BanglaTLit: A Benchmark Dataset for Back-Transliteration of Romanized Bangla

Md Fahim[1,2]*, Fariha Tanjim Shifat[1]*, Fabiha Haider[1]*, Deeparghya Dutta Barua[1], Md Sakib Ul Rahman Sourove[1], Md Farhan Ishmam[1,3], Md Farhad Alam[1]
[1]Research and Development, Penta Global Limited, Bangladesh
[2]CCDS Lab, Independent University, Bangladesh
[3]Islamic University of Technology, Bangladesh
pdcsedu@gmail.com, farhanishmam@iut-dhaka.edu

Abstract

Low-resource languages like Bangla are severely limited by the lack of datasets. Romanized Bangla texts are ubiquitous on the internet, offering a rich source of data for Bangla NLP tasks and extending the available data sources. However, due to the informal nature of romanized text, they often lack the structure and consistency needed to provide insights. We address these challenges by proposing: (1) BanglaTLit, the large-scale Bangla transliteration dataset consisting of 42.7k samples, (2) BanglaTLit-PT, a pre-training corpus on romanized Bangla with 245.7k samples, (3) encoders further-pretrained on BanglaTLit-PT achieving state-of-the-art performance in several romanized Bangla classification tasks, and (4) multiple back-transliteration baseline methods, including a novel encoder-decoder architecture using further pre-trained encoders. Our results show the potential of automated Bangla back-transliteration in utilizing the untapped sources of romanized Bangla to enrich this language. The code and datasets are publicly available: https://github.com/farhanishmam/BanglaTLit.

1 Introduction

In recent years, we have witnessed remarkable progress in various Natural Language Processing (NLP) tasks driven by Large Language Models (LLMs). However, these advancements have not been equally shared across all languages (Joshi et al., 2020), particularly low-resource languages like Bangla, despite its 250 million native speakers globally. A prevalent form of Bangla text is romanized Bangla, which uses phonetically similar Latin scripts to represent Bangla syllables. The widespread use of romanized Bangla on social media and online platforms, largely due to the familiarity with English keyboard layouts such as QWERTY, presents a valuable data source for low-resource languages (Moosa et al., 2022). Despite its ubiquity, significant challenges remain in processing romanized Bangla, primarily due to the lack of standardized datasets.

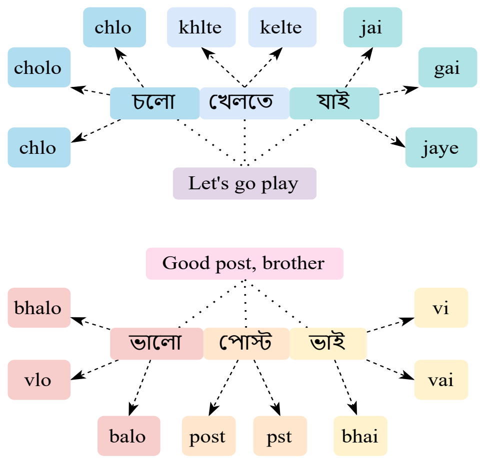

Figure 1: Variations in romanizing Bangla words within a sentence. The flexibility allows the same Bangla word to have multiple romanized forms.

Unlike other languages with complex phonetic mapping, Bangla has a phonemic orthography, meaning it is written as it sounds. This characteristic simplifies romanization and adds flexibility in how Bangla words can be romanized, as illustrated in Figure 1. However, the real complexity lies in the back-transliteration process, i.e., converting romanized texts back to the native Bangla script, as this process must adhere to the grammatical rules of Bangla. Automatic back-transliteration can extend the training data of low-resource languages like Bangla, as romanized texts are informal in nature and do not provide significant insights (Roark et al., 2020). Another potential use case for automated back-transliteration is its deployment as a

*Equal Contribution

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 14656–14672
November 12-16, 2024 ©2024 Association for Computational Linguistics

Dataset	Data Source	LT	PT	Curation	CT	Ver.	#Samples
Shibli et al. (2023)	Fb, YT, Blog	B	✗	HA	B	HE	5k
Roark et al. (2020)	Wiki	M	✗	HA	R,B	HA	10k
Madhani et al. (2023a)	Wiki	M	✗	HA	B	HA	4.6k
Kabiraj et al. (2023)	WhatsApp	B	✗	HA	B	N/A	_
Ours	TBD, Fb, YT, Blog, Wiki	B	✓	HA	B	HE	42.7k

Table 1: Comparison of several Bangla datasets and multilingual transliteration datasets with Bangla samples based on the data source [Fb: Facebook, YT: YouTube, Wiki: Wikipedia, TBD: TrickBD], linguistic type (LT) [B: Bangla, M: Multilingual], availability of pre-training corpus (PT), data curation method [HA: Human Annotated], data curation type (CT) [R: Romanized, B: Back-transliterated], data verification method [HE: Human Expert, HA: Human Annotator], and number of Bangla samples in the dataset. [_] indicates that the number of data samples has not been specified in the paper.

transliteration layer on top of any language model, enabling better interaction with romanized texts and extending the functionality of the native scripts to their romanized counterparts.

Current Bangla transliteration datasets suffer from insufficient data samples, limited data sources, and are mostly subsets of larger multilingual datasets, as evident from Table 1. While current pre-trained sequence-to-sequence models perform well in tasks such as machine translation, summarization, and generative question answering, we observed that these models yield sub-optimal performance in back-transliterating romanized Bangla. However, the available transliteration datasets lack the scale required to pre-train the data-intensive transformer models. Addressing the aforementioned challenges, our contributions can be summarized as follows:

1. We present the first large-scale Bangla transliteration dataset, BanglaTLit, with over 42.7k samples collected from diverse data sources, manually annotated, and verified by experts.

2. We also introduce BanglaTLit-PT, a pre-training corpus for romanized Bangla with over 245.7k samples.

3. We further pretrain five different transformer encoders on BanglaTLit-PT, achieving state-of-the-art performance in several romanized Bangla classification tasks.

4. We establish several baselines including multilingual models, Bangla seq2seq models, LLMs, and a novel encoder-decoder architecture on the proposed BanglaTLit dataset.

2 Related Work

2.1 English Back-Transliteration

Automatic back-transliteration has been a subject of interest in languages like Japanese (Goto et al., 2004; Bilac and Tanaka, 2004) and Korean (Kang and Choi, 2000), which have a rich history of incorporating foreign words into their vocabulary. With the rise of social media, Latin scripts became ubiquitous, leading to increased romanization of nearly all the languages. There is notable literature on back-transliterating Arabic (Chalabi and Gerges, 2012; Ameur et al., 2017; Guellil et al., 2018), Arabic dialects (Al-Badrashiny et al., 2014), Persian (Maleki and Ahrenberg, 2008), and Urdu (Bögel, 2012; Irvine et al., 2012), all of which rely on Perso-Arabic scripts.

Sequiera et al. (2014) explored several word-level back-transliteration strategies for Indic languages like Bangla, Gujarati, Kannada, Malayalam, and Tamil. The following years saw growth in several large-scale back-transliteration datasets for Indic languages Roark et al. (2020); Kunchukuttan et al. (2021); Madhani et al. (2022, 2023a). Hindi, which shares the same Indo-Aryan language family as Bangla but written in Devanagari scripts, has numerous works on back-transliteration (Sinha and Srinivasa, 2014; Parikh and Solorio, 2021). Baruah et al. (2024) explores back-transliteration of Assamese, which shares the same as Bangla.

2.2 Romanized Bangla Tasks

Romanized Bangla has been the source of numerous NLP tasks including sentiment analysis (Hassan et al., 2016; Tripto and Ali, 2018; Basri et al., 2021; Hossain et al., 2022), offensive speech detection (Raihan et al., 2023a; Islam et al., 2024), cyberbullying detection (Ahmed et al., 2021), product demand analysis (Hossain et al., 2022), event detection (Dey et al., 2021), etc. There has also been limited work on back-transliteration systems exclusive to Bangla only (UzZaman et al., 2006; Shibli et al., 2023; Kabiraj et al., 2023). However, the only publicly available Bangla transliteration dataset is proposed by Shibli et al. (2023), which is limited to $5k$ samples only.

2.3 Back-transliteration Methods

Transliteration has been approached in multiple rule-based, statistical, and machine learning-based approaches for languages differing by graphemes and phonemes (Mammadzada, 2023). Dasgupta et al. (2015) utilized statistical machine transliteration and multi-to-multi joint source channel models (Chen et al., 2011). Rizvee et al. (2022) employed a hybrid transliteration framework comprising phonetic transliteration, candidate answer transliteration, and spelling improvement.

Roark et al. (2020) worked on South Asian languages including Bangla utilizing multiple baselines such as, n-grams, LSTMs (Hochreiter and Schmidhuber, 1997), and transformers (Vaswani et al., 2017). Madhani et al. (2023b) fine-tuned the BERT (Devlin et al., 2019) and found promising results on Indic languages. Kabiraj et al. (2023) relied on neural machine translation (Sutskever et al., 2014). Shibli et al. (2023) established that few shot prompting on LLMs like GPT-3 (Brown et al., 2020).

3 Datasets

Following the limitations of existing Bangla transliteration datasets highlighted in Table 1, our dataset design can be simplified into two primary goals – creating a romanized Bangla pre-training corpus, BanglaTLit-PT and a Bangla transliteration dataset, BanglaTLit, comprising pairs of romanized Bangla and back-transliterated Bangla. We aim to ensure that the data sources are diverse, the back-transliterations are human-annotated, and samples are verified by experts.

3.1 BanglaTLit-PT

Multiple data sources are aggregated and extensive data cleaning is performed to create the BanglaTLit-PT corpus, which consists of 245.7k romanized samples.

Source	#Samples
BanglaTLit-PT (Pre-training Corpus)	
- TrickBd	141191
- TB-Emotion	79197
- BnSentMix	13081
- TB-Sentiment	5055
- Madhani et al. (2023b)	4170
- Shibli et al. (2023)	3033
Total	245727
BanglaTLit (Transliteration Dataset)	
- TrickBd	35613
- Madhani et al. (2023b)	4153
- Shibli et al. (2023)	2939
Total	42705
BanglaTLit Splits	
- Train	38705
- Validation	1500
- Test	2500
Total	42705

Table 2: Data source distribution of our pre-training corpus, BanglaTLit-PT and transliteration dataset, BanglaTLit.

3.1.1 Data Sourcing

The BanglaTLit-PT dataset is constructed by aggregating six diverse romanized Bangla datasets, seen in Table 2. We primarily sourced the data by collecting transliterated comments from the TrickBd website[1]. The comments span a wide range of topics, reflecting the diverse interests of the TrickBd community, which include social media, hacking, freelancing, offensive content, support queries, and service requests. The content diversity and variations in romanization provide a rich dataset suitable for transliteration. We further extend this dataset by incorporating romanized Bangla samples from five additional datasets: TB-Emotion, TB-Sentiment (Taawab et al., 2022), Madhani et al. (2023a), Shibli et al. (2023), and BnSentMix (Alam et al., 2024). After aggregating, our dataset has sources from TrickBd, Facebook, YouTube, Blogs, and Wikipedia.

3.1.2 Data Cleaning

After aggregating the data sources, we eliminated duplicate samples and discarded samples with two words or less. We also removed the BBcodes and hyperlinks as they are not relevant to the actual con-

[1]https://trickbd.com/

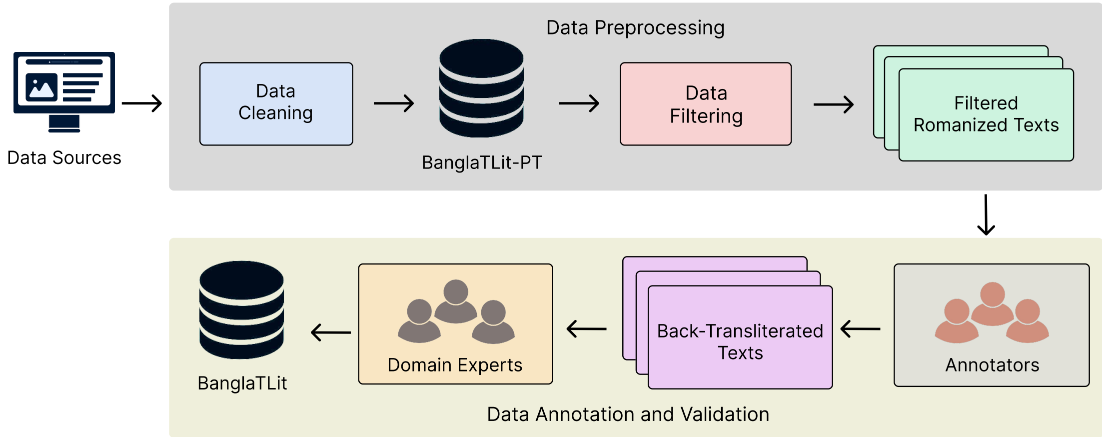

Figure 2: Pipeline of creating BanglaTLit-PT and BanglaTLit datasets. The data collected from various sources are aggregated and thoroughly cleaned to produce the BanglaTLit-PT corpus. The corpus is filtered, annotated, and verified by domain experts to create the BanglaTLit transliteration dataset.

tent and might produce ambiguity in the transliterated text. An arbitrary amount of white space was replaced with a single white space. Leading and trailing white spaces were also removed.

3.2 BanglaTLit

The BanglaTLit dataset contains romanized Bangla and its corresponding back-transliteration pairs by filtering 42.7k samples from the BanglaTLit-PT.

3.2.1 Data Filtering

We initially source the data from Madhani et al. (2023a) and Shibli et al. (2023) as both contained Bangla-Romanized Bangla sample pairs. We expanded the initial dataset by manually annotating 35.6k random samples from the TrickBd dataset. We selected the TrickBd dataset as it consists of comments spanning a wide range of topics *e.g.*, social media, hacking, and service requests. The content diversity and variations in romanization make it suitable for transliteration. Combining these datasets, we obtain a wide range of data sources for the BanglaTLit dataset, including Facebook, YouTube, Wikipedia, blog posts, and tech websites. Since most of the data originates from user comments, the dataset contains a good amount of textual noise, which replicates realistic romanization.

3.2.2 Data Annotation

After a rigorous manual validation of back-transliterations performed by both LLMs and human annotators, we concluded that human annotation is trustworthy and more robust. We hired 12 native Bangla speakers who are university undergraduates with at least 12 years of standard education and are familiar with social media, ensuring they have a solid understanding of romanized Bangla texts. Annotation guidelines were provided as outlined in Appendix A.1, along with our designed back-transliteration tool[2] developed using Google's transliterate API and presented in Appendix A.2.

3.2.3 Data Validation

We aimed to ensure that our dataset met the highest standards by hiring 3 Bangla linguistic experts to re-annotate 1000 random samples from the BanglaTLit dataset. We assessed the similarity of the expert annotations with our annotators using the BLEU, BERT, METEOR, ROUGE-1(F1), ROUGE-2(F1), and ROUGE-L(F1) score which were 72.55%, 96.32%, 83.89%, 87.69%, 49.68%, and 87.63% respectively, signifying the annotation done by the annotators strongly resembles the annotation done by linguistic experts.

We also asked the experts to annotate the same 200 samples and measured the inter-annotator agreement. The agreement levels were 92.38%, 58.27%, and 93.07% measured by Mean ROUGE-1(F1), Mean ROUGE-2(F1), and Mean ROUGE-L(F1) scores, respectively. The scores indicate considerably high inter-annotator agreement between the experts.

[2] https://rongali.vercel.app/

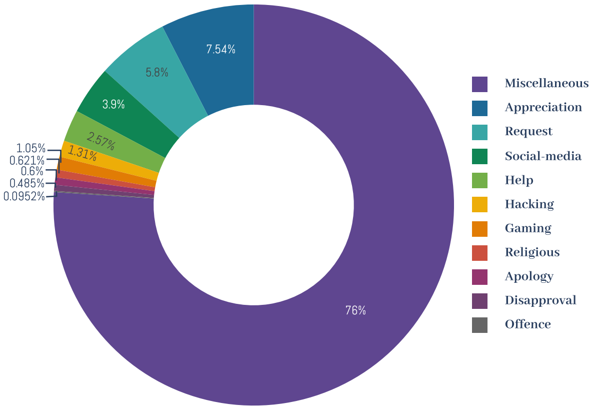

Figure 3: Composition of the categories of the transliterated sentences in BanglaTLit dataset.

Statistics	TL	BTL
Mean Character Length	59.24	58.28
Max Character Length	1406	1347
Min Character Length	3	4
Mean Word Count	10.35	10.51
Max Word Count	212	226
Min Word Count	2	2
Unique Word Count	81848	60644
Unique Sentence Count	42705	42471

Table 3: Dataset statistics of the Transliterated (TL) and Back-Transliterated (BTL) sample pairs of the BanglaTLit dataset.

3.2.4 Dataset Splits

We randomly split the BanglaTLit dataset by keeping 38.7k, 1.5k, and 2.5k samples for train, validation, and test splits, respectively. We also ensure that the samples from the validation and test splits are removed from the BanglaTLit-PT corpus.

3.3 Dataset Statistics

For a better understanding of the BanglaTLit dataset, we present several characters, word, and sentence-level statistics of the romanized and back-transliterated samples in Tab. 3. We also visualize the composing sentence categories in Fig. 3.

4 Methodology

Our methodology comprises two main components: i) Developing a Pretrained Encoder for Transliterated Bangla and ii) Employing the Encoder Aggregated Sequence Modeling.

4.1 TB Encoder

Pretrained models such as BanglaBERT and BanglishBERT are deficient in handling transliterated texts due to the lack of transliterated samples in their pretraining dataset. We enhance their performance by further pretraining them on the BanglaTLit-PT corpus to overcome the limitations. This involves utilizing Masked Language Modeling (MLM) loss (Devlin et al., 2019; Zhuang et al., 2021) as our pretraining objective. MLM randomly masks some input tokens in a sentence with a probability of 15%, replacing the masked ones t_m with a special token *[MASK]*. The model is then trained to predict these masked words based on the context provided by their surrounding words $t_{\setminus m}$. Formally, for a sentence $S = \{t_1, ..., t_T\}$ and mask indices $m \in \mathbb{N}^M$, the negative log-likelihood objective is defined as:

$$L_{MLM}(\theta) = -\mathbb{E}(S) \sim D \log P_\theta(t_m | t_{\setminus m})$$

where θ represents the trainable parameters. Each sentence S is sampled from the entire BanglaTLit-PT dataset D. After further pretraining the models on BanglaTLit-PT, we build Bangla transliteration-enhanced encoder models namely TB-Encoders (Transliterated Bangla Encoders)

4.2 TB-Encoder Aggregated T5 Models

Inspired by previous works (Shin and Lee, 2018; Hu et al., 2023; Zhou et al., 2020), we adopt a dual encoder-based model architecture to generate Bangla texts from transliterated Bangla texts. Given a Bangla transliterated text S, we tokenize it separately using the T5 tokenizer and the TB-model tokenizer to obtain the corresponding tokens. When the tokenizers yield sequences of different lengths, we pad them to the maximum token length. Subsequently, these tokenized sequences are inputted into their respective models to acquire separate representations. For a given text S, we obtain representations from the T5 encoder $\mathbf{h} = \{h_1, h_2, \ldots, h_n\}$ and the TB encoder $\mathbf{e} = \{e_1, e_2, \ldots, e_n\}$.

The representations **h** and **e** are then aggregated using two different feature aggregation techniques i) Summed-based Aggregation and ii) Concat-based Aggregation. In the summed-based aggregation method, each token representation is summed up:

$$H_i = h_i + e_i \quad \text{for} \quad i = 1, 2, \ldots, n$$

In the concatenation-based aggregation method,

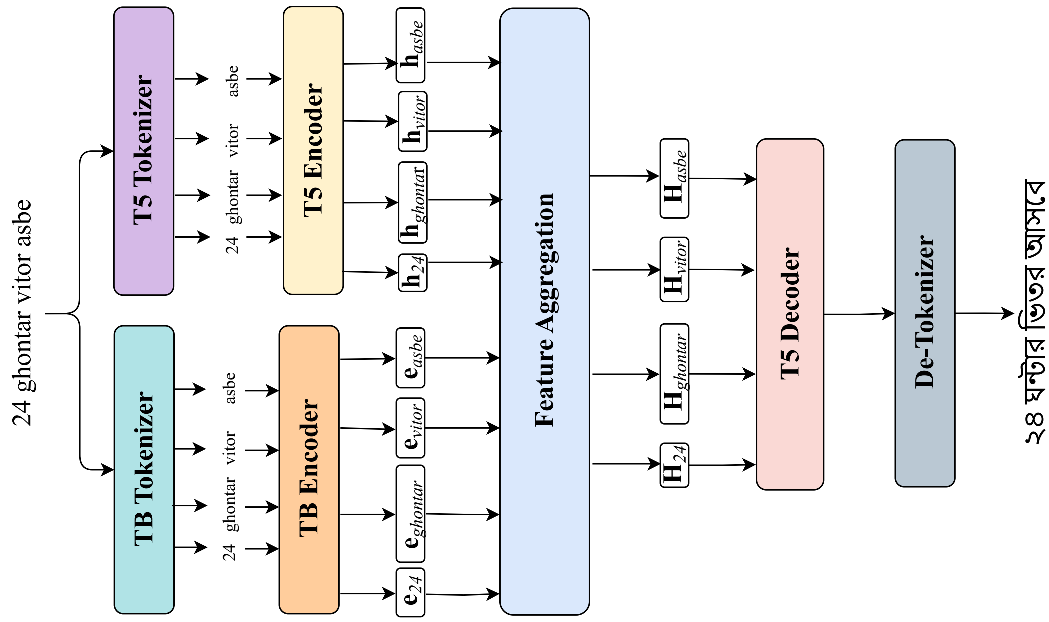

Figure 4: Model architecture of our proposed methodology. The transliterated text will first go through two tokenizers and encoders separately. Then the encoded tokens will be aggregated together and passed through T5 decoder and de-tokenizer to generate back-transliterated Bangla text.

the representation of each token is concatenated:

$$H_i = [h_i; e_i] \quad \text{for} \quad i = 1, 2, \ldots, n$$

Thus, we obtain the aggregated representations $\mathbf{H} = \{H_1, H_2, \ldots, H_n\}$, where $\mathbf{H}$ represents the combined representations resulting from the feature aggregation process. These aggregated representations are then passed into the T5 decoder to generate the corresponding Bangla text.

5 Experimental Results

5.1 TB Encoder Performance

To investigate the effectiveness of TB-Encoder models, we consider three different downstream tasks namely sentiment analysis on TB Sentiment (Taawab et al., 2022), offensive language detection on TB-OLID (Raihan et al., 2023b), and emotion detection on TB-Emotion (Faisal et al., 2024) datasets. A detailed description of these datasets is reported in Appendix A.6.

Firstly, we create strong baselines on these datasets by considering different types of pre-trained models, namely Bangla Language Models (LMs) – BanglishBERT (Bhattacharjee et al., 2021), BanglaBERT (Bhattacharjee et al., 2021), SahajBERT (Neuropark, 2021), and Vac-BERT (Bhattacharyya et al., 2023), Indian LMs – IndicBERT-v2 (Doddapaneni et al., 2023) and

Model	Performance Metric					
	TB-Sent		TB-OLID		TB-Emotion	
	Acc↑	F1↑	Acc↑	F1↑	Acc↑	F1↑
Bangla LM						
BanglishBERT	84.23	84.11	73.40	72.27	45.50	44.54
BanglaBERT	85.38	85.33	76.30	75.06	50.25	48.89
SahajBERT	76.54	76.54	71.57	70.29	39.75	38.79
Vac-BERT	78.85	78.78	68.12	67.36	35.00	33.62
Indian LM						
IndicBERT-v2	79.23	79.20	70.04	68.56	39.50	38.28
MuRIL	80.38	80.17	72.50	70.42	39.02	38.21
Multilingual LM						
XLM-RoBERTa	83.85	83.84	73.40	71.57	43.50	41.15
mDeBERTa-v3	80.38	80.37	67.80	67.74	34.25	32.94
mBERT	81.15	81.03	72.80	70.89	43.50	43.45
Character-based LM						
CharBERT	84.23	84.21	74.00	73.42	46.00	43.90
CharRoBERTa	84.23	84.08	71.90	69.30	40.50	39.15
Prompt-based LLM (0-shot)						
GPT 3.5 Turbo	85.39	85.38	71.80	70.96	40.62	37.24
LLaMa3-8B	69.62	69.61	56.00	55.96	21.74	10.55
TB Encoder (Ours)						
TB-BERT	84.23	84.13	74.50	74.29	49.25	48.89
TB-BanglaBERT	85.00	84.92	77.90	76.54	52.00	50.26
TB-BanglishBERT	86.15	86.07	74.40	73.58	51.25	51.08
TB-mBERT	85.77	85.72	76.30	75.52	50.25	48.85
TB-XLM-R	**88.85**	**88.79**	**78.50**	**77.76**	**54.50**	**53.40**

Table 4: Classification performance of the baselines and Transliterated Bangla (TB) Encoders for the downstream tasks – TB Sentiment Analysis (TB-Sent), TB Offensive Language Detection (TB-OLID), and TB Emotion Recognition (TB-Emotion). TB-x means that the associated model x has been further pre-trained on BanglaTLit-PT using MLM as described in section 4.1.

Model	ROUGE Score			BLEU Score			BERT Score (F1)	METEOR Score
	R-1	R-2	R-L	BLEU	Brevity Penalty	Length Ratio		
Encoder-Decoder LM								
mT5	56.02	19.83	55.90	12.48	76.13	0.82	86.43	48.71
byteT5	15.40	1.71	14.91	6.8e-5	11.28	0.25	72.50	6.88
BanglaT5-small	39.59	8.46	39.58	4.14	84.29	0.94	80.65	32.72
BanglaT5	73.06	33.00	73.13	31.09	91.16	0.95	92.71	69.12
BanglaT5_nmt_en_bn	75.74	34.84	76.14	36.19	**98.71**	1.08	94.05	74.07
Prompt-based LLM								
GPT-3.5 Turbo (0-shot)	66.21	26.18	66.64	20.73	97.94	1.11	90.06	59.97
GPT-4 Turbo (0-shot)	71.71	31.54	71.96	26.56	97.27	1.07	91.65	65.10
GPT-4o (0-shot)	66.62	26.96	67.24	19.28	98.22	1.11	89.37	58.88
LLaMa3-8B (3-shot)	56.05	17.34	56.56	11.01	95.80	1.04	86.61	46.81
Dual Encoder-Decoder LM (Ours)								
TB-BanglishBERT + BanglaT5	75.14	34.65	75.13	32.82	92.25	0.96	93.83	72.34
TB-BanglishBERT + BanglaT5_NMT	77.27	35.98	78.32	35.18	96.58	**0.97**	98.22	75.37
TB-XLM_R + BanglaT5	76.03	35.14	76.24	33.18	95.16	0.96	94.15	74.42
TB-XLM_R + BanglaT5_NMT	**78.92**	**36.56**	**79.75**	**36.07**	98.29	1.05	**98.82**	**78.14**

Table 5: Model benchmarking in our dataset on the test set. Fine-tuning BanglaT5 model beats prompt-based LLMs. Interestingly, GPT-4 shows very competitive results in our dataset. However, the performance of BanglaT5 is improved further while we incorporate our TB encoder models. The sum-based aggregation technique is used while modeling with TB-Encoder with T5 models.

MuRIL (Khanuja et al., 2021), Multilingual LMs – XLM-RoBERTa (Conneau et al., 2019), mBERT (Libovický et al., 2019), and mDeBERTa (He et al., 2021)), Character-based LMs – CharBERT (Ma et al., 2020) and CharRoBERTa (Ma et al., 2020) and prompt-based Large Language Models – GPT 3.5 Turbo (Brown et al., 2020) and LLaMa3-8B (Dubey et al., 2024).

Among the baselines, GPT-3.5 Turbo gives the best performance in TB-Sentiment and TB-OLID datasets with an F1 score of 85.38 and 73.42, respectively, and BanglaBERT gives the best performance in TB-Emotion dataset with an F1 score of 43.90. We observe a significant improvement in the scores using our TB-Encoders.

From Table 4, TB-XLM-R achieves the highest scores, particularly excelling in the TB Sentiment and TB-Emotion datasets. TB-XLM-R improves the accuracy on the TB Sentiment dataset by approximately 3.62% and the F1 score by 3.96% compared to the best performing existing model, GPT 3.5 Turbo. Similarly, in the TB-Emotion dataset, TB-XLM-R outperforms BanglaBERT by an accuracy margin of 4.25% and an F1 score margin of 4.51%. As TB-BanglishBERT and TB-XLM-R show the best results among the TB encoders, we consider these two models for creating the TB-encoder aggregated T5 models as baselines.

5.2 TB Dataset Benchmarking

For the benchmarking on back-transliteration, we consider several pre-trained seq2seq models – mT5(Xue et al., 2021), byte-T5 (Xue et al., 2022), and different variations of BanglaT5 (Bhattacharjee et al., 2023). Table-5 shows the results of the predictions done on the test dataset. The performance is evaluated with ROUGE, BLEU, BERT, and METEOR Score described in sec-A.5. BanglaT5_nmt_en_bn performs the best at generating the back-transliterated outputs, achieving the highest scores across all evaluation metrics. BanglaT5_nmt_en_bn records a ROUGE-1 score of 75.74%, ROUGE-2 score of 34.84%, ROUGE-L score of 76.14%, BLEU score of 36.19%, BERT score of 94.05%, and METEOR score of 74.07%.

In comparison, the prompt-based models, GPT-3.5 Turbo (0-shot), GPT-4 Turbo (0-shot), GPT-4o (0-shot), LLaMa3-8B (3-shot), also exhibit strong performance, with GPT-4 Turbo (0-shot) being the most notable. GPT-4 Turbo (0-shot) achieves a ROUGE-1 score of 71.71%, ROUGE-2 score of 31.54%, ROUGE-L score of 71.96%, BLEU score of 26.56%, BERT score of 91.65%, and METEOR score of 65.10%. Although GPT-4 Turbo (0-shot) performs well among the prompt-based models, it slightly lags behind BanglaT5_nmt_en_bn across all metrics.

Method	ROUGE Score			BLEU Score			BERT Score (F1)	METEOR Score
	R-1	R-2	R-L	BLEU	Brevity Penalty	Length Ratio		
Validation Set								
Sum-based								
TB-BanglishBERT + BanglaT5	68.98	28.88	69.06	26.74	92.93	0.96	92.22	64.45
TB-BanglishBERT + BanglaT5_NMT	72.16	30.35	72.80	32.02	98.24	1.08	94.80	69.57
TB-XLM_R + BanglaT5	69.77	29.25	69.23	27.08	96.80	**0.96**	**97.64**	65.92
TB-XLM_R + BanglaT5_NMT	**73.31**	**31.90**	**75.46**	**34.51**	**98.27**	1.05	96.48	**72.08**
Concat-based								
TB-BanglishBERT + BanglaT5	68.04	28.14	68.87	25.62	91.53	0.95	91.84	63.76
TB-BanglishBERT + BanglaT5_NMT	71.65	29.77	72.14	31.48	96.94	1.09	94.05	68.27
TB-XLM_R + BanglaT5	68.29	27.94	68.37	26.72	96.11	0.94	96.85	63.91
TB-XLM_R + BanglaT5_NMT	72.84	31.24	74.98	33.87	97.92	1.06	95.27	71.84
Test Set								
Sum-based								
TB-BanglishBERT + BanglaT5	75.14	34.65	75.13	32.82	92.25	0.96	93.83	72.34
TB-BanglishBERT + BanglaT5_NMT	77.27	35.98	78.32	35.18	96.58	**0.97**	98.22	75.37
TB-XLM_R + BanglaT5	76.03	35.14	76.24	33.18	95.16	0.96	94.15	74.42
TB-XLM_R + BanglaT5_NMT	**78.92**	**36.56**	**79.75**	**36.07**	**98.29**	1.05	**98.82**	**78.14**
Concat-based								
TB-BanglishBERT + BanglaT5	73.94	33.87	74.27	31.95	91.82	0.95	93.10	71.82
TB-BanglishBERT + BanglaT5_NMT	76.62	34.14	77.76	34.80	95.95	1.06	97.46	73.84
TB-XLM_R + BanglaT5	75.25	34.38	75.74	32.57	94.82	0.95	93.91	72.08
TB-XLM_R + BanglaT5_NMT	78.06	35.84	78.92	35.68	97.87	1.08	97.90	77.43

Table 6: Ablation Study on Different Feature Aggregation Techniques [Sum-based vs Concat-based] in our approach

From Table 5, BanglaT5 and Bangla_NMT_T5 models demonstrate superior performance when combined with TB-encoders. Integrating TB-BanglishBERT with either BanglaT5 or BanglaNMT encoder via sum-based aggregation results in a 2% increase in BLEU score and a 3% increase in METEOR score. The performance of BanglaT5 and Bangla_NMT_T5 is improved further if we aggregate the TB-XLM_R encoder representations with their corresponding encoder representations. TB-XLM_R combined with BanglaT5_NMT achieves the highest overall scores with an R1 score of 78.92%, a BLEU score of 36.07%, and a METEOR score of 78.14%. The performance of the models in the validation set is reported in table 8 in Appendix A.7. The ablation study for sum or concat-based aggregation of the TB-Encoder models is reported in table 6, which shows that `sum-based` aggregation techniques slightly perform better than `concat-based` aggregation techniques.

5.3 Prompt-based LLM Performance

We observed GPT-4-Turbo outperforming GPT-4 and LLaMa3-8B in zero-shot prompting. The GPT family models significantly outperform LLaMa-3B in few-shot settings as well. When not given explicit instructions regarding the output format, these models tend to generate reasoning behind their responses, often including superfluous text. Details of the prompting techniques are provided in Appendix A.8.

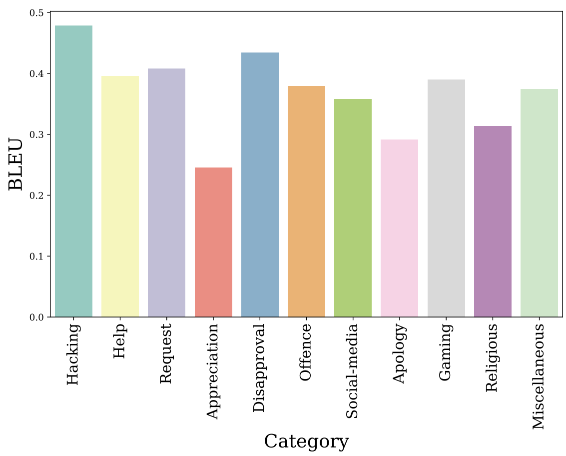

Figure 5: Category-wise BLEU scores for the predictions on the test set using the TB-XLM_R+BanglaT5_NMT.

Romanized Sentence	English Translation	Annotated Sentence	Prediction	R-1(F1)
Top 5 Most Accurate Predictions				
Vi kaj suru kore dici…	Brother, started doing the task	ভাই কাজ শুরু করে দিছি...	ভাই কাজ শুরু করে দিছি...	1.0
banglalink e cholbe?	Will it work with Banglalink?	বাংলালিংক এ চলবে?	বাংলালিংক এ চলবে?	1.0
apni try korsan	Did you try	আপনি ট্রাই করছেন	আপনি ট্রাই করছেন	1.0
kon browser diye try korbo?	Which browser should I try with?	কোন ব্রাউজার দিয়ে ট্রাই করবো?	কোন ব্রাউজার দিয়ে ট্রাই করবো?	1.0
card e tk add korbo kivabe	How do I add money to the card	কার্ড এ টাকা অ্যাড করবো কিভাবে	কার্ড এ টাকা অ্যাড করবো কিভাবে	1.0
Top 5 Least Accurate Predictions				
Kno msg aseni.	No message came	কোন ম্যাসেজ আসেনাই।	কেন মেসেজ আসেনি।	0.0
dbo inshah Allah . sorto projojjo	Will give Insha Allah. Condition applied	দিব ইনশাআল্লাহ। শর্ত প্রযোজ্য	দেব ইনশাল্লাহ আল্লাহ. সব কার্যকরী	0.0
he bro sobossy	Yes bro certainly	হ্যা ব্র অবশ্যই	হে ব্রো সোবসি	0.0
Meyad sheh bro	Validity is over bro	মেয়াদ শেষ ব্র	মিয়াদ শেহ ব্রো	0.0
authenticating dejhiye atke thake.ki korbo	Stuck at athenticating. What to do	অথেন্টিকেটিং দেখায় আটকে থাকে। কি করবো	অথেন্টিকেশন দিয়ে একে থাকে।কি করবো	0.18

Table 7: Most accurate and inaccurate predictions of the TB-XLM_R+BanglaT5_NMT on test set of our dataset.

6 Error Analysis

Table-7 presents the top five most accurate and inaccurate predictions produced by the TB-XLM_R+BanglaT5_NMT model on our test set. For the incorrect predictions, the model learns the literal word representation of the romanized sentences, which conflicts with the annotated representation of the transliteration. We also analyzed the category-wise model performance, based on BLEU Score, of the XLM_R+BanglaT5_NMT model on our test set. Figure 5 shows the distribution of the BLEU score for each category. The model demonstrates strong performance in the *Hacking*, *Request*, *Help*, and *Disapproval* categories while struggling with the *Appreciation*, *Apology*, and *Religious* categories.

We hypothesize that the model performs poorly in the above categories due to inconsistent spelling, varied use of diacritics, phonetic representations, idiomatic expressions, slang, and context-dependent language. For example, the word for "thank you" might appear as "tnx", "10x", "tenq", "10q", "dhonnobad", and religious greetings like "আসসালামু আলাইকুম" (peace be upon you) can have multiple transliterations, such as "Assalamu Alaikum" and "As-salamu alaykum". This flexibility in romanization makes it challenging for the model to learn consistent patterns and accurately translate these texts, unlike other straightforward categories like *Hacking* and *Help*.

When comparing the outputs of GPT-4 Turbo and LLaMa-3-8b, we found GPT-4 Turbo processing better back-transliteration capabilities than LLaMa-3-8B. As seen in Appendix Table 9, GPT-4 Turbo shows less error than LLaMa compared to the ground truth labeling. We also observe that the incorrect words produced by GPT-4-Turbo are the literal word representation in the transliterated text, which may not align with the annotations.

7 Conclusion

We propose a large-scale Bangla transliteration dataset and a romanized Bangla pre-training corpus. Experiments conducted on several baselines, including a novel dual encoder-decoder model architecture, show promising results in the task of romanized Bangla back-transliteration. Expanding the dataset to include more samples can be beneficial in training larger models or fine-tuning LLMs. Besides, transliteration of Bangla regional dialects and methods based on parameter-efficient fine-tuning of LLMs can be explored in the future. Our research opens new doors of expansion for low-resource languages like Bangla.

Acknowledgments

We would like to express our deepest gratitude to the sponsor of this project, Penta Global Limited, Bangladesh.

Authors Note. During the reviewing and rebuttal period, Bangladesh faced a tragic student movement against the reinstated government job quota system. The protests turned deadly when police forces killed several demonstrators, leading to national outrage and unrest. Over a thousand lives were lost, with many more injured. In the wake of these sacrifices, the nation gained independence once again from a regime of tyranny.

We honor the brave souls of the July student movement, reflecting on their courage, resilience, and fight for justice.

Limitations

The primary data source of BanglaTLit is the TrickBd dataset, which mostly contains comments related to tech support. While these comments capture the intricacies of romanization, they may be limited by being sourced from a single domain. Limited human resources hindered us from annotating a larger dataset. We included zero-shot and few-shot prompting of LLMs but did not fine-tune any LLM due to resource constraints. LLMs have shown promising results, and fine-tuning them should yield better performance in most romanized Bangla tasks.

Ethical Statement

The annotation work was undertaken by hired data annotators and validated by hired linguistic experts. Both the annotators and experts received hourly monetary compensation. For the annotators, we ensured the compensation was above the minimum wage and sufficient for university undergraduates. For experts, we adhered to industry-standard pay scales. Additionally, annotators and experts were assigned a low number of samples per hour to prevent any chance of overwork. To protect privacy, the identities of the annotators and experts were not recorded, and all personal identification information were removed from the dataset.

References

Md Tofael Ahmed, Maqsudur Rahman, Shafayet Nur, Azm Islam, and Dipankar Das. 2021. Deployment of machine learning and deep learning algorithms in detecting cyberbullying in bangla and romanized bangla text: A comparative study. In *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pages 1–10. IEEE.

Mohamed Al-Badrashiny, Ramy Eskander, Nizar Habash, and Owen Rambow. 2014. Automatic transliteration of romanized dialectal arabic. In *Proceedings of the eighteenth conference on computational natural language learning*, pages 30–38.

Sadia Alam, Md Farhan Ishmam, Navid Hasin Alvee, Md Shahnewaz Siddique, Md Azam Hossain, and Abu Raihan Mostofa Kamal. 2024. Bnsentmix: A diverse bengali-english code-mixed dataset for sentiment analysis. *arXiv preprint arXiv:2408.08964*.

Mohamed Seghir Hadj Ameur, Farid Meziane, and Ahmed Guessoum. 2017. Arabic machine transliteration using an attention-based encoder-decoder model. *Procedia Computer Science*, 117:287–297.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Hemanta Baruah, Sanasam Ranbir Singh, and Priyankoo Sarmah. 2024. AssameseBackTranslit: Back transliteration of Romanized Assamese social media text. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1627–1637, Torino, Italia. ELRA and ICCL.

Rabeya Basri, MF Mridha, Md Abdul Hamid, and Muhammad Mostafa Monowar. 2021. A deep learning based sentiment analysis on bang-lish disclosure. In *2021 National Computing Colleges Conference (NCCC)*, pages 1–6. IEEE.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Kazi Samin, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2021. Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. *arXiv preprint arXiv:2101.00204*.

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, Joydeep Mondal, Subhadip Maji, and Arnab Bhattacharya. 2023. Vacaspati: A diverse corpus of bangla literature. *arXiv preprint arXiv:2307.05083*.

Slaven Bilac and Hozumi Tanaka. 2004. Improving back-transliteration by combining information sources. In *International Conference on Natural Language Processing*, pages 216–223. Springer.

Tina Bögel. 2012. Urdu-roman transliteration via finite state transducers.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Achraf Chalabi and Hany Gerges. 2012. Romanized arabic transliteration. In *Proceedings of the Second Workshop on Advances in Text Input Methods*, pages 89–96.

Yu Chen, Rui Wang, and Yi Zhang. 2011. Statistical machine transliteration with multi-to-multi joint source channel model. In *Proceedings of the 3rd Named Entities Workshop (NEWS 2011)*, pages 101–105.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Tirthankar Dasgupta, Manjira Sinha, and Anupam Basu. 2015. Resource creation and development of an english-bangla back transliteration system. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 19(1):35–46.

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.

Noyon Dey, Md Sazzadur Rahman, Motahara Sabah Mredula, ASM Sanwar Hosen, and In-Ho Ra. 2021. Using machine learning to detect events on the basis of bengali and banglish facebook posts. *Electronics*, 10(19):2367.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426, Toronto, Canada. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Moshiur Rahman Faisal, Ashrin Mobashira Shifa, Md Hasibur Rahman, Mohammed Arif Uddin, and Rashedur M Rahman. 2024. Bengali and banglish: A monolingual dataset for emotion detection in linguistically diverse contexts.

Isao Goto, Naoto Kato, Terumasa Ehara, and Hideki Tanaka. 2004. Back transliteration from japanese to english using target english context. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 827–833.

Imane Guellil, Ahsan Adeel, Faical Azouaou, Fodil Benali, Ala Eddine Hachani, and Amir Hussain. 2018. Arabizi sentiment analysis based on transliteration and automatic corpus annotation. In *Proceedings of the 9th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 335–341.

Asif Hassan, Mohammad Rashedul Amin, Abul Kalam Al Azad, and Nabeel Mohammed. 2016. Sentiment analysis on bangla and romanized bangla text using deep recurrent models. In *2016 International Workshop on Computational Intelligence (IWCI)*, pages 51–56. IEEE.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Md Sabbir Hossain, Nishat Nayla, and Annajiat Alim Rassel. 2022. Product market demand analysis using nlp in banglish text with sentiment analysis and named entity recognition. In *2022 56th Annual Conference on Information Sciences and Systems (CISS)*, pages 166–171. IEEE.

Jia Cheng Hu, Roberto Cavicchioli, Giulia Berardinelli, and Alessandro Capotondi. 2023. Heterogeneous encoders scaling in the transformer for neural machine translation. *arXiv preprint arXiv:2312.15872*.

Ann Irvine, Jonathan Weese, and Chris Callison-Burch. 2012. Processing informal, romanized pakistani text messages. In *Proceedings of the Second Workshop on Language in Social Media*, pages 75–78.

Md Hasibul Islam, Kaniz Farzana, Ibrahim Khalil, Shaneen Ara, Md Ruhul Amin Shazid, and Md Humaion Kabir Mehedi. 2024. Unmasking toxicity: A comprehensive analysis of hate speech detection in banglish. In *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, pages 963–968. IEEE.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Shourov Kabiraj, Sajjad Waheed, and Zaber All Khaled. 2023. Transliteration from banglish to bengali language using neural machine translation. In *Proceedings of the Fourth International Conference on Trends in Computational and Cognitive Engineering: TCCE 2022*, pages 429–436. Springer.

Byung-Ju Kang and Key-Sun Choi. 2000. Automatic transliteration and back-transliteration by decision tree learning. In *LREC*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Anoop Kunchukuttan, Siddharth Jain, and Rahul Kejriwal. 2021. A large-scale evaluation of neural machine transliteration for Indic languages. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3469–3475, Online. Association for Computational Linguistics.

Jindřich Libovickỳ, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. CharBERT: Character-aware pre-trained language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 39–50, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yash Madhani, Mitesh M. Khapra, and Anoop Kunchukuttan. 2023a. Bhasa-abhijnaanam: Native-script and romanized language identification for 22 Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 816–826, Toronto, Canada. Association for Computational Linguistics.

Yash Madhani, Mitesh M Khapra, and Anoop Kunchukuttan. 2023b. Bhasha-abhijnaanam: Native-script and romanized language identification for 22 indic languages. *arXiv preprint arXiv:2305.15814*.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Ruchi Khapra, Vivek Seshadri, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra. 2022. Aksharantar: Towards building open transliteration tools for the next billion users. *arXiv preprint arXiv:2205.03018*.

Jalal Maleki and Lars Ahrenberg. 2008. Converting Romanized Persian to the Arabic writing systems. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Sabina Mammadzada. 2023. A review of existing transliteration approaches and methods. *International Journal of Multilingualism*, 20(3):1052–1066.

Ibraheem Muhammad Moosa, Mahmud Elahi Akhter, and Ashfia Binte Habib. 2022. Does transliteration help multilingual language modeling? *arXiv preprint arXiv:2201.12501*.

Neuropark. 2021. sahajbert. Accessed: 2024-09-22.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Dwija Parikh and Thamar Solorio. 2021. Normalization and back-transliteration for code-switched data. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 119–124.

Md Nishat Raihan, Umma Hani Tanmoy, Anika Binte Islam, Kai North, Tharindu Ranasinghe, Antonios Anastasopoulos, and Marcos Zampieri. 2023a. Offensive language identification in transliterated and code-mixed bangla. *arXiv preprint arXiv:2311.15023*.

Md Nishat Raihan et al. 2023b. Offensive language identification in transliterated and code-mixed bangla. *arXiv preprint arXiv:2311.15023*.

Redwan Ahmed Rizvee, Asif Mahmood, Shakur Shams Mullick, and Sajjadul Hakim. 2022. Arobust three-stage hybrid framework for english to bangla transliteration. *International Journal on Natural Language Computing*, 11(15):02.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. Processing south asian languages written in the latin script: the dakshina dataset. *arXiv preprint arXiv:2007.01176*.

Royal Denzil Sequiera, Shashank S Rao, and BR Shambavi. 2014. Word-level language identification and back transliteration of romanized text. In *Proceedings of the 6th Annual Meeting of the Forum for Information Retrieval Evaluation*, pages 70–73.

GM Shahariar Shibli, Md Tanvir Rouf Shawon, Anik Hassan Nibir, Md Zabed Miandad, and Nibir Chandra Mandal. 2023. Automatic back transliteration of romanized bengali (banglish) to bengali. *Iran Journal of Computer Science*, 6(1):69–80.

Jaehun Shin and Jong-Hyeok Lee. 2018. Multi-encoder transformer network for automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 840–845, Belgium, Brussels. Association for Computational Linguistics.

Navneet Sinha and Gowri Srinivasa. 2014. Hindi-english language identification, named entity recognition and back transliteration: shared task system description. In *Working Notes os Shared Task on Transliterated Search at Forum for Information Retrieval Evaluation FIRE'14*, volume 2014.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks.

Advances in neural information processing systems, 27.

Abdullah Al Taawab, Lubaba Tasnia, Mondira Dhar, and Md Humaion Kabir Mehedi. 2022. Positive and negative corpus. V3.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6. IEEE.

Naushad UzZaman, Arnab Zaheen, and Mumit Khan. 2006. A comprehensive roman (english)-to-bangla transliteration scheme.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

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.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Xinyuan Zhou, Emre Yılmaz, Yanhua Long, Yijie Li, and Haizhou Li. 2020. Multi-encoder-decoder transformer for code-switching speech recognition. *arXiv preprint arXiv:2006.10414*.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Appendix

A.1 Annotation Guidelines

The annotators followed the guidelines attached below while annotating the transliterated texts.

1. Spelling mistakes should not be included in the Bengali annotation.
2. Contractions should not be included in the Bengali annotation.
3. If the transliterated text contains emojis/emoticons, they should be placed as-is in the sentence's appropriate location(s).
4. If the transliterated text contains URLs/code snippets/command line arguments, they should be placed as-is in the sentence's appropriate location(s).
5. If the transliterated text contains improper usage of punctuation marks, they should be kept as-is in the transliterated sentence.
6. Colloquialism should be maintained throughout the translation.
7. English words should not be translated into Bengali. Only transliterations are accepted.
8. If acronyms/abbreviations are usually read letter by letter, they should be included in the annotation. This is only meant for acronyms/abbreviations that are pronounced that way.
9. Any mentions of names or PII (Personal Identifiable Information) should be anonymized in the transliteration. Modification of the original text is allowed in these cases.
10. If the transliterated text only contains Bengali letters, a URL, and no actual transliterated content, they should be skipped.

A.2 Annotation Tools

We developed the Rongali tool[3] using Google's transliteration API. As depicted in figure 6, the features include suggestions of the back-transliterated words, suggestions of abbreviated words in Bengali, automated replacement of a single period with

[3] https://rongali.vercel.app/

Model	ROUGE Score			BLEU Score			BERT Score (F1)	METEOR Score
	R-1	R-2	R-L	BLEU	Brevity Penalty	Length Ratio		
Encoder Decoder LM								
mT5	51.79	16.64	51.39	09.38	75.45	0.82	84.91	44.30
byteT5	13.90	1.65	13.51	6.4e-5	11.00	0.25	71.76	6.37
BanglaT5-small	37.53	7.34	37.03	03.40	84.11	0.96	79.79	30.35
BanglaT5	67.85	27.80	67.56	24.53	90.85	0.95	90.90	63.54
BanglaT5_nmt_en_bn	70.49	29.63	70.60	29.36	98.02	1.04	92.30	68.46
Prompt-based LLM								
GPT-3.5 Turbo (0-shot)	61.69	22.56	61.74	15.70	98.08	1.14	89.10	55.53
GPT-4 Turbo (0-shot)	66.27	26.53	66.26	20.51	**98.56**	1.13	90.06	61.29
GPT-4o (0-shot)	61.72	22.73	62.01	16.14	98.23	1.13	89.54	55.76
LLaMa3-8B (3-shot)	53.23	15.71	53.24	10.96	95.98	1.08	86.09	46.16
Dual Encoder-Decoder LM (Ours)								
TB-BanglishBERT + BanglaT5	68.98	28.88	69.06	26.74	92.93	0.96	92.22	64.45
TB-BanglishBERT + BanglaT5_NMT	72.16	30.35	72.80	32.02	98.24	1.08	94.80	69.57
TB-XLM_R + BanglaT5	69.77	29.25	69.23	27.08	96.80	**0.96**	**97.64**	65.92
TB-XLM_R + BanglaT5_NMT	**73.31**	**31.90**	**75.46**	**34.51**	98.27	1.05	96.48	**72.08**

Table 8: The Performance of the models on the validation set. Summed-based aggregation technique is used while modeling with TB-Encoder with T5 models

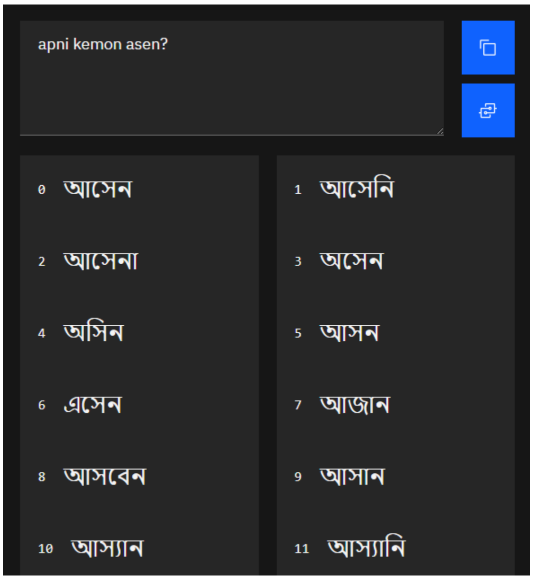

Figure 6: Annotation tool used by the annotators to back-transliterate the transliterated sentences.

'|', keeping multiple periods as ellipses, punctuation as-is in the appropriate location(s) of the sentence. For each word in the transliterated sentence in the text box, the tool suggests the corresponding back-transliterated word and its abbreviation in its suggestion box. The correct suggested word can be selected by selecting its serial number in the suggestion box.

A.3 Word Cloud

The fig.7 shows the word cloud on our whole dataset, which shows the visual representation of the frequency distribution of the words in the dataset. As the most frequent categories are *Appreciation* and *Request* after *Miscellaneous* 3, the highlighted words in fig.7 show words that fall in that category.

A.4 Experiment Setup

All the further pretraining and seq2seq encoder-decoder models were imported from HuggingFace Transformers library (Wolf et al., 2020). For seq2seq the output Bangla sentences were first normalized with the `csebuetnlp normalizer`[4]. In pretraining experiments, we ran the models for 10 epochs in the pretraining transliterated Bangla dataset. In seq2seq experimental setup, the training was conducted over 10 epochs. Model checkpoints were saved epoch-wise, with a limit of three checkpoints retained throughout the training process.

In the pretraining stage, the batch size was 32, and the learning rate = $1 * 10^{-5}$. For the experiment on the downstream tasks, we also consider the same model configurations but with batch size = 16. For the encoder-decoder models, We utilized a per-device batch size of 4 and employed a learning rate of $2 * 10^{-5}$ with L2 regulariza-

[4]https://github.com/csebuetnlp/normalizer.git

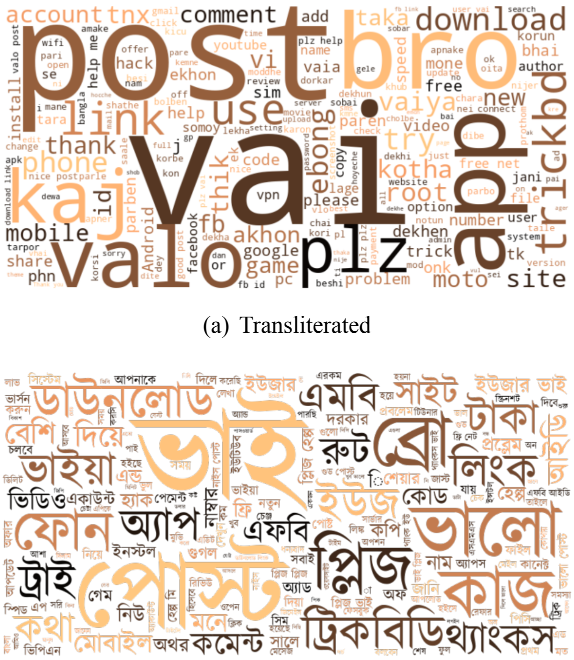

Figure 7: Word cloud constructed from our dataset taking transliterated and back-transliterated texts separately.

tion (weight decay of 0.01). To facilitate experimentation and analysis, we integrated logging with Weights and Biases to streamline the tracking of training progress. For the prompt-based models that we used, GPT-3.5 Turbo, GPT-4 Turbo, GPT-4o, and LLaMa3-8B, we used prompting to generate the texts. The GPT-based models were accessed using their OPENAI API KEY. LLaMa3-8B model was accessed through AWS Bedrock. For the GPT-based models, 0 shots prompting were used to generate the texts while that for LLaMa3-8B required 3-shots prompting **??**. We trained the LMs on with NVIDIA Tesla P100 GPUs and 2xT4 GPUs with 16GB RAM.

A.5 Performance Metrics

The performance metrics used to evaluate the performance of the models are ROUGE Scores, ROUGE-1 F1, ROUGE-2 F1, ROUGE-L F1, BLEU Scores, brevity penalty, length ratio, BERT Score, METEOR Score.

ROUGE Scores The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics commonly used to evaluate the quality of ground truth and machine translation. ROUGE scores measure the overlap of n-grams between the generated text, i.e. the annotated back transliterated text, and the reference text. The key variants of ROUGE used are, ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L) which measures the overlap of unigrams, bigrams, and longest common subsequence (LCS) between the generated and reference texts, respectively. We used the F1 score of the ROUGE scores. ROUGE-1 F1 captures the basic content similarity, and ROUGE-2 F1 assesses the fluency and coherence of the generated text (Lin, 2004).

BERT Score The BERT (Bidirectional Encoder Representations from Transformers) score evaluates the semantic similarity between the generated and reference texts. It provides precision, recall, and F1 scores based on contextual embedding. We used the BERT F1 score for assessing the performance (Zhang et al., 2020).

BLEU Score The BLEU (Bilingual Evaluation Understudy) score measures the n-gram precision of the translated text with respect to one or more reference translations. We used BLEU Score, Brevity Penalty, and Length Ratio for evaluation. The brevity penalty is used to penalize translations that are too short. It is calculated based on the ratio of the length of the generated text to the length of the reference text. The length ratio is the ratio of the length of the generated text to the length of the reference text (Papineni et al., 2002).

METEOR Score The METEOR (Metric for Evaluation of Translation with Explicit ORdering) score evaluates the generated text based on aligning words and phrases with the reference text. It considers precision, recall, and harmonic mean, incorporating synonymy and stemming (Banerjee and Lavie, 2005).

A.6 Downstream Task Datasets

TB-OLID This dataset (Raihan et al., 2023b) contains 5,000 Facebook comments, of which 2,381 are offensive and 2,619 are non-offensive. The train and test sets contained 4,000 and 1,000 comments, respectively. The selection process involved filtering out non-Latin script comments from the initial corpus of 100,000 comments, followed by the offensive-keyword-based search for offensive comments. The comments are manually annotated following the OLID hierarchical taxonomy - with a label for whether it is offensive or not and one for the target (individual,

Transliterated	English Translation	Back-transliterated	LLaMa3-8B	GPT-4-Turbo
Tnx bro ato sundor akta gift dewar jonno	Thanks bro for giving this lovely gift	থ্যাংক্স ব্রো এত সুন্দর একটা গিফট দেওয়ার জন্য	ধন্যবাদ ভাই আতো সুন্দর একটা গিফট দিয়েছেন দেওয়ার জন্য	টিএনএক্স ব্রো এত সুন্দর একটা গিফট দেওয়ার জন্য
hmm playstore update den	Update the playstore	হুম প্লেস্টোর আপডেট দেন	হাম্ প্লেস্টোর আপডেট দেন	হ্মম প্লেস্টোর আপডেট দেন
voice number niye akta post den	Make a post about voice number	ভয়েস নাম্বার নিয়ে একটা পোস্ট দেন	কল নাম্বার নিয়ে একটা পোস্ট দেন	ভয়েস নাম্বার নিয়ে একটা পোস্ট দেন
bro kom takay beshi fb mb kivabe kene?	Bro how to buy much more facebook MB with little money?	ব্রো কম টাকায় বেশি এফবি এমবি কিভাবে কেনে ?	ভাই কম টাকায় বেশি ফেসবুক মেসেঞ্জার কিভাবে করেন?	ব্রো কম টাকায় বেশি এফবি এমবি কিভাবে কেনে?
sob to dekhi fak	Everything looks fake	সব তো দেখি ফেক	সেব তো দেখি ফাক	সব তো দেখি ফাক
earn to hobe na..coz..keo ad dite asbe na so majkhane ami payment dite dite fokir hoye jabo..	There will be no income..because..no one will come to give ad in the meanwhile I will become a beggar by making the payments	আর্ন তো হবে না ..কজ..কেউ এড দিতে আসবে না সো মাঝখানে আমি পেমেন্ট দিতে দিতে ফকির হয়ে যাবো ..	আয় হবে না..কারণ কেউ আদ দিতে আসবে না সে মজখানে আমি পেমেন্ট দিতে দিতে ফকির হয়ে যাবো..	আর্ন তো হবে না..কজ..কেউ এড দিতে আসবে না সো মাঝখানে আমি পেমেন্ট দিতে দিতে ফকির হয়ে যাবো..
gmail create korar system den	Give the system to create gmail	জিমেইল ক্রিয়েট করার সিস্টেম দেন	গ্রামার মাস্টার আইও ক্রিয়েট করার সিস্টেম দেন	জিমেইল ক্রিয়েট করার সিস্টেম দেন
vhaiya ata vdo hobe thik moto hoitace na	Brother it will be a video it is not working properly	ভাইয়া এটা ভিডিও হবে ঠিক মতো হইতাছে না	ভাইয়া আটা ভিডিও হবে ঠিক মোট হয়তাছে না	ভাইয়া আটা ভিডিও হবে ঠিক মতো হইতাছে না
tnx 2 gb paici vai	Thanks got 2 GB	থ্যাংক্স ২ জিবি পাইসি ভাই	ধন্যবাদ ২ গুগল পাইসি ভাই	টিএনএক্স ২ জিবি পেয়েছি ভাই
Theme install dewar somoy opera diea hobe na	Installing theme will not work with opera	থিম ইনস্টল দেওয়ার সময় অপেরা দিয়ে হবে না	থিম ইন্সটল দেওয়ার সময় অপেরা দীয়া হবে না	থিম ইনস্টল দেওয়ার সময় অপেরা দিয়ে হবে না

Table 9: Performance comparison of LLaMa-3-8B and GPT-4-Turbo models for few samples from the test set of our dataset.

group, untargeted). Also labeled is whether a comment contains code-mixing along with the transliteration.

TB-Sentiment This dataset (Taawab et al., 2022), drawn from Positive and Negative Corpus[5], is a collection of 1,300 comments scraped from Facebook and YouTube, of which 647 are positive and 653 express negative sentiment. We split these 80:20 into the train and test sets.

TB-Emotion This dataset (Faisal et al., 2024) contains a total of 80,098 data entries comprising both Bengali and Banglish. It is organized into six distinct emotional categories: anger(15,179), disgust(13,098), fear(7,565), joy(17,836), surprise(10,107), and sadness(16,309). It offers a diverse and rich dataset sourced from platforms such as EmoNoBa, UBMEC, MONOVAB, and comments from YouTube and Twitter posts via official APIs. The collected samples are annotated by majority voting. Then, after duplicate removal, the dataset was transliterated. While experimenting, we considered 1600 and 400 samples for training and testing respectively instead of the total dataset.

A.7 Valdation Set Results

Table 8 shows the performance of the models in generating the back-transliterated text after training the models with the training dataset. For the Language Models (LMs), the configuration used for generating the validation dataset is the same as that used in the test dataset. For the prompt-based models, the prompts used for the validation dataset are the same as those used in test set **??**. The performance of the models is evaluated with the performance metrics described in A.5.

A.8 Prompts

The following prompts are used for the classification and back-transliteration tasks for the prompt-based models, Gemma-2B, LLaMa-8B, GPT-3.5 Turbo, GPT-4 Turbo, GPT-4o and LLaMa3-8B.

Generalized prompt for downstream classification Tasks on Bengali transliterated texts using GPT models

You are an expert Bengali <task_name> assistant. You always classify <task_name> from the given English transliterated sentence. You always have to abide by the conditions that are mentioned below:
CONDITION 1: Classify from these <n> classes.
CONDITION 2: If the sentence belongs to <class_1>, then output 0, if to <class_2> then output 1, <for_n_classes>
Here is the sentence:
<transliterated sentence>
Based on the above sentence, give the <task_output> with an integer like the following format:
Q# <answer>

[5]https://data.mendeley.com/datasets/s6mtp2zzpc/3

Prompt for generating back-transliteration on test and validation set from our dataset using GPT models

You are an expert Bengali back-transliteration assistant. You always generate the Bangla phonetic back transliteration in Bengali from
the given English transliterated sentence. You always have to abide by the conditions that are mentioned below:
CONDITION 1: Do not translate English words. Instead, write the Bengali phonetic version in Bangla
CONDITION 2: Keep the punctuation and emojis as it is.
Here is the sentence:
<transliterated sentence>
Based on the above sentence, generate the back-transliterated sentence in the following format:
Q# <generated back-transliterated sentence>

Prompt for generating back-transliteration on test and validation set from our dataset using LLaMa3-8B

<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
You are an expert back-transliteration assistant. You always back-translate Bangla from the given English transliterated sentence. You always have to abide by the conditions that are mentioned below:
CONDITION 1: Do not translate English words. Instead, write the Bengali phonetic version in Bangla.
CONDITION 2: Keep the emojis and punctuations as it is.
The examples are given as:
TB: Ami.bai,,,,, hecker????
output: আমি।ভাই,,,,, হ্যাকার????
TB: rana vai tuner dan plz
output: ইউজার ভাই টিউনার দেন প্লিজ
TB: clg e jai
output: কলেজ এ যাই
<|eot_id|>
<|start_header_id|>user<|end_header_id|>
Here is the sentence:
<transliterated sentence>
Based on the above sentence, do the back-transliteration and give a single sentence in the following format.
Q# <generated back-transliterated sentence>
<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>

Generalized prompt for downstream classification tasks on Bengali transliterated texts using LLaMa3-8B

<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
You are an expert <task_name> detection assistant. You always classify <task_name> from the given English transliterated sentence. You always have to abide by the conditions that are mentioned below:
CONDITION 1: Classify from these <n> classes.
CONDITION 2: If the sentence belongs to <class_1>, then output 0, if to <class_2> then output 1, <for_n_classes>
<|eot_id|>
<|start_header_id|>user<|end_header_id|>
Here is the sentence:
<transliterated sentence>
Based on the above sentence classify and give answer an integer.
<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>