BanglaByT5: Byte-Level Modelling for Bangla

Pramit Bhattacharyya **Arnab Bhattacharya**
Dept. of Computer Science and Engineering,
Indian Institute of Technology Kanpur,
India
pramitb@cse.iitk.ac.in arnabb@cse.iitk.ac.in

Abstract

Large language models (LLMs) have achieved remarkable success across various natural language processing tasks. However, most LLM models use traditional tokenizers like BPE and SentencePiece, which fail to capture the finer nuances of a morphologically rich language like Bangla (Bengali). In this work, we introduce **BanglaByT5**, the first byte-level encoder-decoder model explicitly tailored for Bangla. Built upon a small variant of Google's ByT5 architecture, BanglaByT5 is pre-trained on a 14GB curated corpus combining high-quality literary and newspaper articles. Through zero-shot and supervised evaluations across generative and classification tasks, BanglaByT5 demonstrates competitive performance, surpassing several multilingual and larger models. Our findings highlight the potential of BanglaByT5 as a lightweight yet powerful tool for Bangla NLP, particularly in resource-constrained and scalable environments. BanglaByT5 is publicly available for download from https://huggingface.co/Vacaspati/BanglaByT5.

1 Introduction

Large Language Models (LLMs) have redefined natural language processing (NLP) by achieving strong results across multiple tasks like machine translation, question answering, and paraphrasing. However, these models rely on subword tokenization (e.g., BPE, SentencePiece), which fragments words inconsistently and poses significant challenges when applied to *morphologically rich* Indian languages like Bangla (Brahma et al., 2025; Nehrdich et al., 2024). In contrast, *byte-level modelling* operates directly on raw bytes, enabling models to handle linguistic variations uniformly across scripts and domains.

In this paper, we introduce **BanglaByT5**, the first monolingual byte-level encoder-decoder model for Bangla, built on the small ByT5 architecture and pre-trained on a 14GB balanced corpus combining VĀCASPATI (literature) (Bhattacharyya et al., 2023) and IndicCorp (news) corpora (Kakwani et al., 2020). BanglaByT5 is evaluated on classification and generative tasks under zero-shot and fine-tuned settings. It outperforms similar-size models like IndicBART and BanglaT5, as well as BLOOM-1.1B (Scao et al., 2023), and performs within 5% of GPT2-XL (Radford et al., 2019) despite being 5 times smaller.

Our contributions are as follows:

- We propose **BanglaByT5**, the first monolingual byte-level encoder-decoder model for Bangla.
- We conduct extensive evaluation across tasks and settings, showing competitive or superior performance to larger models.

2 Related Work

ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) have achieved strong results on NLU benchmarks. For generation tasks, decoder-only models like GPT-2/3 (Radford et al., 2019; Brown et al., 2020) and encoder-decoder models like T5 (Raffel et al., 2023), and mT5 (Xue et al., 2021)) have become prominent. ByT5 (Xue et al., 2022), a byte-level extension of mT5, has shown advantages for morphologically rich languages. While character-aware models like CharacterBERT (Boukkouri et al., 2020) and others (Ling et al., 2015; Chung et al., 2016; Jozefowicz et al., 2016; Wang et al., 2019; Wei et al., 2021; Kim et al., 2016) incorporate subword-free representations, they still rely on token boundaries. Other efforts (Garcia et al., 2021; Kudo, 2018) address tokenization challenges through vocabulary adaptation or randomized subword segmentation. Recent multilingual models like LLaMA-3 (Grattafiori et al., 2024), Mistral (Jiang et al., 2023), and IndicBART (Dabre et al., 2022) include Bangla. ByT5 model for mor-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 5551–5560
November 4-9, 2025

phologically rich language like Sanskrit (Nehrdich et al., 2024) has also been adopted. Monolingual models like BanglaT5 (Bhattacharjee et al., 2023) and Paramanu (Niyogi and Bhattacharya, 2024) are available for Bangla.

3 BanglaByT5

In this section, we discuss in detail the various aspects of our proposed model, BanglaByT5.

3.1 Pretraining Data

We curated a corpus by merging the VĀCASPATI (Bhattacharyya et al., 2023) and IndicCorp (Kakwani et al., 2020) (Bangla subset) corpora for pretraining the BanglaByT5 model. The merged corpus, 14 GB in size, contains 94,70,41,525 words and 7,51,51,084 sentences with 12.60 words per sentence. Since IndicCorp is a newspaper dataset with ∼3.8 million articles and VĀCASPATI is entirely curated from literary data, we can assure the quality of the merged corpora, which is essential for training any GenAI model (Luccioni and Viviano, 2021). We have not used data from other sources such as websites or blogs since it was not feasible for us to ascertain the quality of such data. We have used the preprocessing steps mentioned in (Bhattacharyya et al., 2023) (Appx A.1). The preprocessed corpus is used for the pretaining of BanglaByT5.

3.2 Pretraining Objective

ByT5, an encoder-decoder model, follows the same training objective as the original T5 model, specifically *span corruption denoising* task applied at byte level compared to token or subword level for T5. In ByT5, a fixed percentage of continuous byte spans are randomly selected and replaced with special sentinel tokens. The model is then trained to reconstruct the original spans, treating this as a sequence-to-sequence generation task.

3.3 Model Architecture and Hyperparameters

Before the pretraining of the model on the merged corpus, we trained a byte-level tokenizer with 384 vocabulary size that includes 100 special tokens. This tokenizer generated 7,53,32,70,552 tokens for the merged corpus, resulting in a fertility score of 7.96, which is neither too high as Google-byt5-small (15.02) nor too small as BanglaT5 (1.20). This tokenizer is used for pretraining BanglaByT5.

We pre-trained the small variant of the Google-ByT5 model (Xue et al., 2022) with 12 hidden layers, 6 attention heads, 1472 hidden size, 3584 feed-forward size with gated-GELU activation (Shazeer, 2020). The model was trained with a batch size of 16 and a gradient accumulation step of 2 for over 3e6 steps, utilizing two A100 40GB GPU instances. We employed the Adam optimizer (Kingma and Ba, 2017) with a learning rate of 3e-5, a linear warm-up for the first 500 steps, and a cosine learning rate scheduler. We train the model with a context size of 512 (∼5 sentences). The resulting model has ∼300M parameters yielding a token-to-parameter ratio of ∼25.14.

4 Evaluation

In this section, we explore the efficacy of BanglaByT5. We adopted a two-fold approach. First, we asked BanglaByT5 to generate responses to curated Bangla prompts to evaluate its generative abilities in the zero-shot setting. Zero-shot evaluation is particularly important because it reveals the model's inherent generative ability without reliance on domain-specific adaptation. Then, we evaluated the performance of BanglaByT5 on both classification and generation-based downstream tasks in supervised fine-tuning mode.

4.1 Prompt Generation in Zero-Shot Setting

We adopted a two-stage evaluation methodology to assess the responses generated by BanglaByT5 and competing models. In the first stage, we evaluated the model's responses across four key dimensions using LLaMa-3.1-8B (Grattafiori et al., 2024) and Mistral-7B (Jiang et al., 2023) as LLM-as-a-Judge (Gu et al., 2025). LLM-as-a-judge is consistent with human evaluators for Bangla (**?**), hence we have used this for our experiments. The four key metrics used are *Fluency, Coherence, Relevance* and *Creativity* (definitions in Appx A.3). Prompt used for LLM-as-a-judge is shown in Figure 1 of Appx A.2. Each prompt is run 5 times to capture the variation in generation by the LLMs and is graded on a scale of 1 to 10. Table 1 shows the performance of BanglaByT5 against other models. From Table 1, it is evident that the generation ability of BanglaByT5 is comparable to GPT2-XL (the best performing model) and is better than any other model even if it is twice (GPT2-Large) or thrice (BLOOM-1.1B (Scao et al., 2023)) in size.

We further evaluated the generation ability of BanglaByT5 model by comparing its responses for the 2000 prompts against the responses by two

Model	Parameters	Mistral-7B				LLaMA-3.1-8B			
		Fluency	Relevance	Coherence	Creativity	Fluency	Relevance	Coherence	Creativity
mt5-small	240M	1.60 ± 1.3	2.60 ± 1.6	2.50 ± 1.4	1.00 ± 1.4	2.00 ± 1.4	2.60 ± 1.5	2.50 ± 1.1	2.60 ± 1.8
mt5-base	580M	6.00 ± 1.3	6.00 ± 1.2	6.60 ± 1.1	4.00 ± 1.6	6.00 ± 1.6	6.60 ± 1.5	6.00 ± 1.1	4.50 ± 1.6
mt5-large	1.1B	8.60 ± 1.1	8.60 ± 1.3	8.60 ± 1.2	6.00 ± 1.4	8.60 ± 1.4	8.60 ± 1.5	8.60 ± 1.1	6.00 ± 1.6
google-byt5-small	300M	6.60 ± 0.2	6.60 ± 0.4	6.30 ± 0.3	4.00 ± 0.6	6.60 ± 0.3	6.60 ± 0.2	6.00 ± 0.3	4.00 ± 0.7
google-byt5-base	580M	7.60 ± 0.1	7.60 ± 0.6	7.60 ± 0.7	5.00 ± 0.6	7.60 ± 0.3	7.60 ± 0.5	7.00 ± 0.3	5.00 ± 0.7
google-byt5-large	1.2 B	9.00 ± 0.2	9.00 ± 0.4	9.00 ± 0.3	6.00 ± 0.6	9.00 ± 0.3	9.00 ± 0.2	9.00 ± 0.3	6.00 ± 0.7
GPT-2 Medium	355M	8.00 ± 0.1	7.00 ± 0.5	6.00 ± 0.3	6.00 ± 0.4	8.00 ± 0.2	7.00 ± 0.4	6.00 ± 0.2	6.00 ± 0.6
GPT-2 Large	774M	9.00 ± 0.8	9.00 ± 0.4	8.00 ± 0.6	6.50 ± 0.5	9.00 ± 0.1	8.60 ± 0.8	8.00 ± 0.2	6.00 ± 0.4
GPT-2 XL	1.5B	$\mathbf{9.00 \pm 0.2}$	$\mathbf{9.00 \pm 0.5}$	$\mathbf{9.00 \pm 0.5}$	$\mathbf{6.50 \pm 0.6}$	$\mathbf{9.00 \pm 0.2}$	$\mathbf{9.00 \pm 0.5}$	$\mathbf{8.67 \pm 0.2}$	$\mathbf{7.00 \pm 0.6}$
BLOOM	560M	7.00 ± 0.3	6.50 ± 0.2	6.00 ± 0.3	4.00 ± 0.5	6.50 ± 0.2	6.50 ± 0.3	7.00 ± 0.4	4.00 ± 0.5
BLOOM	1.1B	8.00 ± 0.3	7.70 ± 0.2	7.70 ± 0.3	5.00 ± 0.5	7.50 ± 0.2	8.00 ± 0.3	7.70 ± 0.4	5.50 ± 0.5
IndicBART	272M	6.30 ± 0.4	7.00 ± 0.2	6.00 ± 0.3	3.00 ± 0.5	6.50 ± 0.1	7.30 ± 0.3	7.00 ± 0.4	3.50 ± 0.5
BanglaT5	240M	1.60 ± 1.1	3.00 ± 1.3	2.50 ± 1.2	1.00 ± 1.4	2.00 ± 1.4	3.60 ± 1.5	2.50 ± 1.1	2.80 ± 1.6
Paramanu	334M	6.30 ± 0.4	7.00 ± 0.2	6.00 ± 0.3	3.00 ± 0.5	6.50 ± 0.1	7.30 ± 0.3	7.00 ± 0.4	3.50 ± 0.5
BanglaByT5	300M	8.60 ± 0.2	8.60 ± 0.4	8.30 ± 0.3	5.00 ± 0.6	8.60 ± 0.3	8.60 ± 0.2	8.00 ± 0.3	5.00 ± 0.7

Table 1: LLM evaluation of Bangla generation using Mistral-7B and LLaMA-3.1-8B as LLM-as-a-Judge

Model	Parameters	LLaMA-3.1-8B			Mistral-7B		
		BERTScore	BLEU	METEOR	BERTScore	BLEU	METEOR
MT5-SMALL	300M	49.31 ± 1.40	1.50 ± 1.20	1.40 ± 1.30	48.89 ± 1.56	1.20 ± 1.10	1.32 ± 1.20
MT5-BASE	580M	49.56 ± 1.60	1.56 ± 1.40	1.45 ± 1.30	49.99 ± 1.78	1.65 ± 1.42	1.74 ± 1.30
MT5-LARGE	1.2B	67.03 ± 1.70	19.29 ± 1.90	37.37 ± 1.60	61.66 ± 1.58	9.90 ± 1.86	23.22 ± 1.54
BYT5-SMALL	300M	71.10 ± 1.50	1.71 ± 1.30	13.17 ± 1.65	70.32 ± 1.42	1.19 ± 1.23	12.71 ± 1.56
BYT5-BASE	580M	71.32 ± 1.50	3.54 ± 1.60	14.27 ± 1.55	70.56 ± 1.45	2.57 ± 1.35	16.15 ± 1.58
BYT5-LARGE	1.2B	75.80 ± 1.60	8.66 ± 1.80	21.20 ± 1.63	74.23 ± 1.55	7.54 ± 1.70	19.15 ± 1.56
BLOOM-560M	560M	72.49 ± 1.40	6.96 ± 1.50	32.84 ± 1.68	72.43 ± 1.45	8.52 ± 1.56	30.61 ± 1.52
BLOOM-1B	1.1B	72.63 ± 1.50	7.23 ± 1.60	33.45 ± 1.52	74.68 ± 1.65	9.46 ± 1.64	31.50 ± 1.54
GPT-2 MEDIUM	355M	76.08 ± 1.60	11.76 ± 1.70	33.56 ± 1.64	75.00 ± 1.54	8.26 ± 1.52	30.45 ± 1.44
GPT2-LARGE	774M	80.51 ± 1.60	31.26 ± 1.70	50.92 ± 1.50	80.40 ± 1.64	30.66 ± 1.61	49.25 ± 1.54
GPT2-XL	1.5B	$\mathbf{81.69 \pm 1.60}$	$\mathbf{32.40 \pm 1.80}$	$\mathbf{51.56 \pm 1.50}$	$\mathbf{81.29 \pm 1.56}$	$\mathbf{31.86 \pm 1.60}$	$\mathbf{50.46 \pm 1.56}$
IndicBART	272M	61.81 ± 1.60	1.86 ± 1.40	5.40 ± 1.57	62.58 ± 1.44	1.86 ± 1.35	4.75 ± 1.46
BanglaT5	240M	63.81 ± 1.60	2.86 ± 1.45	7.49 ± 1.57	64.08 ± 1.42	2.86 ± 1.30	6.75 ± 1.40
Paramanu	334M	62.31 ± 1.40	2.00 ± 1.72	6.00 ± 1.56	62.88 ± 1.45	2.26 ± 1.38	6.25 ± 1.45
BanglaByT5	300M	78.21 ± 1.10	12.41 ± 1.96	34.08 ± 1.61	75.84 ± 1.03	9.06 ± 1.38	31.63 ± 1.54

Table 2: Benchmarking the generation ability of BanglaByT5 models in zero-shot setting

widely used reference models LLaMA-3.1-8B and Mistral-7B. Figure 3 of Appx A.2 shows example of two such prompts. We further benchmarked the response generated by BanglaByT5 by comparing it with the competing models. Each model was executed 5 times, and the mean and standard deviations are shown in Table 2 of Appx A.2.

Table 1 and Table 2 indicate that the generation-ability of BanglaByT5 is better than models with twice to thrice its parameter size and is comparable to GPT2-XL, which is five times larger. This generation ability of BanglaByT5 makes it a suitable model for deployment in low-resource settings. In Section 5, we have discussed the deployment potential of BanglaByT5 in detail.

4.2 Supervised Fine Tuning

We further investigated the performance BanglaByT5 on various downstream tasks and benchmarked it against similar and larger parameter models on both classification and generation tasks.

Sentiment Classification: We used the dataset curated by (Islam et al., 2018), which comprises 3 polarity labels, positive, negative, and neutral, and is collected from social media comments on news and videos covering 13 domains, including politics, education, and agriculture. It consists of 5,709 negative, 6,410 positive, and 3,609 neutral sentences. For this classification task, we used *macro-F1* as an evaluation metric.

NER: We chose the publicly available Naamapadam (Mhaske et al., 2023) (Bengali subset) for this classification task. The dataset consists of 961.7K sentences for training, and 4.9K sentences have been used for evaluation. The tokens are tagged into 4 classes: Person (Per), Location (Loc), Organization (Org), and other (O). We have used *macro-F1* as an evaluation metric for the task.

Machine Translation: Machine Translation (MT) is one of the most studied generative tasks in Bangla. For this task, we curated a dataset by merging the dataset created by Gala et al. (2023) (1,022

Model	Parameters	Sentiment	NER	MT (sacreBLEU)	Paraphrasing	GEC (GLEU)
mt5-small	240M	62.50 ± 1.35	28.10 ± 1.42	20.10 ± 1.43	32.80 ± 1.56	63.00 ± 1.47
mT5-Base	580M	67.50 ± 1.35	33.10 ± 1.45	23.10 ± 1.43	35.50 ± 1.57	65.00 ± 1.87
ByT5-small	300M	64.60 ± 0.20	30.60 ± 0.55	21.86 ± 0.50	34.28 ± 1.60	63.00 ± 1.45
ByT5-base	580M	67.60 ± 0.20	32.80 ± 0.55	23.86 ± 0.50	35.48 ± 1.60	66.10 ± 1.45
GPT-2 Medium	355M	63.00 ± 1.50	29.00 ± 1.20	22.00 ± 1.50	34.00 ± 1.70	63.00 ± 1.40
GPT-2 Large	774M	67.80 ± 1.44	**35.00 ± 1.35**	24.20 ± 1.62	**36.50 ± 1.60**	66.20 ± 1.50
BLOOM	560M	64.90 ± 1.47	31.50 ± 1.39	22.80 ± 1.63	34.60 ± 1.62	63.20 ± 1.55
IndicBART	272M	63.40 ± 1.45	30.80 ± 1.36	22.40 ± 1.60	34.40 ± 1.56	62.50 ± 1.50
BanglaT5	240M	67.80 ± 1.40	33.00 ± 1.30	22.50 ± 1.50	34.80 ± 1.60	64.50 ± 1.50
Paramanu	334M	66.00 ± 1.40	32.20 ± 1.30	21.90 ± 1.40	33.50 ± 1.50	63.70 ± 1.50
BanglaByT5	300M	**68.30 ± 0.20**	33.60 ± 0.35	**24.36 ± 1.50**	35.78 ± 1.60	**66.27 ± 1.40**

Table 3: Comparison of different models on various downstream tasks

sentences) and Costa-jussà et al. (2022) (3,001 sentences) and creating a comprehensive dataset of 4,023 sentences. We used 80% of the data for training the model and tested on the remaining 20%. We used the *sacreBLEU* score as the evaluation metric for the task.

Paraphrasing: Paraphrasing refers to the rephrasing of a sentence or passage using different words and structures while preserving its original meaning. We used the publicly available paraphrasing dataset curated by Akil et al. (2022) for this task. The dataset consists of 5,763 sentences, of which 80% is used for training and the remaining 20% for testing. We used the *sacreBLEU* score as the evaluation metric for the task.

Grammatical Error Correction: Grammatical Error Detection (GEC) refers to the task of automatic detection and correction of grammatical errors in a sentence. It is one of the most important generative tasks as it also tests the model's understanding of generating semantically correct sentences. We used the VAIYAKARANA dataset curated by Bhattacharyya and Bhattacharya (2024). The dataset consists of 1,11,256 sentences divided into 12 finer classes. Similar to the other generative tasks, we have used 80% of the dataset for training and 20% for testing. We used *GLEU* as the evaluation metric for the task.

We compared the performance of BanglaByT5 on the specified downstream tasks against similar parameter models such as mT5-small, mT5-base, Google-ByT5-small, Google-ByT5-base, BanglaT5, IndicBart, Paramanu, GPT2-medium and GPT2-large. All the pre-trained models are run for 5-25 epochs on a single instance of Nvidia A100-46 GB GPU. We have used beam search for inferencing (using 10 beams) and set the temperature at 0.7 and the top_k value at 70. The maximum output length has been set at 512 for all the models. Table 3 shows the result of all the models on the downstream tasks based on the evaluation metrics discussed in this section.

Table 3 shows that BanglaByT5 outperforms all similar parameter models on generative tasks such as MT, paraphrasing and GEC while giving comparable results on classification tasks like Sentiment classification and NER. BanglaByT5 also performs similarly to the GPT2-Large model on all the generative tasks, outperforming it on classification tasks. The results indicate the efficacy of BanglaByT5 as a generative model for Bangla. Additionally, we have benchmarked the performance of BanglaByT5 against larger models like Google-byt5-large, mT5-large, GPT2-XL and BLOOM-1.1B, results of which are shown in Table A1 of Appx A.4. BanglaByT5 outperforms BLOOM-1.1B in all tasks and performs within 5% of the other larger models.

5 Deployability

In this section, we evaluate the scalability of the BanglaByT5 model under CPU-only and GPU-accelerated environments to assess its deployment potential. *GPU-scalability* reflects how well the model leverages parallelism for high-throughput or real-time deployment, while *CPU-scalability* captures performance in low-resource environments. This dual perspective is essential for understanding the potential of deployment in cloud and offline settings. *Latency* denotes the average time (in seconds) required to generate an output of a single prompt, including tokenization, model forward pass and decoding. *Throughput*, on the other hand, focusses on number of prompts processed per second. We also monitor the peak memory usage for a prompt in both CPU and GPU mode, as this is a critical consideration for deployment.

To evaluate the deployment potential, we curated

Batch	Latency (sec)	Throughput	Memory(MB)
1	0.5646	1.77	2216.37
2	0.2692	3.71	2330.37
4	0.1550	6.45	2418.31
8	0.0949	10.54	2582.26
16	0.0855	11.7	2601.56
32	0.0828	12.07	2742.99
64	0.0806	12.41	2743.99

Table 4: CPU-only scalability results for BanglaByT5 across increasing batch sizes

Batch	Latency (sec)	Throughput	GPU-Mem (MB)
1	1.0927	0.92	1166.09
2	0.1592	6.28	1177.29
4	0.1296	7.72	1192.34
8	0.0810	12.34	1238.53
16	0.0439	22.80	1322.05
32	0.0409	24.48	1487.59
64	0.0146	68.26	1812.68

Table 5: GPU scalability results for BanglaByT5 across increasing batch sizes

a dataset of 200 prompts for the paragraph generation task with varying word lengths (5-25), with an average of 9.81 words per prompt, similar to the average word length found in VĀCASPATI (Bhattacharyya et al., 2023).

Table 4 shows the variation in latency, throughput and memory required in CPU-only mode with an increase in batch size. From the table, it is seen that latency decreases and throughput increases with an increase in batch size, which is the ideal scenario. The peak memory usage is ∼2,744 MB (2.68 GB). Hence, the model can be deployed in an offline system with as low as 4 GB of RAM.

Table 5 shows the variation in latency and throughput along with CPU and GPU requirements in gpu-available mode on the same 200 prompts. The maximum GPU requirement is ∼1.77 GB. Further analysis shows that maximum cpu-requirement is ∼588 MB when batch size is 1.

Table 4 and Table 5 demonstrate that GPU acceleration yields substantial gains in throughput and reduces latency per sentence, but GPU memory usage increases sharply with batch size. In contrast, CPU-based inference falls behind in throughput but remains viable for offline deployments, especially in systems with limited memory.

Byte-level modelling produces more tokens than subword tokenization, thereby increasing training and inference time. Large training time is a bottleneck for deployment. To evaluate the performance of BanglaByT5 with subword-tokenization (SentencePiece), we have conducted a stress test with the same set of 200 sentences used for scalability experiments on both BanglaT5 and BanglaByT5. Table 6 shows that the average latency and throughput of BanglaByT5 are better than that of BanglaT5, whereas BanglaT5 requires less memory than BanglaByT5. However, all modern-day devices generally have 512 MB of RAM, thus facilitating the deployment of BanglaByT5. On comparing the SFT time of both BanglaByT5 and BanglaT5 on the sentiment analysis task, we found that BanglaByT5 requires around 120 minutes for 20 epochs, while BanglaT5 takes around 100 minutes for the same. Table 7 shows the performance of BanglaByT5 and BanglaT5 on two downstream tasks after 3 epochs. It is evident that even without extensive finetuning due to resource constraints, BanglaByT5 still outperforms the competing models.

Metric	BanglaT5	BanglaByT5
Params	240M	300M
Avg Latency	27 ms	**24 ms**
Throughput	37.00	**41.70**
Peak RAM	**250 MB**	490 MB
RAM Overhead	**30 MB**	60 MB

Table 6: Comparison of BanglaByT5 with BanglaT5 after stress testing with 200 sentences

	Sentiment		GEC	
Model	Macro-F1	Training Time	GLEU	Training Time
BanglaT5	25.67	16 mins	41.30	580 mins
BanglaByT5	**54.30**	18 mins	**61.35**	660 mins

Table 7: Comparison of BanglaByT5 with BanglaT5 on downstream tasks

6 Conclusions

We presented **BanglaByT5**, a byte-level monolingual language model explicitly tailored for morphologically rich languages like Bangla. Through rigorous evaluation, we demonstrate that BanglaByT5 surpasses existing Bangla models and matches or outperforms larger multilingual models in both generation and classification tasks. Moreover, its scalability in low-resource environments positions it as a practical and impactful tool for Bangla NLP. BanglaByT5 is available for download from `https://huggingface.co/Vacaspati/BanglaByT5`.

7 Limitations

Lack of large quantity of quality data: Bangla inherently suffers from large quantity of quality data. We have been able to curate only 14GB of data, prompting us to use a small variant of google-ByT5. Our results indicate that a larger variant pre-trained over a large quality corpus will benefit Bangla.

Hallucination: Hallucination is an inherent property of any LLMs (Xu et al., 2025). Hence, we cannot always guarantee the factual correctness of responses generated by BanglaByT5.

Memorization: Similar to hallucination, memorization is also an inherent property of LLMs (Hartmann et al., 2023). However, Carlini et al. (2019) showed that models with $<=$300M parameters show minimal memorization. Appx A.5 discusses the memorization ability of BanglaByT5 in detail.

8 Ethics Statement

The corpus used for pre-training BanglaByT5 is curated by merging IndicCorp (Kakwani et al., 2020) and VĀCASPATI (Bhattacharyya et al., 2023). The authors of VĀCASPATI have provided us with the corpus, and IndicCorp is publicly available. Hence, there is no copyright infringement in the curation of the merged corpus. Since IndicCorp is a newspaper corpus and VĀCASPATI is a literary corpus, there are minimal chances of having objectionable and offensive statements. For Grammar Error Correction (GEC) work, the authors of VAIYAKARANA also provided us with the dataset. Hence, there is no copyright infringement.

Carbon Footprint: We estimate the carbon emissions incurred during the pretraining of BanglaByT5 on 2 NVIDIA A100 (40GB) GPUs, each with a Thermal Design Power (TDP) of 250W, for 600 training hours. This results in an energy consumption of approximately $0.25kW \times 2 \times 600hours = 300kWh$. Assuming an average carbon intensity of 0.7 $kgCO_2/kWh$, the total carbon emission is estimated as:

$$300\,\mathrm{kWh} \times 0.7\,\frac{\mathrm{kg\,CO_2}}{\mathrm{kWh}} = \mathbf{210}\,\mathrm{kg\,CO_2}$$

which is significantly lower than the emissions reported for large-scale models such as GPT2-XL (Strubell et al., 2019).

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, Online only. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin 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, Dubrovnik, Croatia. Association for Computational Linguistics.

Pramit Bhattacharyya and Arnab Bhattacharya. 2024. Vaiyakarana : A benchmark for automatic grammar correction in bangla. *Preprint*, arXiv:2406.14284.

Pramit Bhattacharyya, Joydeep Mondal, Subhadip Maji, and Arnab Bhattacharya. 2023. VACASPATI: A diverse corpus of Bangla literature. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1130, Nusa Dua, Bali. Association for Computational Linguistics.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Junichi Tsujii. 2020. Characterbert: Reconciling elmo and bert for word-level open-vocabulary representations from characters. *Preprint*, arXiv:2010.10392.

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.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. *Preprint*, arXiv:1802.08232.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany. Association for Computational Linguistics.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, and 19 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh Khapra, and Pratyush Kumar. 2022. IndicBART: A pre-trained model for indic natural language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1849–1863, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1184–1192, Online. Association for Computational Linguistics.

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.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Valentin Hartmann, Anshuman Suri, Vincent Bindschaedler, David Evans, Shruti Tople, and Robert West. 2023. Sok: Memorization in general-purpose large language models. *Preprint*, arXiv:2310.18362.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Muhammad Ifte Khairul Islam, Md. Tarek Habib, Md. Sadekur Rahman, Md. Riazur Rahman, and Farruk Ahmed. 2018. A Context-Sensitive Approach to Find Optimum Language Model for Automatic Bangla Spelling Correction. *International Journal of Advanced Computer Science and Applications*, 9(11).

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.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *Preprint*, arXiv:1602.02410.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2741–2749. AAAI Press.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

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, Melbourne, Australia. Association for Computational Linguistics.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based neural machine translation. *Preprint*, arXiv:1511.04586.

Alexandra Luccioni and Joseph Viviano. 2021. What‘s in the box? an analysis of undesirable content in the Common Crawl corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational*

Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 182–189, Online. Association for Computational Linguistics.

Arnav Mhaske, Harshit Kedia, Sumanth Doddapaneni, Mitesh M. Khapra, Pratyush Kumar, Rudra Murthy, and Anoop Kunchukuttan. 2023. Naamapadam: A large-scale named entity annotated data for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10441–10456, Toronto, Canada. Association for Computational Linguistics.

Sebastian Nehrdich, Oliver Hellwig, and Kurt Keutzer. 2024. One model is all you need: Byt5-sanskrit, a unified model for sanskrit nlp tasks. *Preprint*, arXiv:2409.13920.

Mitodru Niyogi and Arnab Bhattacharya. 2024. Paramanu: A family of novel efficient generative foundation language models for indian languages. *Preprint*, arXiv:2401.18034.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, and Pawan Sasanka Ammanamanchi et al. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2019. Neural machine translation with byte-level subwords. *ArXiv*, abs/1909.03341.

Junqiu Wei, Qun Liu, Yinpeng Guo, and Xin Jiang. 2021. Training multilingual pre-trained language model with byte-level subwords. *Preprint*, arXiv:2101.09469.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *Preprint*, arXiv:2312.12148.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2025. Hallucination is inevitable: An innate limitation of large language models. *Preprint*, arXiv:2401.11817.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Paul Barham, Sharan Narang, Jacob Devlin, and Rami Sepassi. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5166–5180.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

A Appendix

A.1 Data Cleaning

- *Cleaning of Unicode characters*: Unicode characters "0020" (space), "00a0" (no-break space), "200c" (zero width non-joiner), "1680" (ogham space mark), "180e" (mongolian vowel separator), "202f" (narrow no-break space), "205f" (medium mathematical space), "3000" (ideographic space), "2000" (en quad), "200a" (hair space) are removed from the texts.
- *Cleaning of different punctuation marks*: Usage of punctuation marks have also evolved alongside words in Bangla.In total we have removed all 36 types of Bangla punctuation marks.

A.2 Prompt Generation

Figure 1 shows the prompt use for LLM-as-a-judge evaluation for BanglaByT5. Figure 2 shows the number of distribution of words in curated prompts whereas Figure 3 shows few example prompts used in Section 4.1 for assessing the zero-shot generation ability of BanglaByT5. Table A1 shows the performance of BanglaByT5 with larger models such as GPT2-XL.

Model	Parameters	Sentiment	NER	MT (sacreBLEU)	Paraphrasing	GEC (GLEU)
mT5-Large	1.2B	69.60 ± 1.30	35.00 ± 1.40	25.30 ± 1.65	37.73 ± 1.50	69.32 ± 1.40
Google-ByT5-Large	1.2B	70.90 ± 1.40	36.90 ± 1.50	26.35 ± 1.62	38.54 ± 1.50	69.88 ± 1.40
GPT2-XL	1.5B	**72.47 ± 1.40**	**37.30 ± 1.56**	**28.58 ± 1.72**	**38.84 ± 1.56**	**70.83 ± 1.50**
BLOOM-1.1B	1.1B	66.00 ± 1.45	31.50 ± 1.35	23.50 ± 1.55	35.60 ± 1.45	65.20 ± 1.53
BanglaByT5	300M	68.30 ± 0.20	33.60 ± 0.35	24.36 ± 1.50	35.78 ± 1.60	66.27 ± 1.40

Table A1: Performance comparison of BanglaByT5 against larger models on 5 Bangla NLP tasks

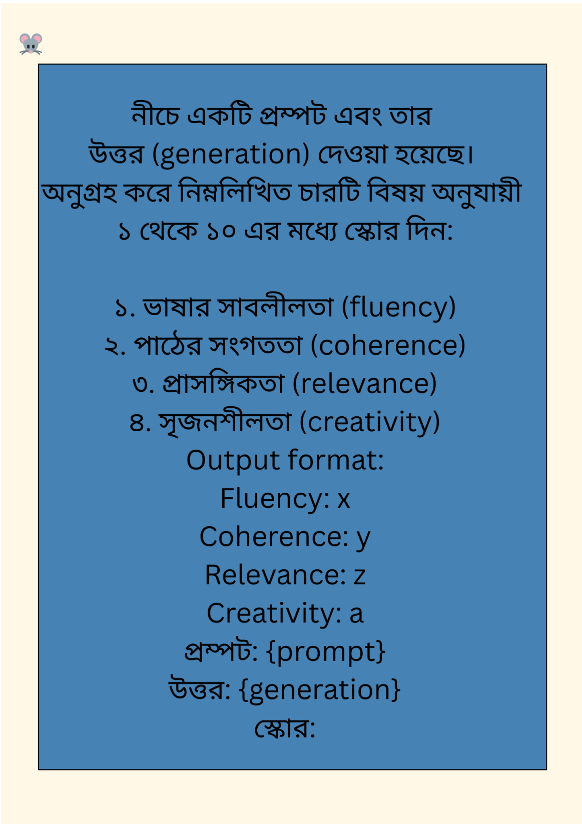

Figure 1: Prompts used for LLM-as-a-judge evaluation

A.3 Evaluation Metrics for Zero-shot Evaluation

BanglaByT5 generation ability have been evaluated using four metrics keeping LLaMa-3.1 (8B) and Mistral-7B.

- **Fluency** — It refers to the grammatical correctness and naturalness of the generated language.
- **Coherence** — Coherence signifies the consistency and structure of multi-turn responses.
- **Relevance** — Relevance refers to the contextual alignment with the original prompt.
- **Creativity** — Creativity is defined as the novelty and expressiveness of the generated response.

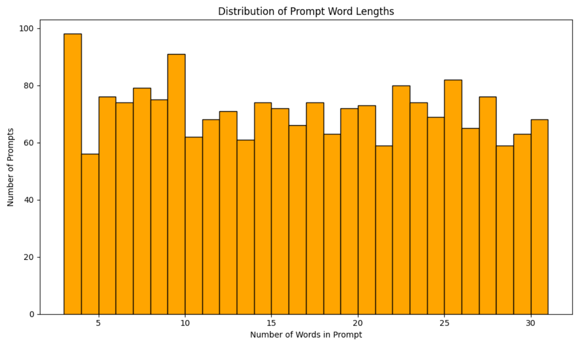

Figure 2: Distribution of prompt length by word

A.4 Benchmarking against Larger Models

In this section we benchmarked BanglaByT5 against larger models like google-ByT5-large, mT5-large, GPT2-XL and BLOOM-1.1B on the downstream tasks specified in Sec 4.2. BanglaByT5 outperforms BLOOM-1.1B on all tasks and perform with 2-5% of the other models in spite of being 4-5 times smaller. Table A1 shows the result of BanglaByT5 and the larger models.

A.5 Memorization

Memorization is an inherent ability of LLMs. In this section, we evaluated the memorization ability of BanglaByT5. We curated a canary dataset mostly with names, locations, numbers and email IDs, which are more susceptible to memorization. We evaluate canary memorization using Exact Match (EM), i.e., the percentage of generated sentences that exactly match canary sentences. We finetuned the model for 3 epochs with LoRA on the canary dataset and tested it on 250 test sentences. We get an EM score of 0.00, which means the model is not reproducing the canaries verbatim, suggesting no direct memorization. We also evaluated perplexity, the exponential of average loss over the canary set, indicating how confidently the model reproduces them. The perplexity of the model after instruction tuning with the canary dataset is 47.55 (high), indicating that BanglaByT5

Figure 3: Examples of prompts used for experiments on generation ability of BanglaByT5

is not memorizing.

A.6 Model HyperParameters

All models were instruction-tuned using the Low-Rank Adaptation (LoRA) method (Hu et al., 2021), a parameter-efficient fine-tuning approach for pre-trained models (Xu et al., 2023). The LoRA hyper-parameters were set as follows:

- Rank (r): 16
- LoRA alpha (α): 32
- LoRA dropout: 0.05
- Bias: none

For all the models, the following hyperparameter values have been used for generation:

- temperature = 0.7
- top_k = 50
- num_beams = 10
- max_length = 1800

The default values have been used for all other hyperparameters.