge by 0.09–0.24 and can invert in sign, while for the remaining models the two estimates agree within 0.05. The second mean row excludes HellaSwag, where the judge is unreliable (§6.1). Per-split scores are in Tables 4–6.

6 Analysis

6.1 Format Adherence Confounds Exact-Match Accuracy

The three metrics in Table 1 were chosen so that their disagreements are informative and they disagree systematically. Exact-match accuracy requires both knowing the answer and emitting it in the instructed format; the judge requires only the former; RER measures failure of the latter alone. Prior work established that format sensitivity adds noise to monolingual benchmarks (Sclar et al., 2024; Zheng et al., 2024; Alzahrani et al., 2024). In our paired setting the effect is worse than noise: RER differs *between languages* for the same model, so exact match does not merely blur the cross-lingual gap, it biases it. Shafayat et al. (2024) observe the same phenomenon qualitatively, noting that eliciting format-conforming output is markedly harder in Bengali than in English and fall back to manual answer assessment; our RER quantifies that failure mode at scale. Three patterns in Tables 1 and 5 make the case; Figure 2 previews them.

Apparent incapability that is actually format failure. mistral:7b scores 0.048 exact-match on English OpenBookQA while the judge scores it 0.678; its RER there is 0.914, meaning nine of ten responses never began with a valid option label. Its macro exact-match score (0.170 EN) would place it far below every other model, while its judge score (0.627 EN) sits modestly below the other 7–8B models (0.669–0.793) rather than at the catastrophic floor its exact-match score implies. An earlier draft concluded that Mistral consistently underperforms across languages; the RER column shows that conclusion was substantially a prefix-matching artifact and we correct it here. The genuine Mistral finding is different and appears only under the judge: proportionally, it loses 48% of its English judge score in Bengali (0.627 to 0.328), the largest relative drop among the 7–8B models (38% for qwen2.5:7b, 26% for llama3.1:8b), consistent with its undocumented Bengali support.

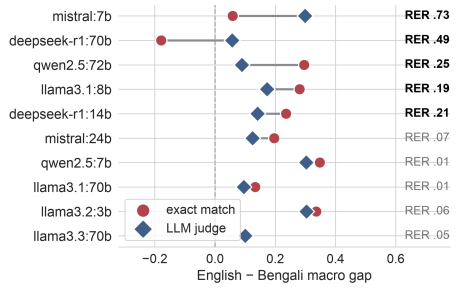


Figure 2: Exact-match vs. LLM-judge estimates of the macro English−Bengali gap per model, sorted by divergence; the right column gives each model's larger macro RER (bold above 0.1). Format failure separates the estimates by up to 0.24 and inverts the sign for deepseek-r1:70b; near-zero-RER models (grey) agree.

Reasoning-tuned models are penalized hardest. The DeepSeek-R1 distillations produce chain-of-thought before the final answer, which frequently breaks the required answer prefix. deepseek-r1:70b has an English RER of 0.494 and an English exact-match of 0.458, against a judge score of 0.935, the highest English judge score in the pool. Exact-match protocols thus rank

the strongest reasoner in the pool as one of the weakest models, a concrete instance of the answer-extraction failures that motivate harness-robustness work (Yu et al., 2025).

Language-asymmetric format failure biases the gap estimate. Because RER is not equal across languages, the bias does not cancel in the EN−BN difference. For qwen2.5:72b, Bengali RER (0.247) far exceeds English RER (0.009) and the exact-match gap (0.295) is more than triple the judge gap (0.089): most of this model's apparent Bengali deficit is formatting. llama3.1:8b shows the same pattern on OpenBookQA, where accuracy collapses from 0.790 (EN) to 0.172 (BN) while RER rises from 0.000 to 0.658. For deepseek-r1:70b the asymmetry runs the other way (EN RER 0.494 vs. BN 0.200) and exact match reports a *negative* gap of −0.179, Bengali above English, where the judge reports a positive gap of 0.057. Across the pool the pattern is monotone: the three models with RER below 0.05 in both languages show gap estimates agreeing within 0.05, the two models with intermediate RER (peaking near 0.07) diverge by at most 0.07 and the five models with RER above 0.1 in either language diverge by 0.09 to 0.24 (deepseek-r1:14b lowest, mistral:7b highest).

Robustness to judge errors. The judge itself is imperfect and its failure is systematic rather than random: on English HellaSwag, seven models score 0.583–0.905 exact match while the judge places them between 0.352 and 0.423; five of those seven have RER at or near zero, so format failure cannot explain the divergence and only the two chain-of-thought DeepSeek-R1 models escape (0.898 and 0.933). Our hypothesis is response-style sensitivity in the judge itself: HellaSwag asks for a continuation; terse label answers are judged not equivalent to the gold continuation while reasoning models restate it in full. The conclusions above do not depend on the split: excluding HellaSwag (second mean row of Table 1), qwen2.5:72b's exact-match gap (0.247) still more than doubles its judge gap (0.116) and the deepseek-r1:70b sign inversion persists (−0.188 vs. +0.064). The aggregate direction is itself unstable: the pooled exact-match gap exceeds the judge gap with HellaSwag included (0.180 vs. 0.169) and falls below it without (0.163 vs. 0.192); the per-split flip is visible by comparing Figures 1 and 7; the lesson is that exact match is unreliable in both directions, with large per-model biases that partially cancel in aggregate. The RER evidence is judge-independent throughout.

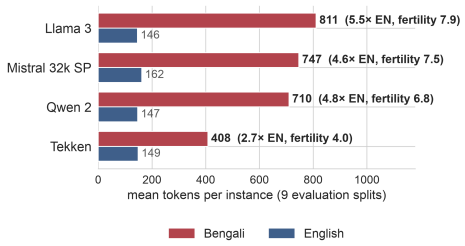


Figure 3: Bengali vs. English tokenization cost by vocabulary: mean tokens per instance over the 9 evaluation splits, annotated with the BN/EN ratio and Bengali fertility. Exact values incl. bytes per token: Table 3 (Appendix B).

The practical conclusion for low-resource evaluation is that a single exact-match number cannot be interpreted as a capability gap: report format-error rates alongside accuracy and treat gap estimates as unreliable wherever RER is non-negligible and asymmetric. This finding is invariant to our translation-quality caveats: translation noise affects both metrics equally; the divergences are metric-internal.

6.2 Script Fragmentation: Fixed Terrain, Variable Cost

The paired corpora let us measure exactly what current subword vocabularies do to Bengali text (Figure 4 and additional plots in Appendix B).

Per-vocabulary cost. Figure 3 is the concrete result: English cost is nearly identical across the 4 vocabularies (146–162 tokens per instance) while Bengali cost varies twofold. Mistral Small's Tekken vocabulary encodes the same Bengali text in half the tokens of Llama 3 (408 vs. 811 per instance; fertility 4.0 vs. 7.9), from a family with no documented Bengali or Indic support and at a vocabulary size nearly identical to Llama 3's (131k vs. 128k): vocabulary composition, not size or model scale, separates Bengali tokenization cost across open models. The absolute cost is a direct inference multiplier: under the dominant Llama 3 vocabulary the same content costs 5.5 times more context in Bengali (BoolQ: 1,115 tokens vs. 196; HellaSwag: 1,359 vs. 256), shrinking the effective context window for Bengali users independent of any accuracy effect. Pooled across models, Bengali fertility is 6.7–7.4 tokens per word against 1.25–1.4 for English (Figure 6), within documented cross-lingual disparities, which reach 15× for some languages (Petrov et al., 2023; Ahia et al., 2023).

Sub-character fragmentation. Every Bengali

code point occupies 3 bytes of UTF-8, so bytes per token tests whether a vocabulary's tokens reach the script's characters. Three of the 4 vocabularies fall below that boundary on Bengali text (2.2–2.6 bytes per token, Table 3, Appendix B): their average token covers *less than one character*. Tekken, at 4.4 bytes, is the one vocabulary that clears the boundary, and it is the same one that halves fertility: a single mechanism, Bengali-capable merges, explains both columns. English tokens span 4.4–4.9 bytes, four to five characters. This sub-character segmentation is the text-side counterpart of the challenge in visual processing of Bengali script (Alam et al., 2021); morphologically grounded Indic tokenization (Brahma et al., 2025; Vemula et al., 2025) and byte-level Bangla modeling (Bhattacharyya and Bhattacharya, 2025) are natural remedies our suite can help evaluate.

Association with performance. Across model–dataset cells, fertility and tokens-per-instance both correlate with the LLM-judge score at $r = -0.23$ (Appendix B). We do not read this as a performance effect. The correlation is weak, it is confounded with model family, scale and task, and it is pseudo-replicated: the 10 models draw on only 4 vocabularies, so the fertility values repeat and the effective sample is far smaller than 90 cells. What the data does support are the direct measurements: the fertility premium, the context-cost multiplier and the sub-character fragmentation, none of which depend on any link to accuracy.

7 Conclusion

We released a uniformly constructed Bengali suite of 8 benchmarks, evaluated 10 open LLMs on paired English and Bengali versions and used the pairing to answer the title's question. *Format* drives the measured gap, not the real one: language-asymmetric format failure lets exact match distort gap estimates threefold to fivefold or invert their sign, with reasoning-tuned models penalized most; what survives format-insensitive judging, roughly 0.17, is real and narrows sharply with scale. *Script fragmentation* is one phenomenon with a fixed terrain and a variable cost: subword tokenizers cut Bengali's grapheme-cluster script below the character, but how deep the cut runs is set by the vocabulary, not the script. Bengali pays a fivefold average fertility premium and its tokens fall below a single character under three of the 4 vocabularies, yet the premium varies twofold by vocabulary (Tekken halves it) and the association with scores stays weak, so vocabulary composition, not the script itself, sets the tax. The recurring lesson: documentation is not measurement; documented pretraining did not predict smaller gaps in this pool and the best Bengali tokenizer came from a family documenting no Bengali support. For practitioners: report format-error rates, corroborate exact match with a format-insensitive metric and expect reasoning-tuned models to be underrated by prefix matching. The suite and pipeline are a first-pass instrument; human validation, natively authored items and grapheme-aware tokenization come next.

Limitations

Our datasets are machine translations that have not undergone systematic item-level human validation, and the engine comparison in §3.3 was an informal preference test. Translation errors, translationese and cultural mismatch may affect absolute Bengali scores; our analyses are paired and relative, which limits this exposure; stratified human validation is the most important future work.

We rely on a single LLM judge as our format-insensitive capability metric and do not corroborate it with a second, independent judge. LLM judges have documented biases and may err differently across languages (Zheng et al., 2023); §6.1 documents one split (English HellaSwag) where the judge disagrees with the gold key at zero RER, an exposure bounded by the exclusion check and judge-independent RER evidence; human agreement over judge verdicts and corroboration with an independent judge remain future work. A few cells score far below documented capability (GSM8K: llama3.1:8b in English, qwen2.5:7b in Bengali), suggesting residual harness effects; released outputs permit auditing them.

Our tokenization analysis is aggregate: we report per-vocabulary bytes per token and fertility, but do not measure how often individual grapheme clusters are split or relate cluster-level segmentation to per-item correctness, which we leave to future work. Finally, the model lineup predates the newest releases: scores age; protocols do not.

Acknowledgements

We thank the Stanford AI Club and Phison Electronics Corporation for collaboration and computational resources.

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.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923, Singapore. Association for Computational Linguistics.

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.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, and 1 others. 2024. Tokenizer choice for LLM training: Negligible or crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3907–3924, Mexico City, Mexico. Association for Computational Linguistics.

Norah Alzahrani, Hisham Abdullah Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yusef Almushaykeh, and 1 others. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13787–13805, Bangkok, Thailand. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327. Association for Computational Linguistics.

Pramit Bhattacharyya and Arnab Bhattacharya. 2025. BanglaByT5: Byte-level modelling for Bangla. *Preprint*, arXiv:2505.17102.

Maharaj Brahma, N J Karthika, Atul Singh, Devaraj Adiga, Smruti Bhate, Ganesh Ramakrishnan, Rohit Saluja, and Maunendra Sankar Desarkar. 2025. MorphTok: Morphologically grounded tokenization for Indian languages. *Preprint*, arXiv:2504.10335.

Yekun Chai, Yewei Fang, Qiwei Peng, and Xuhong Li. 2024. Tokenization falling short: On subword robustness in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1582–1599, Miami, Florida, USA. Association for Computational Linguistics.

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.

Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning

from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 318–327, Singapore. Association for Computational Linguistics.

Jindřich Libovický and Jindřich Helcl. 2024. Lexically grounded subword segmentation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7420, Miami, Florida, USA. Association for Computational Linguistics.

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.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*.

Fina Polat, Ilaria Tiddi, and Paul Groth. 2024. Testing prompt engineering methods for knowledge extraction from text. *Semantic Web*.

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

Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. 2025. Benchmarking prompt sensitivity in large language models. *Preprint*, arXiv:2502.06065.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

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.

Sarvam AI. 2024. Indic-evals: Translated versions of popular LLM benchmarks. Hugging Face collection. Includes MMLU, BoolQ, ARC-Challenge, GSM8K, and TriviaQA-MCQ translated into 10 Indian languages including Bengali.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Sheikh Shafayat, H M Quamran Hasan, Minhajur Rahman Chowdhury Mahim, Rifki Afina Putri, James Thorne, and Alice Oh. 2024. BEnQA: A question answering benchmark for bengali and english. In *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, and 1 others. 2025. Global MMLU: Understanding and addressing cultural and linguistic biases in multilingual evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Saketh Reddy Vemula, Sandipan Dandapat, Dipti Misra Sharma, and Parameswari Krishnamurthy. 2025. Rethinking tokenization for rich morphology: The dominance of unigram over BPE and morphological alignment. *Preprint*, arXiv:2508.08424.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Qingchen Yu, Zifan Zheng, Shichao Song, Zhiyu Li, Feiyu Xiong, Bo Tang, and Ding Chen. 2025. xFinder: Large language models as automated evaluators for reliable evaluation. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, and 1 others. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*.

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.

A Translated Benchmark Inventory

Table 2 lists all 20 benchmarks translated by the pipeline. The 8 evaluated in this paper are OpenBookQA, ARC (Easy/Challenge), BoolQ, CommonsenseQA, GSM8K, HellaSwag, WinoGrande and MMLU; the remainder are released for future work.

B Additional Tokenization Results

The tokens-per-instance columns of Table 3 are pipeline-computed over the full corpora; its fertility and bytes-per-token columns were measured by direct tokenization of Bengali text with each vocabulary's public tokenizer and agree with the pipeline-computed statistics shown here. Figure 5 reports tokens per instance by dataset and language and Figure 6 shows the scatter of fertility and tokens-per-instance against LLM-judge scores underlying the $r = -0.23$ association reported in §6.2.

C Additional Results Figures

Figure 9 shows family-level LLM-judge score distributions and the standard deviation of scores across languages, complementing the per-model macro-averages in Table 1. Figure 8 plots macro scores by model size and language under both metrics. Tables 4–6 give the per-split scores underlying that table.

D Reproducibility Details

Figure 10 gives a schematic overview of the construction and evaluation pipeline.

All models were evaluated zero-shot using their chat/instruct variants. Prompts instruct the model to output the answer as `Correct Answer: [option]` (with a Bengali-language equivalent instruction for Bengali inputs) and responses were scored as described in §4.2. Exact-match normalization lowercases responses and maps label variants (e.g., "Yes" to "yes") before prefix comparison. The LLM judge is a few-shot GPT-based classifier that receives the question, ground truth and model response and returns a Correct/Incorrect verdict. Evaluation uses each benchmark's released evaluation split as listed in Table 2 (the validation split where test labels are not public); exact per-cell counts ship with the released datasets. Tokenization statistics were computed over the concatenation of system prompt and item for every instance in each dataset, per model tokenizer. The released repository contains the translation pipeline, all prompt templates in both languages, inference scripts, judge prompts and the scripts that produce every table and figure in this paper. Datasets are released with train/dev/test splits mirroring their English sources.

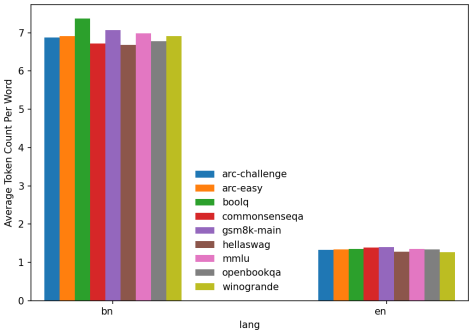
Artifact availability: The translated datasets, translation pipeline and evaluation code are publicly released: code on GitHub and datasets in our Hugging Face collection.

Dataset	Train	Dev	Test	Answer format	Focus
OpenBookQA	4957	500	500	MCQ	Multi-step commonsense reasoning
ARC	3370	869	3548	MCQ	Grade-school science
BoolQ	9427	3270	–	Yes/No	Reading comprehension
CommonsenseQA	9741	1221	1140	MCQ	Commonsense reasoning
GSM8K	7473	–	1319	Number	Grade-school math
HellaSwag	39905	10042	10003	MCQ	Commonsense continuation
WinoGrande	19482	1267	1767	MCQ	Pronoun resolution
MMLU	98487	1528	13869	MCQ	Multidomain knowledge
BigBenchHard	–	–	var.	MCQ	Logical reasoning
AlpacaEval	–	–	10465	Instruction	Instruction following
Anthropic HH	86372	–	35006	Free-form	Safety, helpfulness
APPS	5000	–	5000	Code	Coding
BFCL	–	–	250	Structured	Function calling
Dolly	–	–	7295	Instruction	Varied NLP tasks
HumanEval	–	–	164	Code	Code generation
MATH	8599	–	4999	Exact match	Competition math
MMLU-Pro	–	70	12032	MCQ	Multidomain knowledge
MR-GSM8K	–	–	12024	Exact match	Math meta-reasoning
PIQA	16113	–	3084	MCQ	Physical commonsense
SIQA	33410	1954	–	MCQ	Social reasoning
TruthfulQA	–	–	1634	MCQ	Truthfulness

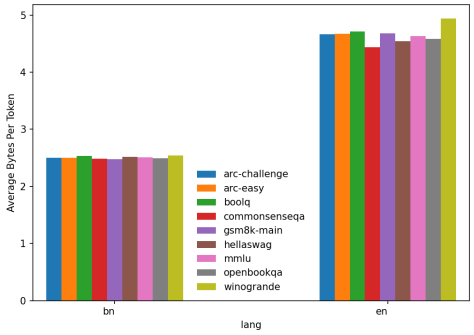
Table 2: All translated benchmarks. The top block lists the 8 evaluated in this paper; the bottom block lists the 12 additional translated benchmarks released for future work.

Vocabulary	Models	BN	EN	BN/EN	Fert.	B/tok
Llama 3	4 LLaMA, R1-70B	811	146	5.5	7.9	2.2
Qwen 2	2 Qwen, R1-14B	710	147	4.8	6.8	2.6
Mistral 32k SP	Mistral 7B	747	162	4.6	7.5	2.3
Tekken	Mistral Small	408	149	2.7	4.0	4.4

Table 3: Per-vocabulary tokenization of Bengali. Tokens per instance (BN, EN, ratio) are pipeline-computed over the full evaluation corpora; Bengali fertility (tokens per word) and bytes per token are measured by direct tokenization of Bengali text with each public tokenizer and agree with the pipeline statistics (Appendix B). A Bengali code point is 3 bytes: only Tekken's tokens clear the character boundary.



(a) Fertility (tokens per word).



(b) Bytes per token.

Figure 4: Paired tokenizer statistics by dataset and language. Bengali fertility is 6.7–7.4 tokens per word against roughly 1.3 for English (a); in (b), under three of the four vocabularies the average Bengali token spans 2.2 - 2.6 bytes, below the 3-byte Bengali code point (model-weighted mean 2.55; Tekken is the exception at 4.4), against 4.4 - 4.9 bytes for English.

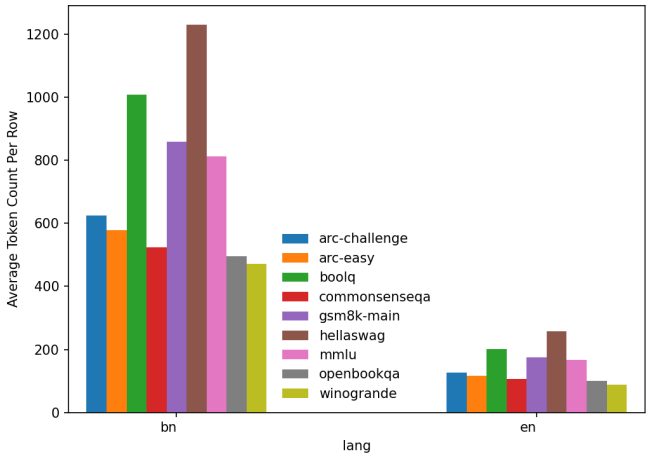
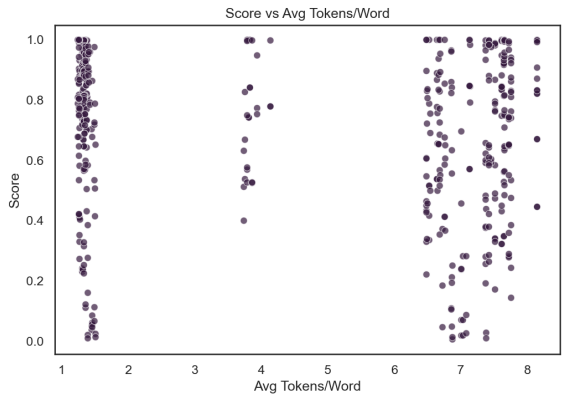
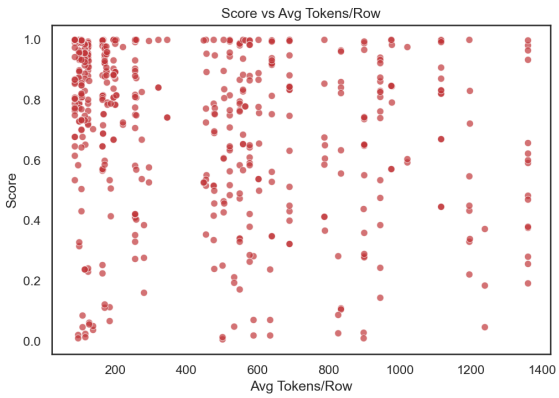


Figure 5: Tokens per instance by dataset and language.



(a) Fertility vs. LLM-judge score.



(b) Tokens per instance vs. LLM-judge score.

Figure 6: Scatter of tokenizer statistics against LLM-judge scores over model–dataset cells. Both associations are weak ($r = -0.23$).

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

Table 4: Exact-match accuracy per evaluation split, English (top block) and Bengali (bottom block).

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

Table 5: Response Error Rate (RER, lower is better) per evaluation split, English (top block) and Bengali (bottom block). Note the strong model and language asymmetries: mistral:7b fails format in both languages, deepseek-r1:70b mainly in English, llama3.1:8b and qwen2.5:72b mainly in Bengali.

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

Table 6: LLM-judge score per evaluation split, English (top block) and Bengali (bottom block).

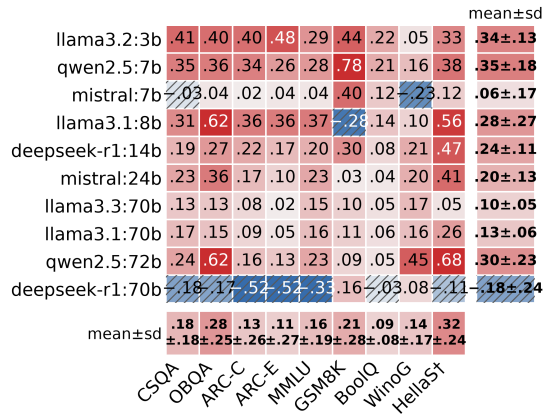


Figure 7: Exact-match counterpart of Figure 1 with identical row and column order. The format artifacts of §6.1 dominate: HellaSwag flips from −0.02 (judge) to +0.32 mean, deepseek-r1:70b's row turns negative because format failure suppresses its English scores and ARC-C/ARC-E swing from consistent judge gaps to standard deviations of 0.26–0.27.

Model	Size	Multiling.	Bengali pretraining
LLaMA 3.1 (Grattafiori et al., 2024)	8B	Limited	✗
LLaMA 3.1 (Grattafiori et al., 2024)	70B	Limited	✗
LLaMA 3.2 (Grattafiori et al., 2024)	3B	Limited	✗
LLaMA 3.3 (Grattafiori et al., 2024)	70B	Limited	✗
Qwen 2.5 (Yang et al., 2025)	7B	Yes	✓
Qwen 2.5 (Yang et al., 2025)	72B	Yes	✓
Mistral (Jiang et al., 2023)	7B	No	✗
Mistral Small (Mistral AI Team, 2025)	24B	Yes	✗
DeepSeek-R1 Distill–Qwen (Guo et al., 2025)	14B	Yes	✓
DeepSeek-R1 Distill–Llama (Guo et al., 2025)	70B	Limited	✗

Table 7: Evaluated models (chat/instruct variants). Bengali coverage follows each model's documentation. The DeepSeek-R1 models are reasoning distillations onto Qwen 2.5 (14B) and Llama 3.3 (70B) bases and inherit those bases' language coverage.

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 and 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 8: Prompt template for English-to-Bengali translation.

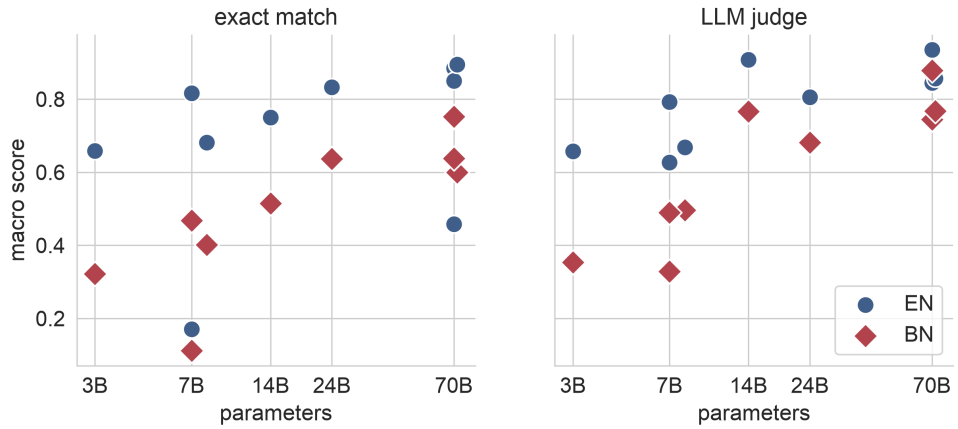
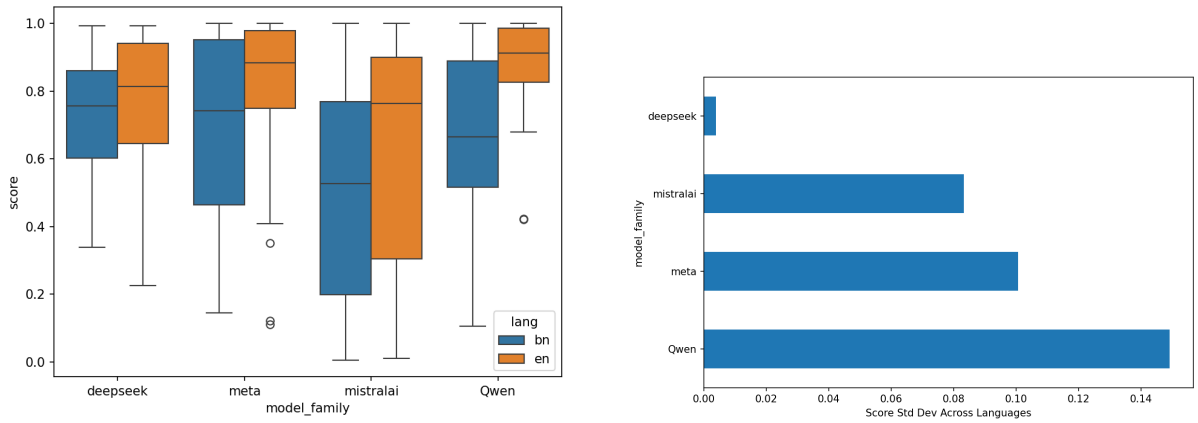


Figure 8: Macro scores by model size and language under exact match (left) and the LLM judge (right). Bengali scores rise faster with scale than English scores; the exact-match panel additionally shows the format-failure outliers of §6.1.



(a) LLM-judge score distributions by model family.

(b) Standard deviation of scores across languages by family.

Figure 9: Family-level score distributions and cross-language robustness. Lower deviation in (b) indicates more stable behavior across English and Bengali; the DeepSeek-R1 family is the most stable, consistent with Table 1.

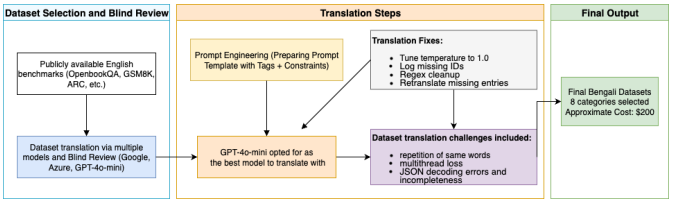


Figure 10: Overview of the construction and evaluation pipeline: benchmark selection, a blind three-way translator comparison, GPT-4o-mini translation with JSON-constrained outputs, automated validation with retranslation and paired English–Bengali evaluation of 10 open LLMs under three metrics.