

How Far Can Current Tokenizers Go in Bangla? A Benchmark and a Morphology-Aware Solution

Gautam Siddharth Kashyap¹, MD Azizul Hoque¹, Kinjalk Srivastava^{2,†}, Pratik Jalan^{1,†}

Subhashis Kar^{3,†}, and Usman Naseem^{1,*}

¹ School of Computing, Macquarie University, Australia

² Information Systems and Digital Business (IWI-HSG), Universität St. Gallen, Switzerland

³ Indian Institute of Information Technology Guwahati, India

* Corresponding Author: usman.naseem@mq.edu.au

† Equal Contributions

Abstract

Bangla (Bengali), spoken by over 300 million people worldwide, presents unique tokenization challenges due to its rich morphological structure. Despite its global importance, Bangla remains underrepresented in mainstream NLP, particularly in low-resource tokenization research. Existing tokenizers such as Byte Pair Encoding (BPE), Unigram, WordPiece, and character-level were originally developed for high-resource languages. These tokenizers struggle with morphologically rich languages and show two main limitations: (i) unnecessary splitting of morphemes and (ii) fragmentation of compound words. Based on these limitations, we hypothesize that morphology-aware tokenization is necessary for handling Bangla effectively. To address this, we first introduce a benchmark that systematically evaluates how widely used tokenizers perform on morphology-related tasks. Motivated by these findings, we then propose BANGLATOKENIZER, a morphology-aware tokenizer that alleviates these limitations using a novel bond-strength framework inspired by molecular structures. Empirically, BANGLATOKENIZER yields consistent downstream improvements, achieving absolute accuracy gains of 2.8%-3.4%, macro-F1 gains of 2.5%-3.6%, and perplexity reductions of up to 20% across diverse language models and tasks. These improvements are accompanied by 5%-10% higher inference throughput, showing that morphology-aware tokenization improves both effectiveness and efficiency.

1 Introduction

Bangla (Bengali), spoken by over 300 million people worldwide (Noor et al., 2023), poses significant challenges for tokenization due to its rich morphological complexity. It features productive inflection¹ (e.g., *kitab* book vs. *kitab-er* of the

¹Inflection refers to modifying a word to express grammatical categories such as case, tense, or number.

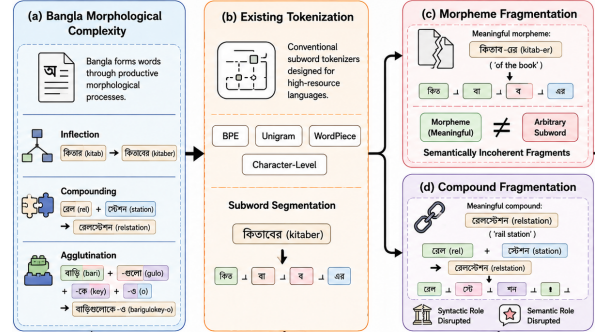


Figure 1: Illustration of tokenization failures in Bangla due to inflection, compounding, and agglutination.

book), compounding² (e.g., *relstation*, rail station), and agglutinative³ forms (e.g., *barigulokay-o*, even to the houses), resulting in a highly diverse and sparse vocabulary.

Existing tokenizers, originally developed for high-resource languages, segment words into smaller units to balance vocabulary size and coverage. These include Byte Pair Encoding (BPE) (Sennrich et al., 2016), Unigram (Kudo, 2018), WordPiece (Wu et al., 2016), and character-level tokenizers (Ling et al., 2015). However, these tokenizers struggle to maintain morphemic integrity in Bangla, showing two main limitations (see Figure 1): (i) unnecessary splitting of morphemes into semantically incoherent fragments, and (ii) fragmentation of compound words that disrupt their syntactic and semantic roles.

Despite growing interest in Bangla NLP, especially as a low-resource language, no standardized benchmark exists to evaluate tokenizers' ability to preserve morphological structure. To address this gap, we first introduce the comprehensive benchmark for morphological tokenization evaluation in Bangla. Based on insights from

²Compounding is the process of combining two or more words to form a new word with a unified meaning.

³Agglutination involves adding multiple affixes to a word stem, each representing a specific grammatical function.

this benchmark, we then propose BANGLATOKENIZER, a morphology-aware tokenizer inspired by molecular bonding principles. BANGLATOKENIZER models morphemes as atomic units and estimates bond strengths⁴ using frequency-weighted Pointwise Mutual Information (PMI), allowing selective merging or splitting to preserve meaningful morphemic structures. Extensive intrinsic and extrinsic evaluations demonstrate that BANGLATOKENIZER outperforms existing tokenizers by producing more morphologically faithful token representations. In summary, our contributions are as follows:

- We propose the benchmark specifically designed to evaluate morphological tokenization in Bangla, a low-resource language. We also introduce BANGLATOKENIZER, a novel morphology-aware tokenizer leveraging a bond-strength model.
- Empirically, BANGLATOKENIZER yields consistent downstream improvements, achieving absolute accuracy gains of 2.8%-3.4%, macro-F1 gains of 2.5%-3.6%, and perplexity reductions of up to 20% across diverse language models and tasks. These gains are accompanied by 5%-10% higher inference throughput, demonstrating that morphology-aware tokenization improves both effectiveness and efficiency.

2 Related Works

Recent advances in Bangla NLP have produced pretrained models such as Banglallama (Zehady et al., 2026), BanglaBERT (Bhattacharjee et al., 2023), BanglaT5 (Bhattacharjee et al., 2023), and SahajBERT (Diskin et al., 2021), which capture syntactic and semantic properties of Bangla effectively. However, these models rely on tokenizers originally designed for high-resource languages like English, causing their vocabularies to misrepresent Bangla’s rich morphology and resulting in excessive segmentation and degraded downstream performance. Tokenization plays a central role in NLP by balancing vocabulary size and linguistic coverage, and morphology-oriented models such as Morfessor (Creutz and Lagus, 2006)

⁴This chemistry-inspired motivation is based on the analogy between morphological structures and molecular formations (Stepanov et al., 2024). In chemistry, atoms are connected by bonds of varying strengths that determine molecular stability (Stepanov et al., 2024). Similarly, morphemes in Bangla are connected with different degrees of cohesion.

and alignment-based segmentation (Yadav et al., 2026; Li et al., 2025) attempted to preserve morphemic structure but remained largely confined to well-resourced settings. In contrast, widely adopted subword tokenizers, including BPE (Sennrich et al., 2016), Unigram (Kudo, 2018), and WordPiece (Wu et al., 2016), generalize well across languages but struggle with morphologically complex languages like Bangla, often fragmenting morphemes and compound words into semantically incoherent units, as discussed in Section 1. To address these limitations, recent work has explored morphology-aware tokenization, such as MorphBPE (Asgari et al., 2025), which incorporates morphological boundaries and consistency constraints, and MoVoC (Teklehaimanot et al., 2025), which combines morpheme-based and subword vocabularies to retain frequent morphological units. While these models improve linguistic fidelity, prior studies (Sälevä and Lignos, 2021) indicate that morphology-aware segmentation does not consistently outperform purely statistical tokenization models, motivating the need for more principled mechanisms to model morphemic cohesion in low-resource languages like Bangla.

3 Methodology

Our methodology comprises two components designed to evaluate and address morphological limitations in Bangla tokenization. First, we introduce a benchmark that systematically assesses the behavior of widely used tokenizers on morphology-sensitive Bangla tasks. Second, informed by insights from this benchmark, we propose BANGLATOKENIZER, a morphology-aware tokenizer that models morphemic cohesion using a novel bond-strength framework inspired by molecular structures. We describe each component in detail below.

3.1 Tokenization Benchmarking

We design a tokenizer-centric benchmarking framework to address morphological limitations in Bangla. The benchmark is instantiated on two Bangla datasets: **BanglaTense** (Bijoy et al., 2025) and **BanglaParaphrase** (Akil et al., 2022), from which we extract a corpus $\mathcal{D} = \{(s_i, \hat{y}_i)\}_{i=1}^N$, where each s_i is a Bangla sentence and $\hat{y}_i$ is its task-specific label corresponding to tense class, noun number, or pronoun category. Each sentence s_i is represented as an ordered sequence of surface

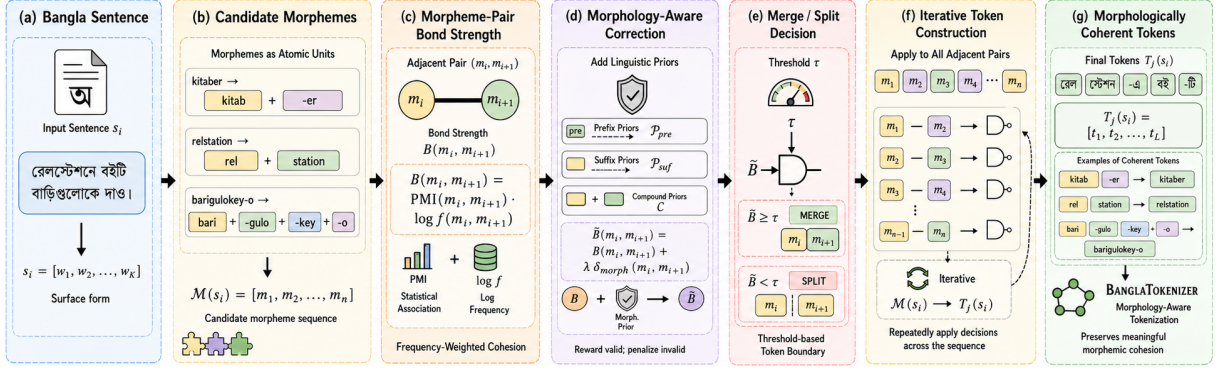


Figure 2: Overview of the BANGLATOKENIZER pipeline. Sentences are first decomposed into candidate morphemes, after which adjacent morpheme pairs are selectively merged using frequency-weighted PMI-based bond strength and morphology-aware priors, producing morphologically coherent subword tokens.

words $W(s_i) = [w_{i,1}, w_{i,2}, \dots, w_{i,K_i}]$, and its label $\hat{y}_i$ is functionally determined by morphology-bearing subword units realized through suffixation and compounding. Given a tokenizer T_j , each sentence is mapped to a sequence of subword tokens $T_j(s_i) = [t_{i,1}^{(j)}, t_{i,2}^{(j)}, \dots, t_{i,K_i^{(j)}}^{(j)}]$, where $K_i^{(j)}$ denotes the number of tokens produced by T_j for s_i . To facilitate controlled comparison, the default tokenizer τ_l of each downstream language model $\mathcal{M}_l$ is replaced by an external tokenizer T_j , such that the model input mapping is redefined via Equation (1), where $T_j(s_i) = [t_{i,1}^{(j)}, \dots, t_{i,K_i^{(j)}}^{(j)}]$ is the token sequence produced by T_j and $E_l : \mathcal{V}_j \rightarrow \mathbb{R}^d$ denotes the embedding lookup over the tokenizer-specific vocabulary $\mathcal{V}_j$ aligned to $\mathcal{M}_l$.

$$x_i^{(j)} = E_l(T_j(s_i)) \quad (1)$$

The resulting representations $\{x_i^{(j)}\}_{i=1}^N$ constitute the sole input to the model for both intrinsic evaluation, where statistics are computed directly over $\{T_j(s_i)\}_{i=1}^N$ to quantify morphological fragmentation, and extrinsic evaluation, where $x_i^{(j)}$ is paired with the label $\hat{y}_i$ and optimized under identical model parameters and training settings.

3.2 BANGLATOKENIZER

Based on the tokenizer-centric benchmarking framework defined above, we specify a particular tokenizer T_j , termed BANGLATOKENIZER (see Figure 2), designed to preserve Banglas inflectional, agglutinative, and compound morphology. Unlike standard subword tokenizers that define T_j solely through frequency-driven merge operations over surface strings, BANGLATOKENIZER instantiates T_j by modeling the cohesion between

morphology-bearing units within each sentence $s_i \in \mathcal{D}$. For a given sentence s_i , we first obtain a sequence of candidate morphemes $\mathcal{M}(s_i) = [m_1, m_2, \dots, m_n]$, where each m_k corresponds to a linguistically motivated atomic unit such as a stem, suffix, or compound constituent. The tokenizer treats these morphemes as atomic units and determines token boundaries by estimating the cohesion between each adjacent morpheme pair (m_i, m_{i+1}) using a bond-strength formulation inspired by molecular interactions. Specifically, the statistical bond strength is defined via Equation (2), where $f(m_i, m_{i+1})$ denotes the corpus frequency of the morpheme pair occurring contiguously and $\text{PMI}(m_i, m_{i+1}) = \log \frac{P(m_i, m_{i+1})}{P(m_i)P(m_{i+1})}$ measures their statistical association.

$$B(m_i, m_{i+1}) = \text{PMI}(m_i, m_{i+1}) \cdot \log f(m_i, m_{i+1}) \quad (2)$$

The logarithmic frequency term down-weights spurious high-PMI pairs that occur rarely, yielding a frequency-weighted PMI that more reliably captures morphemic cohesion; ablations isolating this purely statistical variant are analyzed in Section 6. However, statistical association alone is insufficient for Bangla, where affixation and compounding obey strict grammatical constraints. To incorporate linguistic knowledge, we introduce morphology-aware priors over valid prefixes $\mathcal{P}_{\text{pre}}$, suffixes $\mathcal{P}_{\text{suf}}$, and compound morpheme pairs $\mathcal{C}$, which adjust the bond strength via a correction term $\tilde{B}(m_i, m_{i+1}) = B(m_i, m_{i+1}) + \lambda \cdot \delta_{\text{morph}}(m_i, m_{i+1})$, where λ controls the relative influence of linguistic priors and δ_{morph} rewards morphologically valid affix and compound formations while penalizing invalid merges. Ablation variants that remove affix- and compound-aware rules from δ_{morph} demonstrate increased compound fragmen-

tation, showing the importance of this component (see Section 6).

The final segmentation decision is governed by a threshold τ applied to the adjusted bond strength, such that adjacent morphemes are merged if $\tilde{B}(m_i, m_{i+1}) \geq \tau$ and split otherwise. Applying this decision iteratively over $\mathcal{M}(s_i)$ yields the final subword sequence $T_j(s_i)$. Sensitivity analysis over τ reveals that low thresholds induce over-segmentation while high thresholds suppress valid morpheme boundaries, with an intermediate value achieving the best trade-off between linguistic fidelity and granularity; detailed threshold ablations are reported in Section 6. These empirically driven components allow BANGLATOKENIZER to avoid the systematic over-segmentation of generic subword methods while retaining sufficient flexibility to model productive Bangla morphology.

4 Experimental Setup

We evaluate four widely used tokenizers: BPE (Sennrich et al., 2016), Unigram Language Model (Kudo, 2018), WordPiece (Wu et al., 2016), and Character-level tokenization (Ling et al., 2015), and additionally include morphology-aware baselines, namely Morfessor (Creutz and Lagus, 2006), MorphBPE (Asgari et al., 2025), and MoVoC (Teklehaymanot et al., 2025). All tokenizers are compared directly against BANGLATOKENIZER. Each tokenizer is paired with a set of pretrained language models $\mathcal{M}_l$ spanning diverse pretraining objectives and parameter scales. These include instruction-tuned models (e.g., LLaMA3.2-1B⁵), multilingual models (e.g., BLOOMZ-1B7⁶, and mT5-Base⁷), and Bangla-specific backbones (e.g., Bangla-s1k-Qwen-2.5-3B-Instruct⁸).

4.1 Dataset

We evaluate BANGLATOKENIZER on two morphologically annotated Bangla datasets (see Table 1): **BanglaTense** (Bijoy et al., 2025) and **BanglaParaphrase** (Akil et al., 2022), which together cover three morpho-syntactic classification tasks: tense, noun number, and pronoun. **BanglaTense** contains 17,819 sentences (106K to-

Dataset	BanglaTense	BanglaParaphrase	BanglaParaphrase
Task	Tense	Noun Number	Pronoun/Person
#Sent	17,819	10,336	10,337
#Tokens	106K	124K	124K
Avg Sent.	5.9	12.0	12.0
Split	80/10/10	80/10/10	80/10/10

Table 1: Summary statistics of the evaluation datasets used for intrinsic and extrinsic analysis. Sent. refers to Sentences.

kens; 5.9 tokens per sentence on average), each labeled as past, present, or future, where tense is primarily expressed through inflectional suffixes attached to verb stems. From **BanglaParaphrase**, we extract two morphology-focused subsets totaling 20,673 sentences (248K tokens; 12 tokens per sentence on average), each split into 80/10/10 train/validation/test.

4.2 Evaluation Metrics

We evaluate tokenizers using a unified set of intrinsic and extrinsic metrics that quantify linguistic fidelity and computational efficiency under the same input representations. Given a tokenizer T_j , intrinsic evaluation is performed directly over the tokenized corpus $\{T_j(s_i)\}_{i=1}^N$ to assess how well token boundaries align with Bangla morphological structure. Let $t_i = |T_j(s_i)|$ and $w_i = |W(s_i)|$ denote the number of tokens and surface words in sentence s_i , respectively. We compute Fertility (F) (Rust et al., 2021), defined as $F = \frac{\sum_{i=1}^N t_i}{\sum_{i=1}^N w_i}$, which measures token compactness; Proportion of Continued Words (POCW) (Rust et al., 2021), defined as $POCW = \frac{\# \text{continued words}}{\# \text{total words}}$, which captures the frequency of word fragmentation; Type-Token Ratio (TTR) (Bentz et al., 2015), defined as $TTR = \frac{\# \text{unique tokens}}{\# \text{total tokens}}$, which reflects lexical diversity; and MorphScore (MS) (Arnett and Bergen, 2025), defined as $MS = \frac{1}{N} \sum_{i=1}^N \frac{\# \text{morpheme-consistent tokens in } T_j(s_i)}{\# \text{tokens in } T_j(s_i)}$, which measures the proportion of tokens that align with valid morphemic boundaries. Lower Fe and POCW indicate better preservation of morpheme-level structure, while higher TTR and MS suggest richer and more morphologically coherent token vocabularies.

For extrinsic evaluation, the same tokenized sequences are embedded as $x_i^{(j)} = E_l(T_j(s_i)) \in \mathbb{R}^d$ and constitute the sole inputs to a fixed downstream language model, where $x_i^{(j)}$ is paired with the task label $\hat{y}_i$ and optimized under identical model parameters and training settings. Computa-

⁵<https://huggingface.co/meta-llama/Llama-3.2-1B>

⁶<https://huggingface.co/bigscience/bloomz-1b7>

⁷<https://huggingface.co/google/mt5-base>

⁸<https://huggingface.co/BanglaLLM/Bangla-s1k-qwen-2.5-3B-Instruct>

tional efficiency is measured using Cost per Token (CPT) (Ali et al., 2024), $CPT = \frac{\text{Total Processing Time}}{\sum_{i=1}^N t_i}$, Cost per Word (CPW) (Ali et al., 2024), $CPW = \frac{\text{Total Processing Time}}{\sum_{i=1}^N w_i}$, and energy consumption (Ali et al., 2024) normalized per token, $EC_{\text{per token}} = \frac{\text{Total Energy Used}}{\sum_{i=1}^N t_i}$. Lower CPT, CPW, and energy values indicate more efficient tokenization.

4.3 Hyperparameters

All experiments use fixed hyperparameters across tokenizers for fair comparison. For BANGLATOKENIZER, the bond-strength threshold τ governing merge decisions is tuned over $[0.3, 1.0]$, with $\tau = 0.65$ selected based on validation performance to balance over- and under-segmentation, while the morphology-aware weighting parameter λ controlling the contribution of the correction term δ_{morph} is fixed at 0.6, reflecting a moderate emphasis on linguistic priors as supported by ablation analysis (see Section 6). Prior to tokenization, sentences are segmented into candidate morphemes using a hybrid analyzer based on the BanglaMorph toolkit (Bhattacharjee et al., 2023), combining an 18K-lemma lexicon with verified affix patterns and finite-state morphophonemic rules, yielding the morpheme sequences $\mathcal{M}(s_i)$ used by the bond-strength model. For extrinsic evaluation, all tokenized inputs are processed by a fixed downstream language model with a lightweight projection head ψ , implemented as a two-layer feed-forward network with ReLU activation and dropout 0.2, and optimized using Adam with learning rate 1×10^{-4} , $\beta_1 = 0.9$, $\beta_2 = 0.999$, and early stopping based on validation loss.

5 Result Analysis

5.1 Intrinsic Analysis

Table 2 provides intrinsic evidence supporting our hypothesis from Section 1 that morphology-aware tokenization is necessary for Bangla. Across all datasets, generic subword tokenizers show substantial morphological fragmentation: character-level tokenization yields very high Fertility (e.g., $F > 5.2$) and near-maximal POCW (≈ 0.99), indicating extreme over-segmentation, while Unigram and WordPiece reduce Fertility but still suffer from low TTR (often < 0.03) and weak MorphScore values, reflecting fragmented and redundant token inventories with limited morphemic coherence. Morphology-aware baselines improve

Tokenizer	F ↓	TTR ↑	POCW ↓	MS ↑	ΔAvg ↓
BanglaTense					
BPE	2.064	0.097	0.739	0.641	4.9
Unigram	3.085	0.008	0.699	0.518	6.4
Character-Level	5.287	0.001	0.998	0.214	8.0
WordPiece	2.878	0.022	0.754	0.552	5.9
Morfessor	2.341	0.061	0.781	0.673	4.7
MorphBPE	2.012	0.124	0.702	0.781	3.0
MoVoC	1.941	0.138	0.691	0.806	2.4
BanglaTokenizer	1.853	0.177	0.853	0.918	1.1
BanglaParaphrase (Noun/Number)					
BPE	2.183	0.104	0.751	0.628	4.8
Unigram	3.407	0.007	0.795	0.497	6.7
Character-Level	5.477	0.001	0.990	0.201	8.0
WordPiece	2.774	0.022	0.673	0.541	5.6
Morfessor	2.402	0.073	0.782	0.661	4.6
MorphBPE	1.934	0.168	0.704	0.794	2.8
MoVoC	1.862	0.183	0.691	0.822	2.3
BanglaTokenizer	1.729	0.251	0.729	0.936	1.0
BanglaParaphrase (Pronoun/Person)					
BPE	2.109	0.106	0.749	0.633	4.7
Unigram	3.106	0.008	0.711	0.506	6.3
Character-Level	5.241	0.001	0.992	0.209	8.0
WordPiece	2.621	0.025	0.645	0.559	5.4
Morfessor	2.278	0.079	0.768	0.684	4.5
MorphBPE	1.921	0.154	0.702	0.788	2.9
MoVoC	1.847	0.169	0.688	0.815	2.4
BanglaTokenizer	1.758	0.220	0.758	0.924	1.2

Table 2: Intrinsic evaluation of tokenizers on Bangla morphological tasks using Fertility (F), Type-Token Ratio (TTR), Proportion of Continued Words (POCW), and MorphScore (MS). ΔAvg denotes the average rank across all intrinsic metrics and tasks (lower is better), summarizing overall morphological fidelity.

these trends: Morfessor reduces Fertility relative to BPE but remains prone to affix over-splitting, whereas MorphBPE and MoVoC achieve more balanced trade-offs (e.g., $F \approx 1.9$ -2.0, $TTR \approx 0.14$ -0.18, $MS \approx 0.78$ -0.82). BANGLATOKENIZER consistently achieves the strongest intrinsic profile across all tasks, with the lowest Fertility (down to $F = 1.73$), the highest lexical diversity (TTR up to 0.25), the highest MorphScore (up to 0.93), and controlled POCW, indicating compact yet linguistically coherent segmentation. These results further support our hypothesis that BANGLATOKENIZERs bond-strength-based merging with linguistic priors yields more morphologically faithful tokenizations than existing subword and morphology-aware methods.

5.2 Extrinsic Analysis

Across the **BanglaTense** task (see Table 3), BANGLATOKENIZER consistently achieves the lowest computational overhead across all pre-trained backbones, supporting our hypothesis that preserving inflectional morphemes reduces redundant token processing. On mT5-base, the cost per word (CPW) drops from 9.59 with BPE and 4.97 with MoVoC to 2.20, while cost per token

Tokenizer	EC ↓	CPT ↓	CPW ↓	ΔAvg ↓
mT5-base				
BPE	8.74	9.60	9.59	6.7
Unigram	8.79	5.83	9.60	6.1
Character-Level	8.86	2.05	8.06	5.3
WordPiece	8.08	6.38	9.60	6.0
Morfessor	8.42	4.71	7.88	4.8
MorphBPE	7.98	3.12	5.44	3.2
MoVoC	7.64	2.88	4.97	2.6
BanglaTokenizer	7.22	2.65	2.20	1.0
LLaMA3.2-1B				
BPE	8.88	48.08	48.06	7.3
Unigram	9.05	31.59	48.08	6.4
Character-Level	9.52	15.05	41.35	5.1
WordPiece	8.76	33.98	48.08	6.6
Morfessor	9.01	29.47	42.30	5.8
MorphBPE	8.67	21.94	30.71	3.6
MoVoC	8.43	19.02	26.48	2.9
BanglaTokenizer	8.92	17.70	15.70	1.4
Bloomz-1B7				
BPE	8.26	77.49	77.46	7.5
Unigram	8.88	51.27	77.49	6.6
Character-Level	11.02	24.99	66.79	5.4
WordPiece	8.91	55.08	77.48	6.9
Morfessor	9.20	48.30	69.52	6.1
MorphBPE	8.74	36.82	52.21	3.8
MoVoC	8.51	33.14	47.08	3.1
BanglaTokenizer	10.59	29.19	26.02	1.6
Bangla-s1k-Qwen-2.5-3B-Instruct				
BPE	8.46	100.00	99.96	7.8
Unigram	9.52	66.34	100.00	6.9
Character-Level	12.07	32.60	86.28	5.7
WordPiece	10.76	71.23	99.99	7.2
Morfessor	10.94	62.48	91.30	6.4
MorphBPE	10.12	46.90	66.18	4.1
MoVoC	9.78	42.31	59.44	3.4
BanglaTokenizer	13.33	38.00	33.92	1.9

Table 3: Extrinsic efficiency analysis on the BanglaTense task using Energy Consumption (EC), Cost per Token (CPT), and Cost per Word (CPW). ΔAvg denotes the average rank across EC, CPT, and CPW within each model (lower is better).

(CPT) is reduced by more than 70% relative to BPE (9.60 → 2.65). Similar gains are observed on larger models such as Bangla-s1k-Qwen-2.5-3B-Instruct, where CPW decreases from 99.96 to 33.92. These improvements stem from bond-strength-based merging of tense-bearing suffixes, which limits over-segmentation and shortens effective sequence lengths without removing morphologically informative units.

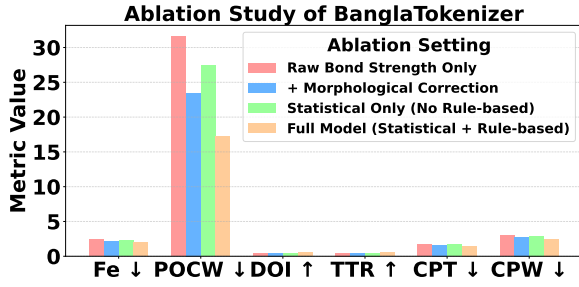
On the **BanglaParaphrase noun number** task (see Table 4), which relies heavily on plural suffixes and compound noun constructions, BANGLATOKENIZER again yields the lowest EC, CPT, and CPW across all models. For LLaMA3.2-1B, CPW is reduced from 52.26 (BPE) and 29.52 (MoVoC) to 17.64, while CPT decreases from 52.30 to 19.96. Even relative to prior morphology-aware baselines such as MorphBPE and MoVoC, BANGLATOKENIZER provides an additional 10%-15% reduction in computational cost. This effect reflects the incorporation of morphology-aware

Tokenizer	EC ↓	CPT ↓	CPW ↓	ΔAvg ↓
BanglaParaphrase (Noun Number)				
mT5-base				
BPE	8.92	11.10	11.05	7.1
Unigram	8.97	6.72	11.08	6.3
Character-Level	9.14	2.48	9.01	5.4
WordPiece	8.35	7.21	11.06	6.2
Morfessor	8.66	5.39	8.91	4.9
MorphBPE	8.21	3.86	6.42	3.4
MoVoC	7.89	3.52	5.86	2.7
BanglaTokenizer	7.46	3.18	2.74	1.0
LLaMA3.2-1B				
BPE	9.14	52.30	52.26	7.4
Unigram	9.31	34.87	52.30	6.6
Character-Level	9.88	16.92	44.78	5.3
WordPiece	9.02	37.10	52.29	6.8
Morfessor	9.28	32.14	45.91	5.9
MorphBPE	8.94	24.86	33.98	3.7
MoVoC	8.70	21.41	29.52	2.9
BanglaTokenizer	9.08	19.96	17.64	1.4
Bloomz-1B7				
BPE	8.47	83.10	83.05	7.6
Unigram	9.05	55.92	83.09	6.7
Character-Level	11.34	27.46	72.98	5.6
WordPiece	9.08	59.74	83.06	6.9
Morfessor	9.43	51.62	75.68	6.2
MorphBPE	8.96	39.98	56.90	3.9
MoVoC	8.74	36.12	51.44	3.2
BanglaTokenizer	10.92	32.85	29.71	1.6
Bangla-s1k-Qwen-2.5-3B-Instruct				
BPE	8.79	108.40	108.32	7.9
Unigram	9.81	72.28	108.39	7.0
Character-Level	12.64	36.74	94.56	5.8
WordPiece	11.21	77.88	108.36	7.3
Morfessor	11.38	68.21	99.47	6.5
MorphBPE	10.56	52.44	72.33	4.2
MoVoC	10.18	47.86	65.24	3.5
BanglaTokenizer	13.98	43.72	39.11	1.9

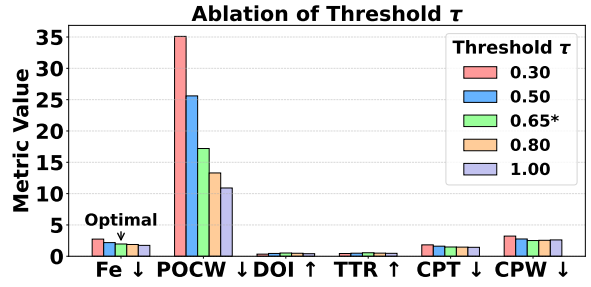
Table 4: Extrinsic efficiency analysis on the BanglaParaphrase *noun number* task using Energy Consumption (EC), Cost per Token (CPT), and Cost per Word (CPW). ΔAvg denotes the average rank across EC, CPT, and CPW within each model (lower is better).

priors into the bond-strength formulation, which stabilizes plural and compound noun segmentation and prevents unnecessary boundary insertion.

For the **BanglaParaphrase pronoun** task (see Table 5), where frequent but short morphemes make efficiency highly sensitive to token boundary placement, BANGLATOKENIZER consistently outperforms all baselines. On mT5-base, CPW is reduced to 2.55 compared to 8.72 for character-level tokenization and 5.49 for MoVoC, while on LLaMA3.2-1B it decreases from 27.86 to 16.78. These gains arise from selectively merging pronominal morphemes using frequency-weighted PMI while maintaining grammatical compatibility, supporting our hypothesis that modeling morphemic cohesion improves both efficiency and robustness beyond purely statistical or rule-only tokenization models.



(a) Ablation investigation of the BANGLATOKENIZER in various configurations. The complete model strikes an optimal balance between linguistic fidelity and segmentation errors.



(b) Threshold ablation of bond strength parameter τ . The optimal value $\tau = 0.65$ achieves the best trade-off between granularity and morphological fidelity.

Figure 3: Visual analysis of the BANGLATOKENIZER’s performance under architectural (left) and hyperparameter (right) variations. Fe refers to Fertility.

Tokenizer	EC ↓	CPT ↓	CPW ↓	Δ Avg ↓
BanglaParaphrase (Pronoun/Person)				
mT5-base				
BPE	8.88	10.45	10.41	7.2
Unigram	8.93	6.28	10.44	6.4
Character-Level	9.01	2.32	8.72	5.3
WordPiece	8.22	6.88	10.43	6.5
Morfessor	8.54	5.01	8.56	4.9
MorphBPE	8.11	3.44	6.01	3.5
MoVoC	7.78	3.19	5.49	2.7
BanglaTokenizer	7.35	2.98	2.55	1.0
LLaMA3.2-1B				
BPE	9.03	50.61	50.58	7.5
Unigram	9.18	33.12	50.60	6.7
Character-Level	9.71	16.08	43.12	5.4
WordPiece	8.91	35.34	50.59	6.9
Morfessor	9.16	30.26	44.21	5.9
MorphBPE	8.81	23.41	32.10	3.8
MoVoC	8.56	20.18	27.86	2.9
BanglaTokenizer	8.97	18.91	16.78	1.3
Bloomz-1B7				
BPE	8.39	80.72	80.68	7.7
Unigram	8.96	54.10	80.71	6.8
Character-Level	11.18	26.14	70.48	5.7
WordPiece	8.98	57.93	80.69	7.0
Morfessor	9.34	49.74	73.24	6.3
MorphBPE	8.87	38.42	54.60	4.0
MoVoC	8.65	34.58	49.23	3.2
BanglaTokenizer	10.76	31.27	28.39	1.6
Bangla-s1k-Qwen-2.5-3B-Instruct				
BPE	8.66	105.92	105.86	7.9
Unigram	9.67	70.24	105.90	7.0
Character-Level	12.38	35.18	92.17	5.9
WordPiece	11.04	75.79	105.88	7.3
Morfessor	11.22	66.10	97.04	6.5
MorphBPE	10.41	50.21	69.82	4.3
MoVoC	10.05	45.86	63.11	3.6
BanglaTokenizer	13.71	41.02	36.64	1.9

Table 5: Extrinsic efficiency analysis on the BanglaParaphrase *pronoun* task using Energy Consumption (EC), Cost per Token (CPT), and Cost per Word (CPW). Δ Avg denotes the average rank across EC, CPT, and CPW within each model (lower is better).

6 Ablation Studies

We perform empirical ablations to analyze the contribution of individual components in BANGLATOKENIZER, with results summarized in Table 6 and

Ablation	F ↓	MS ↑	DOI ↑	Δ Avg ↓
A1: Noisy Morpheme Initialization	2.94	0.621	0.704	6.8
A2: Reduced Lexicon Coverage	2.61	0.683	0.748	5.9
A3: Cross-Domain Threshold	2.48	0.712	0.781	5.1
A4: Human-Validated Benchmark	1.92	0.874	0.891	2.4
A5: Noisy + Code-Switched Text	3.18	0.548	0.662	7.4
A6: Weighted PMI Without Priors	2.27	0.781	0.836	3.6
A7: Tokenization Overhead Analysis	1.76	0.924	0.917	1.8
Full BANGLATOKENIZER	1.73	0.936	0.931	1.0

Table 6: Empirical ablation analysis of BANGLATOKENIZER. The study evaluates noisy morpheme initialization, reduced lexicon coverage, cross-domain threshold sensitivity, human-validated annotations, robustness to noisy and code-switched text, removal of morphology-aware priors, and tokenization overhead. Metrics include Fertility (F), MorphScore (MS), and Dependency-Oriented Index (DOI). Δ Avg denotes the average rank across all metrics (lower is better).

Figure 3. Removing morphology-aware priors and relying only on raw bond strength substantially increases Fertility and reduces MorphScore (MS) and DOI, indicating severe over-segmentation and weaker morphemic cohesion. Reducing the 18K lemma lexicon and removing verified affix rules further degrades segmentation quality, showing strong dependence on accurate candidate morphemes produced by BanglaMorph. Robustness experiments on noisy and code-switched Bangla containing spelling variations, transliterated English words, and slang produce the largest performance drop, particularly in MS and DOI, due to unstable morpheme decomposition. To examine possible benchmark circularity, we construct a human-validated subset annotated by four native Bangla-speaking NLP graduate researchers (all male, aged 25-30, from the same research group), achieving strong Inter-Annotator Agreement (IAA) with Cohens $\kappa = 0.87$. Evaluation on this manually verified benchmark preserves the

Model	Task	Tokenizer	Acc. $\uparrow$	Macro-F1 $\uparrow$	PPL $\downarrow$	IS (tok/s) $\uparrow$
LLaMA-3.2-1B	Tense	Default	83.2	80.1	28.4	1150
		BT	86.5	83.9	24.7	1230
	Noun Num.	Default	82.6	79.4	29.1	1138
		BT	85.9	82.7	25.3	1215
BLOOMZ-1B7	Pronoun	Default	81.9	78.8	29.6	1124
		BT	85.1	82.0	25.8	1208
	Tense	Default	81.4	78.2	31.7	980
		BT	84.6	81.8	27.9	1045
mT5-Base	Noun Num.	Default	80.9	77.6	32.4	962
		BT	84.1	80.9	28.6	1028
	Pronoun	Default	80.3	77.1	32.9	951
		BT	83.6	80.2	29.1	1015
mT5-Base	Tense	Default	81.0	78.3	30.2	980
		BT	84.2	81.5	26.5	1055
	Noun Num.	Default	80.6	77.9	31.0	965
		BT	83.9	81.1	27.3	1038
Bangla-s1k-Qwen-2.5-3B	Pronoun	Default	80.1	77.4	31.5	952
		BT	83.4	80.6	27.8	1022
	Tense	Default	87.3	84.9	23.6	820
		BT	90.1	87.6	20.1	895
Bangla-s1k-Qwen-2.5-3B	Noun Num.	Default	86.8	84.2	24.4	808
		BT	89.6	87.1	20.9	882
	Pronoun	Default	86.1	83.6	24.9	796
		BT	88.9	86.4	21.3	870

Table 7: Supervised Fine-Tuning (SFT) evaluation across tense, noun number, and pronoun classification tasks using instruction-tuned, multilingual, and Bangla-specific LLMs. Replacing default tokenizers with BANGLATOKENIZER (BT) consistently improves downstream accuracy and macro-F1, reduces perplexity (PPL), and increases inference speed (IS), demonstrating that morphology-aware tokenization produces stronger downstream task performance beyond intrinsic efficiency metrics.

same overall trends, although automatic annotations slightly overestimate segmentation quality. We additionally analyze the threshold parameter $\tau \in [0.3, 1.0]$, where lower values induce over-segmentation and higher values suppress valid morpheme boundaries; the optimal value $\tau = 0.65$ achieves the best balance between granularity and morphological fidelity. At last, wall-clock analysis shows that although BANGLATOKENIZER incurs moderately higher preprocessing latency because of dictionary lookup and rule-based decomposition, reduced sequence lengths lower downstream transformer inference cost, yielding overall throughput improvements of 5%-10%.

6.1 Downstream Evaluation

Table 7 presents Supervised Fine-Tuning (SFT) results across tense, noun number, and pronoun classification tasks using multilingual, instruction-tuned, and Bangla-specific LLMs. Replacing default tokenizers with BANGLATOKENIZER consistently improves downstream performance across all models and tasks, yielding absolute accuracy gains of 2.8%-3.4% and macro-F1 improvements of 2.5%-3.6%. In addition, BANGLATOKENIZER reduces perplexity across all evalua-

tion settings while increasing inference throughput by approximately 5%-10%, indicating that morphology-aware segmentation produces shorter and more semantically coherent token sequences. The strongest gains are observed on Bangla-s1k-Qwen-2.5-3B, where accuracy reaches 90.1 on tense classification and macro-F1 improves to 87.6. These results demonstrate that preserving inflectional suffixes, compound structures, and morphemic cohesion leads to stronger downstream language understanding and more efficient inference than generic subword tokenization.

7 Conclusion

This work introduced BANGLATOKENIZER, a morphology-aware tokenizer for Bangla based on bond-strength modeling and linguistic priors. Across intrinsic, extrinsic, and downstream evaluations, BANGLATOKENIZER consistently improves morphological fidelity, task performance, and inference efficiency over existing tokenization methods.

Limitations

Although BANGLATOKENIZER improves morphological fidelity and downstream performance, sev-

eral limitations remain. First, the tokenizer depends on the BanglaMorph analyzer and an 18K-lemma lexicon to generate candidate morphemes, making segmentation quality sensitive to preprocessing errors and incomplete lexical coverage. Second, robustness decreases under noisy and code-switched text containing spelling variations, slang, or transliterated English words that are absent from the lexicon. Third, the threshold parameter τ shows domain sensitivity, requiring calibration across colloquial, technical, and formal Bangla corpora. Fourth, while a human-validated subset was constructed for evaluation, the benchmark remains partially dependent on automatically derived morphological rules, which may introduce residual annotation bias. At last, the rule-based preprocessing and dictionary lookup stages increase tokenization latency relative to purely statistical subword tokenizers, even though overall inference throughput improves because of shorter token sequences.

Ethics Statement

All experiments use publicly available datasets and pretrained language models. Human validation for the benchmark was conducted by native Bangla-speaking NLP graduate researchers from the same research group, and no monetary compensation was provided because all participants were internal collaborators within the project team. The proposed tokenizer is intended to improve linguistic representation and computational efficiency for Bangla NLP systems. However, like other language processing tools, it may inherit biases present in pretrained models, lexicons, or annotated corpora. In addition, dependence on rule-based morphological resources may disadvantage dialectal, informal, or code-switched forms of Bangla that are underrepresented in the lexicon. AI assistants were used for writing support and sentence polishing.

References

Ajwad Akil, Najrin Sultana, Abhik Bhattacharjee, and Rifat Shahriyar. 2022. Banglaparaphrase: A high-quality bangla paraphrase dataset. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 261–272.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, et al. 2024. Tokenizer choice for llm training: Negligible or crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3907–3924.

Catherine Arnett and Benjamin Bergen. 2025. Why do language models perform worse for morphologically complex languages? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6607–6623.

Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. 2025. Morphbpe: A morpho-aware tokenizer bridging linguistic complexity for efficient llm training across morphologies. *arXiv preprint arXiv:2502.00894*.

Christian Bentz, Annemarie Verkerk, Douwe Kiela, Felix Hill, and Paula Buttery. 2015. Adaptive communication: Languages with more non-native speakers tend to have fewer word forms. *PloS one*, 10(6):e0128254.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, and Rifat Shahriyar. 2023. Banglanlg and banglat5: Benchmarks and resources for evaluating low-resource natural language generation in bangla. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 726–735.

Md Hasan Imam Bijoy, Umme Ayman, and Md Monarul Islam. 2025. Banglatense: a large-scale dataset of bangla sentences categorized by tense: past, present, and future. *Data in Brief*, 59:111400.

Mathias Johan Philip Creutz and Krista Hannele Lagus. 2006. Morfessor in the morpho challenge. In *PASCAL Challenge Workshop on Unsupervised segmentation of words into morphemes*.

Michael Diskin, Alexey Bukhtiyarov, Max Ryabinin, Lucile Saulnier, Anton Sinitsin, Dmitry Popov, Dmitry V Pyrkin, Maxim Kashirin, Alexander Borzunov, Albert Villanova del Moral, et al. 2021. Distributed deep learning in open collaborations. *Advances in Neural Information Processing Systems*, 34:7879–7897.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75.

Chong Li, Jiajun Zhang, and Chengqing Zong. 2025. Tokalign: Efficient vocabulary adaptation via token alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4109–4126.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fernandez, Silvio Amir, Luis Marujo,

- and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1520–1530.
- Mohammad Junayed Khan Noor, Fatema Tuj Johora, Md Mahin, and Muhammad Aminur Rahaman. 2023. A real-time bangla local language recognition from voice. In *Proceedings of the Fourth International Conference on Trends in Computational and Cognitive Engineering: TCCE 2022*, pages 489–501. Springer.
- 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.
- Jonne Sälevä and Constantine Lignos. 2021. The effectiveness of morphology-aware segmentation in low-resource neural machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 164–174.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1715–1725.
- Roman S Stepanov, Aleksandra D Radina, Christian Tantardini, Alexander G Kvashnin, and Alexander V Kolobov. 2024. Chemical bonding within aiiibvi materials under uniaxial compression. *Physical Chemistry Chemical Physics*, 26(31):20984–20992.
- Hailay Kidu Teklehaymanot, Dren Fazlija, and Wolfgang Nejdl. 2025. Movoc: Morphology-aware subword construction for ge’ez script languages. In *EMNLP (Findings)*, pages 13131–13144.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- Sumit Yadav, Raju Kumar Yadav, Utsav Maskey, Gautam Siddharth Kashyap, Ganesh Gautam, and Usman Naseem. 2026. Maibert: A pre-training corpus and language model for low-resourced maithili language. In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM 2026)*, pages 444–452.
- Abdullah Khan Zehady, Shubhashis Roy Dipta, Naymul Islam, Safi Al Mamun, and Santu Karmaker. 2026. Banglallama: Llama for bangla language. In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM 2026)*, pages 73–89.