Robustness of LLMs to Transliteration Perturbations in Bangla

Fabiha Haider[1] *, Md Farhan Ishmam[2] *, Fariha Tanjim Shifat[1,3], Md Tasmim Rahman[1], Md Fahim[1,4] †, Md Farhad Alam Bhuiyan[1]

[1]*Penta Global Limited* [2]*University of Utah* [3]*Missouri S&T*
[4]*CCDS, Independent University, Bangladesh*

*Equal Contribution †Project Lead

{fabihahaider4,farhan.ishmam,fahimcse381}@gmail.com

Abstract

Bangla text on the internet often appears in mixed scripts that combine native Bangla characters with their Romanized transliterations. To ensure practical usability, language models should be robust to naturally occurring script mixing. Our work investigates the robustness of current LLMs and Bangla language models under various transliteration-based textual perturbations, *i.e.*, we augment portions of existing Bangla datasets using transliteration. Specifically, we replace words and sentences with their transliterated text to emulate realistic script mixing, and similarly, replace the top k salient words to emulate adversarial script mixing. Our experiments reveal interesting behavioral insights and vulnerabilities to robustness in language models for Bangla, which can be crucial for deploying such models in real-world scenarios and enhancing their overall robustness. Our code is available at: https://github.com/farhanishmam/BTL-Robustness.

1 Introduction

In the digital era, Bangla is often written in its romanized form using English scripts due to the ubiquity of the QWERTY layout (Haider et al., 2024). With the growing popularity of Bangla keyboard layouts, particularly among mobile users, Bangla-English mixed script texts have become more common. This phenomenon is known as script-mixing, where multiple scripts are used in a single piece of text (Srivastava et al., 2020).

The current generation of Large Language Models (LLMs) has also excelled in tasks on transliterated or romanized Bangla (Fahim et al., 2024). However, their robustness to textual perturbations in Bangla has yet to be evaluated. Textual perturbation refers to any form of change or modification to the input text that can potentially impact the model's performance in a given task (Li et al., 2020a). Such perturbations can emulate realistic conditions (Moradi and Samwald, 2021) (e.g., removal or replacement of a word) or adversarial conditions (Li et al., 2018) (e.g., removal of most salient tokens (Raiyan et al., 2025)). Our work explores a form of replacement-based perturbation where words or sentences in the original Bangla scripts are replaced by their transliterations.

Current datasets in Bangla are limited to a single script, either in Bangla (Hasan et al., 2020; Islam et al., 2021) or English (Fahim et al., 2024). While code-mixed texts have been a topic of interest, where Bangla and English words are mixed, the datasets are usually limited to the English scripts (Alam et al., 2024). Evaluation of LLMs under script mixing can be crucial for deploying the model in realistic scenarios. We hence propose a scalable augmentation strategy to produce script-mixed text in Bangla and evaluate the robustness of models against such forms of perturbations. Our contributions can be summarized as:

- We present the first study to evaluate LLMs in three Bangla transliteration-based perturbations encompassing both realistic and adversarial settings.
- Our augmentation framework can be used to produce text that emulates script-mixing in Bangla at scale.
- Our experiments on a rich suite of closed-sourced and open-sourced LLMs, as well as Bangla language models, highlight the robustness vulnerability in Bangla.

2 Related Work

2.1 Textual Perturbation

Textual perturbations are either formulated as adversarial attacks that exploit the vulnerability of a system using an input, often tailored to that particular model (Li et al., 2020a), or common

Proceedings of the Second Workshop on Bangla Language Processing (BLP-2025), pages 338–346

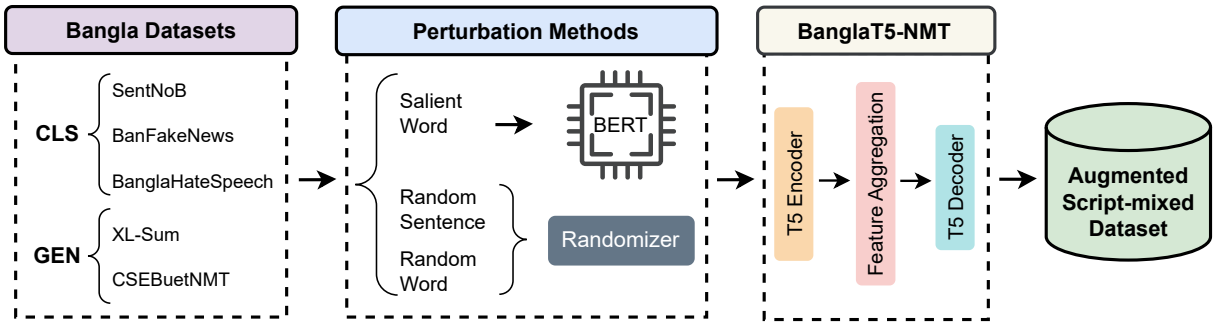

Figure 1: High level overview of our dataset perturbation pipeline.

perturbations that are typically encountered by texts in realistic scenarios (Moradi and Samwald, 2021). Adversarial perturbations saw some earlier success with rule-based methods, *e.g.*, synonym replacement, in both black-box and white-box settings (Jin et al., 2020; Alzantot et al., 2018). Few methods relied on using language models to generate adversarial examples (Li et al., 2020b; Garg and Ramakrishnan, 2020).

Realistic textual perturbations include character and word-level perturbations, *e.g.*, insertion, deletion, and replacement, which are used to simulate realistic errors in text (Moradi and Samwald, 2021; Le et al., 2022). Ours similarly uses word and sentence-level transliteration as realistic perturbations to simulate script mixing in text. We also experiment with the transliteration of the most salient word as a form of adversarial perturbation.

2.2 Robustness of Language Models

The robustness of language models refers to their inherent ability to sustain performance when exposed to input variations (Morris et al., 2020). While such studies on robustness are prevalent in English (Moradi and Samwald, 2021; Li et al., 2020a), the challenge is exacerbated in multilingual and low-resource contexts (Kaing et al., 2024). Robustness also refers to the language model's generalization capabilities under distribution shifts (Hendrycks et al., 2020). Our study focuses on evaluating this robustness in low-resource contexts, specifically examining the Bangla language under script distribution shifts.

2.3 Transliteration, Code-mixing, and Script-mixing

Transliterated texts, where native words are represented in foreign scripts, have been common in Indic languages through romanization (Madhani et al., 2023). This phenomenon is particularly prevalent in Bangla, where romanized scripts are used to write Bangla text. Current language models have shown strong performance on back-transliteration, *i.e.*, producing the original Bangla text from transliterated input (Fahim et al., 2024). Several downstream tasks have been explored on transliterated Bangla, including sentiment analysis (Hassan et al., 2016) and hate speech detection (Haider et al., 2024).

A closely related setting is code-switching or code-mixing, where words from multiple languages appear in the same text using different scripts (Sheth et al., 2025). Code-switching between Bangla and English is particularly common among Bangla speakers (Alam et al., 2024), though LLMs have shown degraded performance on such code-switched text (Mohamed et al., 2025). Our work differs in that we evaluate the robustness through Bangla dataset augmentations that mimic script-mixing (Srivastava et al., 2020), *i.e.*, multiple scripts coexist within the same text block.

3 Methodology

Our framework involves applying three types of perturbations to popular Bangla classification and generation datasets, as shown in Fig. 1.

3.1 Textual Perturbation

Each perturbation $p \in \mathcal{P}$ is defined as a function $p : \mathcal{T} \rightarrow \mathcal{T}^{\text{tr}}$, that takes an input text in native Bangla scripts B to produce text in transliterated scripts. For a model f, we quantify the average case performance as robustness over the test set distribution $\mathcal{D}$ (Hendrycks and Dietterich, 2019; Ishmam et al., 2025),

$$\mathbb{E}_{p\sim\mathcal{P}}[\mathbb{P}_{(B,y)\sim\mathcal{D}}((f(p(B)) = y)].$$

Dataset	SentNob	BanFakeNews	BanglaHateSpeech	XL-Sum	CSEBuetNMT
Total Samples	12k	49k	30K	8k	2.7M
Evaluated Samples	1568	2000	750	1012	1000
Vocab Size	24K	415K	64K	226K	1.3M
Min Word Length	3	1	1	7	1
Max Word Length	93	4650	537	3726	8353
Min Sentence Length	1	1	1	1	1
Max Sentence Length	20	679	78	370	262

Table 1: Statistics and number of samples taken for evaluation from the evaluation datasets.

The textual perturbation is implemented as a function that takes a slice of the input text and passes it to a transliteration model $f^{\text{tr}} : \mathcal{T} \rightarrow \mathcal{T}^{\text{tr}}$ to produce the transliterated text. The slicing of the text differs for each perturbation and has been defined in the latter sections.

3.1.1 Random Word Perturbation

For each word w_i in an input text $B = \{w_1, w_2, \ldots w_n\} \in \mathcal{T}$ and a random word-level mask vector,

$$M = \{m_1, m_2, \ldots, m_n\}, \quad m_i \sim \text{Bernoulli}(p),$$

where $p \in [0, 1]$ is the probability of perturbing a word, the random word perturbation can be defined:

$$p_{\text{rw}}(w_i) = \begin{cases} f^{\text{tr}}(w_i), & \text{if } m_i = 1, \\ w_i, & o/w. \end{cases} \quad (1)$$

3.1.2 Random Sentence Perturbation

Similar to §3.1.1, the sentence perturbation segments the input text B into sentences, $B = \{w_{1:i_1}, w_{i_1+1:i_2}, \ldots w_{i_{n-1}+1:n}\} \in \mathcal{T}$, and uses sentence-level mask vectors. Following Eq.1, we define random sentence perturbation,

$$p_{\text{rs}}(w_{i:j}) = \begin{cases} f^{\text{tr}}(w_{i:j}), & \text{if } m_i = 1, \\ w_{i:j}, & o/w. \end{cases} \quad (2)$$

3.1.3 Salient Word Perturbation

Let $s_i = S(w_i, B)$ denote the saliency score assigned to word w_i, measuring its influence on the model's output. We calculate the saliency scores by averaging the attention scores of a BanglaBERT model (Bhattacharjee et al., 2022) across the sequence length, heads, and layers. We define a proportion p, and organize the words based on the descending order of saliency scores. We now define the set of top-p salient word indices,

$$\mathcal{I}_{sal} = \{i | s_i \text{ is among top } p \text{ scores}\}.$$

The salient word perturbation can be similarly defined as:

$$p_{\text{sal}}(w_i) = \begin{cases} f^{\text{tr}}(w_i), & \text{if } i \in \mathcal{I}_{sal}, \\ w_i, & o/w. \end{cases} \quad (3)$$

For each perturbation, the probability of perturbation p is taken as 20%. We use the BanglaT5_NMT model (Bhattacharjee et al., 2023) fine-tuned on the BanglaTLit dataset (Fahim et al., 2024) as our transliteration model f^{tr}.

3.2 Tasks & Datasets

We evaluate on five tasks: machine translation with CSEBuetNMT dataset (Hasan et al., 2020), hate speech detection with BanglaHateSpeech dataset (Romim et al., 2021), sentiment analysis with Sent-Nob dataset (Islam et al., 2021), fake news detection with BanFakeNews dataset (Hossain et al., 2020), and text summarization with XL-Sum dataset (Hasan et al., 2021). The number of samples taken from each dataset and their statistics are provided in Tab.1.

3.3 Baselines

We evaluate closed-source models: Claude-3.5 Sonnet, and GPT-4o (Hurst et al., 2024), open-source models: Qwen-2.5 32B (Qwen et al., 2025), Llama-3 70B (Grattafiori et al., 2024), and the Bangla language models: BanglaBERT (Bhattacharjee et al., 2022) and BanglaT5 (Bhattacharjee et al., 2023).

3.4 Evaluation Metrics

For classification, we use the standard metrics: Macro-F1 (M-F1), Weighted-F1 (W-F1), and Accuracy (Acc). Similarly, for generation tasks, we use BLEU score, Brevity Penalty, and ROUGE-2-F1.

4 Experimental Results

We evaluate model robustness across three perturbation strategies on classification and generation tasks (Tables 2 and 3). Most models achieve peak

Dataset	Model														
	Claude-3.5 Sonnet			GPT-4o			Qwen-2.5 32B			Llama-3 70B			BanglaBERT		
	M-F1	W-F1	Acc	M-F1	W-F1	Acc	M-F1	W-F1	Acc	M-F1	W-F1	Acc	M-F1	W-F1	Acc
SentNob															
Bangla Text (Base)	63.90	66.30	66.19	64.37	66.53	65.83	56.78	56.79	56.57	45.07	45.16	48.18	45.80	48.16	49.50
Random Words	63.73	66.01	65.96	63.48	65.46	64.88	52.50	52.54	52.00	45.77	45.81	48.69	45.13	47.19	48.05
Δ Base	-0.17	-0.29	-0.23	-0.89	-1.07	-0.95	-4.28	-4.25	-4.57	+0.70	+0.65	+0.51	-0.67	-0.97	-1.45
Random Sentences	60.86	63.04	62.92	58.93	60.85	59.90	48.69	48.71	48.00	46.48	46.67	49.90	35.88	37.48	41.30
Δ Base	**-3.04**	**-3.26**	**-3.27**	**-5.44**	**-5.68**	**-5.93**	**-8.09**	**-8.08**	**-8.57**	+1.41	+1.51	+1.72	**-9.92**	**-10.68**	**-8.20**
Salient Words	63.82	65.88	65.78	61.99	63.96	63.18	49.94	50.01	50.00	40.18	40.21	45.23	44.42	46.63	47.48
Δ Base	-0.08	-0.42	-0.41	-2.38	-2.57	-2.65	-6.84	-6.78	-6.57	**-4.89**	**-4.95**	**-2.95**	-1.38	-1.53	-2.02
BanFakeNews															
Bangla Text (Base)	66.80	66.88	68.42	85.93	85.93	85.93	52.07	78.11	78.00	50.58	75.88	75.36	92.98	92.99	93.00
Random Words	61.00	59.20	59.71	84.91	84.91	84.94	48.54	72.80	72.45	55.10	82.64	82.47	32.71	31.79	48.60
Δ Base	-5.80	-7.68	-8.71	-1.02	-1.02	-0.99	-3.53	-5.31	-5.55	+4.52	+6.76	+7.11	**-60.27**	**-61.20**	**-44.40**
Random Sentences	57.24	58.00	61.82	85.76	85.76	85.80	51.30	76.97	76.77	53.55	80.33	80.16	50.09	49.54	57.60
Δ Base	-9.56	-8.88	-6.60	-0.17	-0.17	-0.13	-0.77	-1.14	-1.23	+2.97	+4.45	+4.80	-42.89	-43.45	-35.40
Salient Words	47.86	49.02	55.00	84.89	84.89	84.90	47.55	71.33	71.00	52.55	78.82	78.74	33.14	32.23	48.80
Δ Base	**-18.94**	**-17.86**	**-13.42**	**-1.04**	**-1.04**	**-1.03**	**-4.52**	**-6.78**	**-7.00**	**+1.97**	**+2.94**	**+3.38**	-59.84	-60.76	-44.20
BanglaHateSpeech															
Bangla Text (Base)	85.54	87.29	87.56	79.90	82.13	82.13	83.77	83.77	84.00	53.21	53.17	59.39	91.45	92.33	92.27
Random Words	85.94	87.48	87.67	78.04	80.23	80.00	80.91	80.91	81.00	51.96	51.93	58.38	87.16	88.25	88.00
Δ Base	+0.40	+0.19	+0.11	-1.86	-1.90	-2.13	-2.86	-2.86	-3.00	-1.25	-1.24	**-1.01**	-4.29	-4.08	-4.27
Random Sentences	81.95	83.81	83.78	75.52	77.74	77.33	71.44	71.44	72.00	62.58	62.62	65.66	59.97	59.11	59.87
Δ Base	**-3.59**	**-3.48**	**-3.78**	**-4.38**	**-4.39**	**-4.80**	**-12.33**	**-12.33**	**-12.00**	+9.37	+9.45	+6.27	**-31.48**	**-33.22**	**-32.40**
Salient Words	84.52	86.04	86.10	77.59	79.90	79.73	76.89	76.89	77.00	51.70	51.70	58.60	76.89	77.96	77.33
Δ Base	-1.02	-1.25	-1.46	-2.31	-2.23	-2.40	-6.88	-6.88	-7.00	**-1.51**	**-1.47**	-0.79	-14.56	-14.37	-14.94

Table 2: Macro-F1(M-F1), Weighted-F1(W-F1), Accuracy(Acc) score for the classification tasks: sentiment analysis, fake news detection, and hate speech classification on SentNob, BanFakeNews, and BanglaHateSpeech, respectively. Gray indicates base/clean text performance, and **cyan** indicates worst performance degradation.

Dataset	Model														
	Claude-3.5 Sonnet			GPT-4o			Qwen-2.5 32B			Llama-3 70B			BanglaT5		
	BLEU	BP	R2-F1	BLEU	BP	R2-F1	BLEU	BP	R2-F1	BLEU	BP	R2-F1	BLEU	BP	R2-F1
XL-Sum															
Bangla Text (Base)	0.000	1.00	0.00	0.002	0.99	0.01	0.001	0.99	0.01	0.004	0.97	0.01	0.025	0.62	0.03
Random Words	0.000	1.00	0.00	0.002	0.98	0.00	0.002	0.99	0.01	0.003	0.98	0.01	0.016	0.59	0.02
Δ Base	**-0.00**	0.00	-	-0.00	-0.01	-	+0.00	0.00	-	-0.00	+0.01	-	-0.01	-0.03	-
Random Sentences	0.000	1.00	0.00	0.002	0.98	0.01	0.001	0.99	0.01	0.003	0.98	0.01	0.017	0.59	0.03
Δ Base	+0.00	0.00	-	-0.00	-0.01	-	+0.00	0.00	-	-0.00	+0.01	-	-0.01	-0.03	-
Salient Words	0.000	1.00	0.00	0.001	0.99	0.00	0.001	0.99	0.00	0.003	0.98	0.01	0.006	0.60	0.01
Δ Base	+0.00	0.00	-	**-0.00**	0.00	-	**-0.00**	0.00	-	**-0.00**	+0.01	-	**-0.02**	-0.02	-
CSEBuetNMT															
Bangla Text (Base)	0.215	0.94	0.223	0.215	0.96	0.220	0.171	0.99	0.191	0.059	0.95	0.073	0.241	0.93	0.233
Random Words	0.191	0.94	0.201	0.184	0.97	**0.195**	0.134	0.97	**0.150**	0.051	0.92	0.065	0.180	0.91	**0.185**
Δ Base	**-0.02**	0.00	-	-0.03	+0.01	-	-0.04	-0.02	-	**-0.01**	**-0.03**	-	-0.06	-0.02	-
Random Sentences	0.199	0.89	0.212	0.109	0.95	0.125	0.052	0.96	0.066	0.065	0.94	**0.079**	0.027	0.79	0.037
Δ Base	-0.02	**-0.05**	-	**-0.11**	**-0.01**	-	**-0.12**	**-0.03**	-	+0.01	-0.01	-	**-0.21**	**-0.14**	-
Salient Words	0.262	0.96	**0.251**	0.180	0.97	0.192	0.113	0.98	0.140	0.052	0.92	0.068	0.180	0.90	0.184
Δ Base	+0.05	+0.02	-	-0.03	+0.01	-	-0.06	-0.01	-	-0.01	-0.03	-	-0.06	-0.03	-

Table 3: BLEU score, BP: Brevity Penalty, R2-F1 for the translation and summarisation tasks on the CSEBuetNMT and XL-Sum datasets, respectively. Gray indicates base/clean text performance, and **cyan** indicates worst performance degradation. The difference between the R2-F1 scores is not calculated as it doesn't hold any meaningful value.

performance on clean text and show degradation under the perturbations.

4.1 Effect of Perturbation Techniques

Random sentence and salient word perturbations induce higher performance drops than random word perturbations. For instance, Claude-3.5 Sonnet shows a 3.27% accuracy drop on SentNob under random sentence perturbation versus only 0.23% for random word perturbation. The vulnerability also varies across tasks, *e.g.* random sentence perturbation is more challenging on

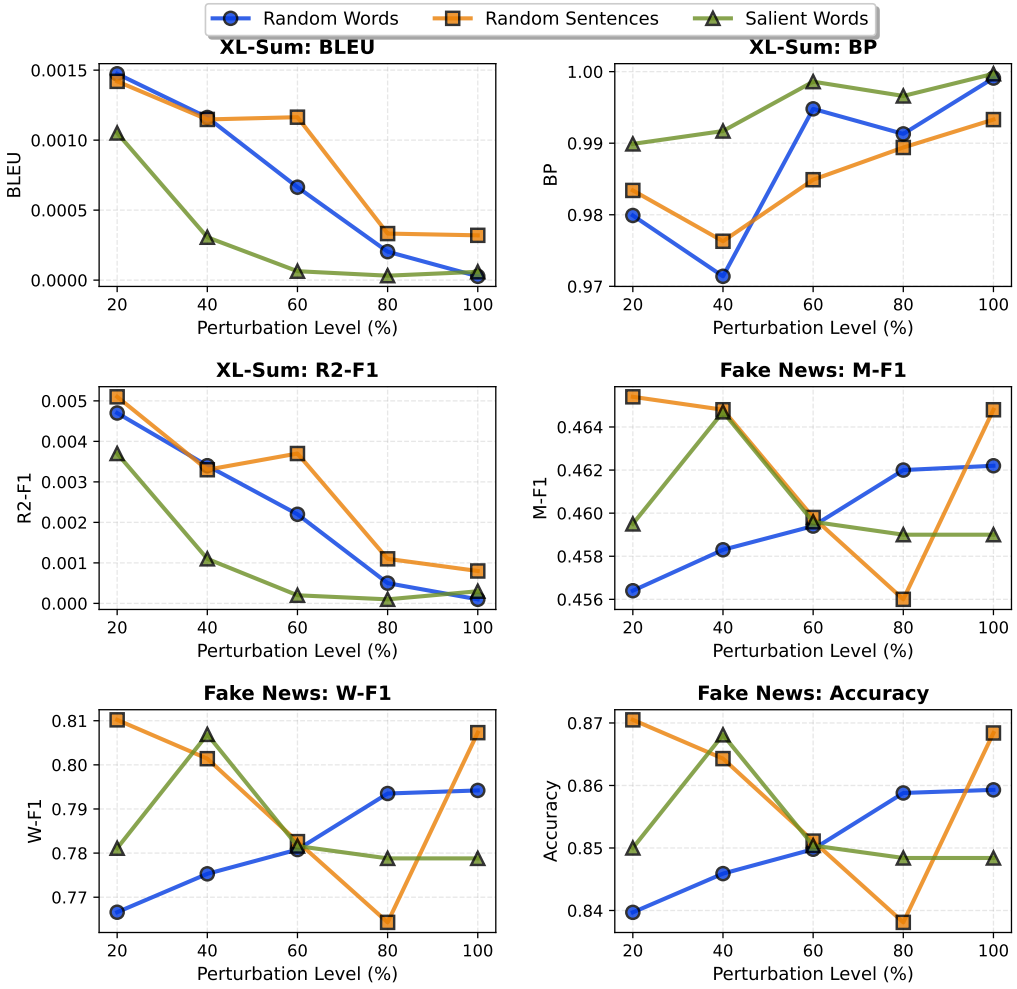

Figure 2: Impact of varying perturbation levels on the performance of the GPT-4o model in classification (Fake News) and generation (XL-Sum) tasks.

SentNob, BanglaHateSpeech, and CSEBuetNMT, while salient word perturbation is more severe on BanFakeNews and XL-Sum.

Smaller language models, like BanglaBERT and BanglaT5, show higher vulnerability, confirming their strong reliance on key lexical and semantic cues. Among LLMs, GPT-4o and Llama-3 exhibit relatively better robustness, maintaining smaller performance drops across all metrics, compared to Claude-3.5 Sonnet and Qwen-2.5. However, GPT-4o was less robust on generative tasks, *e.g.*, on the CSEBuetNMT dataset. We attribute the model-wise performance variance to the pretraining data distribution and exposure to code-mixed and script-mixed data during training.

4.2 Performance Degradation across Tasks

For classification tasks, LLMs showed relatively consistent degradation patterns: 3-8.5% on SentNob, 1-19% on BanFakeNews, and 1.5-12.5% on BanglaHateSpeech across all metrics. By contrast, BanglaBERT suffers dramatically larger drops, with F1-score degradation reaching 60% and accuracy declining by 44.5%. For generative tasks, the degradation was relatively higher for CSEBuetNMT than XL-sum, with BanglaT5 being more vulnerable than the LLMs.

4.3 Performance across Perturbation Levels

In Fig. 2, we observe a substantial decline in the summarization metrics BLEU and R2-F1, showing GPT-4o's vulnerability to increasing perturbation levels across all perturbation types. Random-word and salient-word perturbations show a consistent downward trend for the classification task. In contrast, random-sentence perturbation dips sharply at the 80% level, followed by an unexpected rebound at 100%. This suggests that the model becomes confused when only a small number of sentences are transliterated, whereas fully perturbed input

allows it to settle into a more stable interpretation.

5 Discussion

We discuss the underlying causes of performance degradation in script-mixed scenarios, promising steps for mitigation, and other future directions.

5.1 The Tokenization Bottleneck

The substantial performance degradation observed in script-mixed texts can be largely attributed to fundamental limitations in tokenization. Firstly, the choice of tokenization method varies across models and can be an inherent limitation in script-mixing. For instance, models such as BERT and T5 employ WordPiece (Schuster and Nakajima, 2012) and SentencePiece (Kudo and Richardson, 2018) tokenization, respectively, which exhibit reduced robustness compared to the Byte Pair Encoding (BPE) (Gage, 1994) used in modern LLMs. The older tokenization methods struggle to maintain consistent granularity of mixed tokens, leading to suboptimal encoding.

Secondly, the process of tokenization itself constitutes an inherent architectural bottleneck, especially for cross-script processing. In script-mixed texts, using tokenizers trained predominantly on one script, typically Latin, penalizes foreign or untrained scripts (Land and Arnett, 2025). These tokenizers frequently fragment non-English tokens into excessive subword units or map them to rare and underrepresented vocabulary entries, occasionally resorting to unknown token markers. This phenomenon reflects a deeper issue of *vocabulary bias*, where tokenizers optimized on monolingual or Latin-script-dominant corpora show systematic disadvantages when processing alternative scripts, resulting in unnecessarily long token sequences and potential information loss at the encoding stage.

5.2 BLT and Multi-script Tokenizers

Byte Latent Transformers (BLT) (Pagnoni et al., 2025) have shown great empirical robustness to input perturbations and warrant investigation in script-mixing scenarios, as their byte-level processing naturally sidesteps script tokenizing limitations. Multilingual or transliteration-aware tokenizers with joint-script vocabularies offer a potential direct solution. Such tokenizers would require balancing the training data to ensure equitable representation across scripts and prevent the replication of existing script biases.

5.3 Script Normalization

A practical and easier approach to improve script-mixing robustness can be achieved through script normalization, *i.e.*, conversion of mixed scripts to a single script that is the most dominant throughout the input text. One option is to train a dedicated normalizer model, *e.g.*, a sequence-to-sequence model similar to BanglaT5-NMT (Fahim et al., 2024), but for script conversion. Alternatively, LLMs with reasoning capabilities could be prompted to normalize scripts in the thinking process first before proceeding with the task.

5.4 Can training improve robustness?

The language models can be either continually pre-trained or fine-tuned on the script-mixed dataset. Continual pre-training on multilingual or multi-script corpora should mitigate monoscript bias and enable models to learn robust cross-script correspondences. By exposing models to diverse script combinations during pre-training, we can potentially encode invariance to script perturbations directly into the model's representations. Task-specific fine-tuning on script-mixed text could also be a viable approach, but raises difficulty in estimating the distribution of scripts, leading to plausibly higher degradation due to overfitting.

5.5 Extension to Multimodal Settings

Our perturbation pipeline can be extended to multimodal scenarios, *e.g.*, visual question answering (Antol et al., 2015; Ishmam et al., 2025) on Bangla-regional images (Barua et al., 2025), which can investigate cross-visual perturbations, such as swapping cultural elements between images, or evaluating on script-mixed questions.

6 Conclusion

Our work evaluates LLMs and Bangla LMs under transliteration-based perturbations on random words, random sentences, and salient words. Our framework provides a scalable method for augmenting existing Bangla datasets to produce their script-mixed counterparts, thereby assessing the robustness of language models. Our findings reveal that discriminative models are vulnerable to script-mixing, whereas generative models are relatively more robust. We envision that our work will open doors for future research in Bangla script-mixing.

Limitations

Our study uses only transliteration-based perturbations, which are a subset of replacement-based perturbations. Other categories of perturbations, *e.g.,* insertion, deletion, and paraphrasing, haven't been explored and could provide a holistic view of the model's robustness. Our proposed robustness enhancement strategies have not been empirically verified and could be a potential future direction.

References

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*.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Deeparghya Dutta Barua, Md Sakib Ul Rahman Sourove, Md Fahim, Fabiha Haider, Fariha Tanjim Shifat, Md Tasmim Rahman Adib, Anam Borhan Uddin, Md Farhan Ishmam, and Md Farhad Alam. 2025. Chitrojera: A regionally relevant visual question answering dataset for bangla. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 473–491. Springer.

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

Md Fahim, Fariha Tanjim Shifat, Fabiha Haider, Deeparghya Dutta Barua, MD Sakib Ul Rahman Sourove, Md Farhan Ishmam, and Md Farhad Alam Bhuiyan. 2024. BanglaTLit: A benchmark dataset for back-transliteration of Romanized Bangla. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14656–14672, Miami, Florida, USA. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, ... Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Fabiha Haider, Fariha Tanjim Shifat, Md Farhan Ishmam, Deeparghya Dutta Barua, Md Sakib Ul Rahman Sourove, Md Fahim, and Md Farhad Alam. 2024. Banth: A multi-label hate speech detection dataset for transliterated bangla. *arXiv preprint arXiv:2410.13281*.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2612–2623, Online. Association for Computational Linguistics.

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.

Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*,

pages 2744–2751, Online. Association for Computational Linguistics.

Md Zobaer Hossain, Md Ashraful Rahman, Md Saiful Islam, and Sudipta Kar. 2020. BanFakeNews: A dataset for detecting fake news in Bangla. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2862–2871, Marseille, France. European Language Resources Association.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Md Farhan Ishmam, Ishmam Tashdeed, Talukder Asir Saadat, Md Hamjajul Ashmafee, Abu Raihan Mostofa Kamal, and Md Azam Hossain. 2025. Visual robustness benchmark for visual question answering (vqa). In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6623–6633. IEEE.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Hour Kaing, Chenchen Ding, Hideki Tanaka, and Masao Utiyama. 2024. Robust neural machine translation for abugidas by glyph perturbation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 311–318, St. Julian's, Malta. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Sander Land and Catherine Arnett. 2025. Bpe stays on script: Structured encoding for robust multilingual pretokenization. *arXiv preprint arXiv:2505.24689*.

Thai Le, Jooyoung Lee, Kevin Yen, Yifan Hu, and Dongwon Lee. 2022. Perturbations in the wild: Leveraging human-written text perturbations for realistic adversarial attack and defense. *arXiv preprint arXiv:2203.10346*.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2020a. Contextualized perturbation for textual adversarial attack. *arXiv preprint arXiv:2009.07502*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul Nc, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra. 2023. Aksharantar: Open indic-language transliteration datasets and models for the next billion users. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 40–57.

Amr Mohamed, Yang Zhang, Michalis Vazirgiannis, and Guokan Shang. 2025. Lost in the mix: Evaluating llm understanding of code-switched text. *Preprint*, arXiv:2506.14012.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations. *arXiv preprint arXiv:2108.12237*.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, et al. 2025. Byte latent transformer: Patches scale better than tokens. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9238–9258.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Syed Rifat Raiyan, Md Farhan Ishmam, Abdullah Al Imran, and Mohammad Ali Moni. 2025. Frugalprompt: Reducing contextual overhead in large lan-

guage models via token attribution. *arXiv preprint arXiv:2510.16439*.

Nauros Romim, Mosahed Ahmed, Hriteshwar Talukder, and Md. Saiful Islam. 2021. Hate speech detection in the bengali language: A dataset and its baseline evaluation. In *Proceedings of International Joint Conference on Advances in Computational Intelligence*, pages 457–468, Singapore. Springer Singapore.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Rajvee Sheth, Samridhi Raj Sinha, Mahavir Patil, Himanshu Beniwal, and Mayank Singh. 2025. Beyond monolingual assumptions: A survey of code-switched nlp in the era of large language models. *Preprint*, arXiv:2510.07037.

Abhishek Srivastava, Kalika Bali, and Monojit Choudhury. 2020. Understanding script-mixing: A case study of hindi-english bilingual twitter users. In *Proceedings of the 4th Workshop on Computational Approaches to Code Switching*, pages 36–44.