BaTEClaCor: A Novel Dataset for Bangla Text Error Classification and Correction

Nabilah Tabassum Oshin*, Syed Mohaiminul Hoque*, Md Fahim, Amin Ahsan Ali, M Ashraful Amin, A K M Mahbubur Rahman
Center for Computational & Data Sciences
Independent University, Bangladesh
Dhaka-1229, Bangladesh
{1830668, 1830032, md.fahim,}@iub.edu.bd
{aminali, aminmdashraful, akmmrahman}@iub.edu.bd

Abstract

In the context of the dynamic realm of Bangla communication, online users are often prone to bending the language or making errors due to various factors. We attempt to detect, categorize, and correct those errors by employing several machine learning and deep learning models. To contribute to the preservation and authenticity of the Bangla language, we introduce a meticulously categorized organic dataset encompassing 10,000 authentic Bangla comments from a commonly used social media platform. Through rigorous comparative analysis of distinct models, our study highlights BanglaBERT's superiority in error-category classification and underscores the effectiveness of BanglaT5 for text correction. BanglaBERT achieves accuracy of 79.1% and 74.1% for binary and multiclass error-category classification while the BanglaBERT is fine-tuned and tested with our proposed dataset. Moreover, BanglaT5 achieves the best Rouge-L score (0.8459) when BanglaT5 is fine-tuned and tested with our corrected ground truths. Beyond algorithmic exploration, this endeavor represents a significant stride in enhancing the quality of digital discourse in the Bangla-speaking community, fostering linguistic precision and coherence in online interactions. The dataset and code is available at https://github.com/SyedT1/BaTEClaCor.

1 Introduction

The Bangla language is an Indo-Aryan language with deep historical roots. It is spoken by approximately 230 million people globally and is the 6th most spoken language in the world as stated by the CIA World Factbook [1]. Bangla is renowned for its intricate and unique style, holding cultural and literary significance, and reflecting a rich heritage spanning generations. However, within the contemporary world of communication, particularly on platforms like social media, the fluidity of making typographical errors often results in deviations from the language's original form. So, The complexity of the Bangla script with its 50 letters comprising 11 vowels and 39 consonants is often reflected in the digital landscape .[2]

Among the set of Bangla letters, certain complex characters contribute to the challenge of writing that results in a divergence between written and spoken communication. Phonetically similar alphabets in Bangla share the same pronunciation or phonetic utterance that allows interchangeability and consequently leads to errors within words as shown in Figure 1 (Mittra et al., 2019). For instance, Figure 1 shows the interchange of letters having similar phonetic qualities that generate error words impacting the language's authenticity and coherence (Sifat et al., 2020).

Phonetically Similar Letters	:	"ন" and "ণ" ; "শ" and "স"
Vowel Characters	:	"ি" and "ী" ;"ৌ" and "ৈ"
Consonant Clusters	:	"ঞ্জ" and "জ্ঞ" ; "ম্ভ" and "ম্ব"
Informal Style	:	"খাইতেসি" ; "করতেসিলাম"

Figure 1: Examples of Different types of errors

In the realm of online platforms, such as YouTube and other social media networks, users frequently embrace an informal variant of the Bangla language that is characterized by regional speech patterns and influenced by local dialects or colloquial expressions typical to the residents of the area. This informal variant derived from the original standard Bangla tends to deviate from its roots and originality. This shift can be attributed to the fast-paced and dynamic nature of online communication where brevity, quickness, and informal

* These authors contributed equally to this work.

[1] https://www.cia.gov/the-world-factbook/countries/world/

[2] https://en.wikipedia.org/wiki/Bengali_alphabet

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 124–135
December 7, 2023

expression often take precedence over traditional linguistic norms as an example shown in Figure 1.

Textual error detection and correction of the Bangla language hold significant importance as corrected text preserves language integrity, promotes literacy, and conveys professionalism. Online interactions further underscore the necessity of Bangla text correction as it enables clear communication on a global scale, enhances brand reputation, facilitates cross-cultural communication, and reduces the chances of misinterpretation. Notably, there have been datasets used for similar purposes, predominantly consisting of samples collected from Bangla newspapers, blogs, or synthetically generated.(Mridha et al., 2019) (Sifat et al., 2020). However, they may not fully represent the day-to-day informal and formal interactions of Bangla language speakers on various online platforms where several types of errors can be more prevalent. To address this gap in the existing resources, we introduce a novel dataset for Bangla text error correction named **BaTEClaCor**: A Novel Dataset for **Ba**ngla **T**ext **E**rror **Cla**ssification and **Cor**rection. The dataset is licensed under CC -BY-NC 4.0 (Creative Commons Attribution)

Through a comprehensive approach, this research aligns itself with the larger goal of fostering a digitally literate and linguistically precise digital space for the Bangla community. Our contributions are:

- Introduction of an expansive and authentic dataset comprising 10k of diverse Bangla comments from YouTube videos. The dataset can enhance the generation capability of transformer-based models by providing valuable insights into the informal and regionally influenced Bangla language.
- Performance analysis of several advanced machine learning and deep learning models including BanglaBERT, LSTM, and XLM-RoBERTa to detect errors within Bangla YouTube comments and classify them based on specific error categories while the models are fine-tuned and tested with the proposed dataset.
- Analyzing the performance of BanglaT5 to correct different categories of textual errors including phonetic and grammatical errors while fine-tuning and testing with our proposed dataset.

These contributions enhance the quality of linguistic interactions online and pave the way for a more precise and digitally literate environment for Bangla speakers, fostering meaningful communication, and understanding in the digital realm.

2 Related Work

Numerous endeavors have been undertaken to enhance Bangla text correction despite its status as a low-resource language. Notably, a Bangla spell-checking technique was proposed and tested on a dictionary consisting of pairs of 50,000 correct and incorrect Bangla words. N-gram models were generated for each candidate word. To identify non-word errors, a comprehensive Bangla word dictionary of around 600,000 words was compiled from various online repositories, newspapers, social networking sites, and Bangla blogs (Mittra et al., 2019). The study primarily addresses word-level errors and may lack in encompassing the full spectrum of errors, including contextual and informal errors

H.A.Z. Sameen presented a novel approach for Bangla grammatical error detection using a T5 Transformer model. The training set comprised 9385 sentence pairs, while the testing set included 5,000 test sentences (Shahgir and Sayeed, 2023). It's mentionable that the incorrect sentences in the paired samples were not explicitly categorized to identify specific error types, and instead, errors were indicated using a particular symbol without detailed error categorization.

Chowdhury Rafeed introduced BSpell, a CNN-blended BERT-based Bangla spell checker (Rahman et al., 2022). The synthetic dataset of The Prothom-Alo 2017 online newspaper was used for training, Additionally, 6,300 errorful sentences from Nayadiganta online newspaper were annotated for testing. It's essential to note that the training data's synthetic nature and the usage of newspaper text may not effectively capture the nuances of informal online interactions.

Another method for synthetic error dataset generation was presented using a few sets of popular newspapers mimicking Bangla writing patterns. The study employed a Bangla corpus consisting of 6.5 million sentences. From this corpus, 8,637 frequently occurring words were selected for analysis (Sifat et al., 2020). The study's outcomes revealed the stochastic nature of error generation.

Although these studies collectively contribute

significantly to the advancement of Bangla text correction techniques encompassing various methodologies and datasets, we aimed to address their limitations by constructing a distinct dataset that encompasses the specific error types and reflects the real-world informality prevalent in online communication.

3 Introducing A New Dataset

3.1 Motivation Behind Creation of a New Dataset:

As discussed earlier, existing Bangla datasets used for textual error correction mostly featured samples derived from newspaper articles, blogs and bangla repositories or were synthetically generated. Such sources often portray a formal, official use of the language, which may deviate significantly from its common application in online interactions. Recognizing the need to capture the intricacies of language as it is typically used, we turned to social platforms such as YouTube and Facebook. The driving force behind crafting this new dataset arises from the vital significance of linguistic precision, coupled with the evolving digital environment that defines modern communication particularly among Bangla-speaking internet users of Bangladesh.

3.2 Source of Data Samples:

We selected YouTube as our primary source of sample collection due to its immense popularity in Bangladesh, boasting approximately 34.50 million Bangladeshi users, according to Google's advertising resource. This platform serves as a microcosm of the country's linguistic diversity attracting users from various backgrounds, different levels of literacy, and typing patterns.

To compile this unique dataset, we performed web scraping utilizing YouTube's API on randomly listed videos as shown in Figure 3 and 2 having more than 500k views within August 2023. The random selection minimizes potential bias and ensures a variety of linguistic expressions and errors. Around 60 comments per video were taken to collect ample data for analysis from each video and to provide a balanced dataset size. Selecting comments with three or more words ensures that the dataset contains substantial content for meaningful analysis, and it also minimizes unnecessary padding. This approach optimizes dataset efficiency and is well-suited for machine learning and deep learning models that require fixed-length input sequences.

Algorithm 1 Pseudocode for Comment Scrapping

```
Input: API_KEY = Youtube's API ,
       video_list = ["video_id1", "video_id2", ...]
Output: comments
Initialize comments [ ]
Initialize existing_comments { }
For each video_id in video_list do:
    Retrieve video details from(video_id, API_KEY)
    Extract video title from video details
    Initialize comments_counter = 0
    WHILE comments_counter < 60:
        Retrieve comments
        Preprocess comments
            For each comment do:
                IF comment (Is bengali = True) &&
                (Length_of_comment >= 3 )      &&
                comment NOT IN existing_comments{ }:
                    Append comment TO comments [ ]
                    Add comment TO existing_comments{ }
                    comments_counter += 1
```

Figure 2: Pseudocode for data collection

3.3 Labeling and Annotation:

The labeling and annotations in this dataset were carried out by three of the authors through a careful manual process, ensuring a high level of precision and reliability. The team extensively referred to linguistic references, particularly the authoritative work **Bangla Byakaran O Nirmiti** by Dr. Solaiman Kabir, and the **Bangla Ovidhan** dictionary. These resources played a vital role in guaranteeing the accuracy and linguistic correctness of the dataset, making it a valuable asset for the Bangla language community. A detailed overview of the labeling and annotation procedures is presented in Figure 3.

3.4 Structure and Features of Dataset:

BaTEClaCor dataset aims to serve as a valuable resource for researchers and practitioners seeking to enhance the accuracy and performance of Bangla typing error detection and correction models.

The dataset comprises 10,000 comments, meticulously filtered to include only those written in Bangla letters. Comments containing irrelevant emojis and symbols were discarded, ensuring the dataset's quality and utility. In Table 1, the dataset's composition reflects its comprehensive nature. Of the 10,000 entries, 4224 pertain to incorrect com-

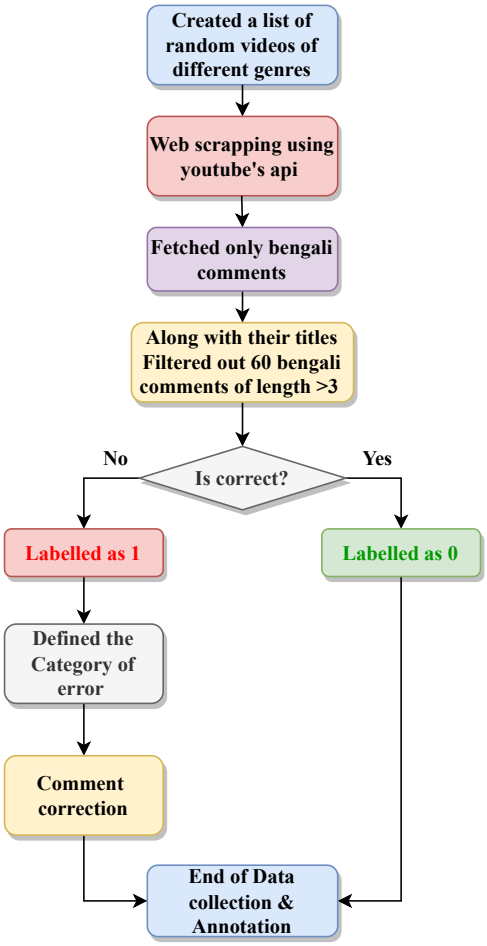


Figure 3: Flowchart of data collection and annotation

ments, while the remaining 5,776 constitute accurate comments. These comments span a diverse array of video genres, including News, Entertainment, Politics, Sports, and Miscellaneous as shown in Table 2.

Label	No. of Comments
0	5776
1	4224

Table 1: Distribution of Labels in the Dataset

Table 2 shows that the selection of video categories for this dataset is carefully orchestrated to encompass a broad spectrum of topics that hold immense significance within the Bangla context. While News, Entertainment, Politics, and Sports constitute the bedrock of societal discourse, allowing individuals to voice their opinions and ideas through comments, the Miscellaneous category transcends conventional boundaries, embracing topics such as Lifestyle, Philosophy, Nature, etc to reflect the diverse interests and passions of Bangladeshi people.

Genre	No. of Comments
Entertainment	3450
News	2009
Miscellaneous	1932
Politics	1885
Sports	771

Table 2: Distribution of Comments by Genre

In Table 3, errors within the dataset are categorized into four distinct and most prevalent types, reflecting the intricate nature of the Bangla script and its potential pitfalls.

- **Spelling**: Spelling being the most commonly occurring category of errors, encompass instances of incorrect spellings.
- **Grammatical**: Grammatical errors denote mistakes related to the structural and syntactical aspects of the Bangla language.
- **Code-Switching** :Code-switching, often referred to as the mixing of English and Bangla within a single comment, a phenomenon known as Banglish. These instances may not constitute conventional text errors in terms of comprehension or meaning. However, their categorization aims to maintain linguistic authenticity by preserving the true essence of the Bangla language, ensuring adherence to standard and widely accepted linguistic norms.
- **Multiple Errors**: Multiple errors encompass comments featuring a combination of error types, such as misspellings alongside code switching or grammatical mistakes intertwined with spelling errors.

Error Category	No. of Comments
Spelling	2502
Code Switching	786
Grammatical	638
Multiple Errors	345

Table 3: Distribution of Comments by Error Category

Each record in the dataset features vital information such as video title, genre, original comment, label, and error category. We can see a sample data in Table 4 which contains a comment with an error under a sports video, more specifically a spelling

error highlighted in red. The comment is corrected precisely and marked where the correction was made in green.

Comment	মাহমুদউল্লাহ রিয়াদ ভাই কে দলে দেকতে চাই
Video Title	সেরা ম্যাচ!!! Bangladesh vs Sri Lanka Sports Talkies
Genre	Sports
Label	Error
Error category	Spelling
Correct Comment	মাহমুদউল্লাহ রিয়াদ ভাই কে দলে দেখতে চাই

Table 4: Sample Data

This novel dataset presents an invaluable contribution to the realm of Bangla NLP. By amalgamating accurate and erroneous comments from diverse genres, our dataset provides a nuanced view of real-world language usage and common typing errors. It serves as a resource that can facilitate the development and fine-tuning of typing error detection models, ultimately improving the linguistic quality and effectiveness of online communication in Bangla.

4 Baseline

4.1 Classification Models

4.1.1 Using ML Models

Initially, TF-IDF segments the text into words. Then TF-IDF scores for each word are calculated and utilized to construct a feature vector.

$$TF - IDF(w, d) = \text{TF}(w, d) \cdot \text{IDF}(w)$$

Here, *TF(w, d)* represents the term frequency of the word w in the dataset d. *IDF(w)* is the inverse document frequency of the word w. The feature vectors x are calculated from the TF-IDF score and then used to train the classifier models. The SVM model makes predictions by finding the class of the hyperplane that is closest to the sample data(Dadgar et al., 2016). Random forest model learns to predict the class of a sample by finding the class with the highest probability utilizing the class label and feature vector(Sjarif et al., 2019). For an input feature vector, the XGBoost model predicts the text's class by selecting the class with the highest predicted value(Qi, 2020).

4.1.2 Using DL Models

LSTM: LSTM processes an input sentence $S = x_1, x_2, ...x_n$ from the dataset of x_i words and passes to an embedding layer to get embedded representations $E = e_1, e_2, ...n$. These are taken as input by LSTM model to find hidden representations $H = h_1, h_2, ...n$. The last layer's hidden representations of LSTM model are passed to a linear layer to perform classification (Hochreiter and Schmidhuber, 1997).

LSTM with Attention: The resulting hidden states h_i from an input sequence processed by LSTM are then used in the Attention mechanism to calculate Attention score α_i.

$$\alpha_i = Softmax(W_{hi} + b)$$

$$c_i = \sum_{j=1}^{n} \alpha_{ij} \hat{h}_j$$

Here, the context vector for each sentence is calculated by taking a weighted sum of the hidden states ($\hat{h}_j$) based on the attention weights (α_{ij}) for each time step (Vaswani et al., 2017). The context vectors c are then used for classification.

CNN-LSTM: In the CNN with LSTM architecture, the input sequence x is first processed by a convolutional neural network (CNN), resulting in feature maps f (Kim, 2014). These feature maps are then used by the LSTM model to calculate hidden states h to be passed into a linear layer for classification.

4.1.3 Using Transformer Models

BanglaBERT and XLM-RoBERTa: These deep-learning transformer models are Pre-trained and further fine-tuned. To obtain a fixed-size representation for an input sentence, we typically use the special [CLS] token representation h_{CLS}.

$$P = Pooling(H)$$

Global pooling is applied to obtain a fixed-size representation P. The final hidden states H from the transformer layers, capture the essence of the input text. This pooled representation is used for classification (Bhattacharjee et al., 2022) (**?**).

4.2 Error Corrector Model

Let X represent the set of input sequences (comments) and Y represent the set of target sequences (corrected forms of the comments). For each input

sequence $x_i \in X$, which is a sequence of tokens, the sequence-to-sequence model f, specifically the T5 base model fine-tuned for error correction, generates an output sequence y_i'. This output sequence y_i' corresponds to the corrected version of the input comment x_i. Mathematically, the task can be defined as follows:

$$y_i' = f(x_i)$$

The primary objective of this task is to train the model f in such a way that it minimizes a suitable loss function (e.g., cross-entropy loss) that quantifies the dissimilarity between the predicted sequence y_i' and the actual target sequence y_i. The training dataset with input comments and their corresponding corrected forms, allows the model to learn the mapping from erroneous comments to their accurate versions(Bhattacharjee et al., 2022).

5 Experimental Design

5.1 Preprocessing & Settings

Our initial focus was on text preprocessing to ensure data quality. For binary classification using machine learning models, we explored text encoding techniques paired with specific classifiers. Such as TF-IDF with Random Forest and with XGBoost with 6000 max features and 100 decision trees. For deep learning models, we investigated LSTM networks on a batch size of 100 with varying configurations and optimization using the Adam optimizer. The LSTM model featured an embedding layer of 6 dimensions. Additionally, we explored LSTM with Attention, utilizing an embedding size of 128. LSTM with CNN with an embedding dimension of 300 including a convolutional layer with 128 filters and a kernel size of 5. In addition, we also explored transformer-based models like XLM-RoBERTa and BanglaBert, employing tokenization with a maximum sequence length of 128. These models were trained with batch sizes of 16.

For multiclass classification, The ML models were applied similarly to binary classification. DL model LSTM was incorporated with an embedding layer with dimensions of 50000x100, an input length of 3000, and an LSTM layer of 100 units operated on a batch size of 64. We also explored LSTM with attention and with CNN employing an LSTM layer of 64 units on a batch size of 16. LSTM with CNN included a convolutional layer with 128 filters and a kernel size of 5. In parallel, the transformer models BanglaBert and XLM-Roberta employed tokenization with a maximum sequence length of 128. Both the models utilized the Adam optimizer with a learning rate of 1×10^{-5} and operated over a batch size of 16.

For error correction, we used two pre-trained models named BanglaT5 and BanglaT5-small respectively, and fine-tuned them in our dataset. The batch size for training and evaluation was set to 16. The learning rate used for training the model was set to 2e-5. The weight-decay parameter helps prevent overfitting which is set to 0.01. We also used fp16 which speeds up training and reduces memory usage while maintaining training stability.

5.2 Evaluation Metrics

To compare the model performance on the predictions, we use the following performance-based metrics:

- **Accuracy**: This metric measures the proportion of correctly classified samples over the total number of samples.
- **Macro Precision**: This metric measures the average of the calculation of precision of each class. It treats all the classes equally regardless of their size or prevalence in the dataset.
- **Macro Recall**: This metric calculates the average of the calculation of the recall for each class.
- **Rouge-1**: This metric calculates the number of overlapping unigrams (single words or tokens) between the generated text and the reference text.
- **Rouge-2**: Rouge-2 calculates the number of overlapping bigrams (two-word sequences) between the generated text and the reference text. Similar to Rouge-1, its score ranges from 0 to 1.
- **Rouge-L**: This metric calculates the length of the longest common subsequence between the generated text and the reference text.

The chosen metrics were selected for their suitability in evaluating text error detection and correction tasks. Accuracy is a fundamental metric for classification tasks, while macro precision and macro recall account for class imbalances. On the other hand, Rouge-1, Rouge-2, and Rouge-L are widely used in assessing the quality of the generated text,

and their usage here reflects the nature of the correction task, aligning closely with real-world applications.

6 Result and Analysis

6.1 Binary Classification

6.1.1 Machine Learning Models:

Table 5 shows that among the ML models we applied, the TF-IDF with XGBoost model demonstrated a slight advantage in both accuracy and macro precision compared to the TF-IDF with SVM model due to the capability to handle non-linear relationships effectively through its ensemble learning approach. In contrast, the TF-IDF approach used by both models tends to exhibit limitations in capturing complex linguistic patterns present in Bangla text. On the other hand, the TF-IDF with Random Forest model displayed slightly inferior results, suggesting its struggle with capturing the intricacies of textual data.

6.1.2 Deep Learning Models:

In DL models, LSTM showed considerable results as shown in as shown in Table 5. When compared to the LSTM model, LSTM with Attention demonstrated a 4% higher accuracy, 2% higher macro precision, and 1% higher macro recall highlighting its ability to capture more complex dependencies in the text. Additionally, the hybrid model, CNN-LSTM, outperformed the LSTM model by almost 6% in accuracy, 3% in macro precision, and 4% in macro recall, showcasing its prowess in identifying patterns in sequences of text.

Originally, some of the models were not subjected to fine-tuning. Subsequently, these models were refined based on specific parameters, leading to enhanced results.

6.1.3 Transformer Models:

BanglaBert demonstrated remarkable accuracy and macro precision, outshining the LSTM model by a significant margin as shown in Table 5. In comparison, XLM - Roberta Base, a versatile multilingual Transformer, delivered competitive results, albeit falling slightly short of BanglaBert's performance. These Transformer models capitalized on their advanced architecture and pre-trained representations to effectively handle the intricacies of Bangla text.

6.2 Multiclass Classification

6.2.1 Machine Learning Models:

For the multiclass classification for error categories, the TF-IDF with SVM model showcased moderate performance with TF-IDF as the feature extraction method and Support Vector Machines (SVM) as the classifier. The TF-IDF with Random Forest model displayed results on par with the SVM model, both sharing the TF-IDF feature extraction approach. Conversely, the TF-IDF with XGBoost model showed a marginal improvement, performing around 1.6% better than the SVM model.

6.2.2 Deep Learning Models:

The DL models displayed varying degrees of proficiency in multiclass classification. Remarkably, LSTM with Attention emerged as the top performer, showcasing a significant 5.3% higher accuracy than the LSTM model. This notable lead can be attributed to the enhanced sequence modeling capabilities of LSTM with Attention. Additionally, CNN + LSTM delivered promising results, outperforming the LSTM model by approximately 3.7% in accuracy. This outcome underscores CNN + LSTM's ability to detect intricate patterns within text sequences, making it a valuable asset for multiclass classification tasks.

6.2.3 Transformer Models:

Once again, BanglaBert emerged as the best performer, showcasing a notable 9.5% higher accuracy compared to the LSTM model. This substantial lead can be attributed to BanglaBert's deep learning architecture and its prowess in capturing complex linguistic patterns and semantic meanings, which are crucial for multiclass classification tasks. While XLM - Roberta Base followed closely, performing around 3.7% better than the LSTM model, it still trailed BanglaBert in accuracy Table 5.

During the sample collection process, we encountered a relatively lower number of instances for the grammatical and multiple error categories compared to code-switching and spelling. As a result, we observed that the model is comparatively less proficient in sentences where these categories of errors are present. From our extensive evaluation, we observed that DL models outperformed the ML models, underscoring their ability to capture essential linguistic nuances and long-term dependencies within the text, crucial for classification tasks. Transformer models, including BanglaBert and XLM-RoBERTa Base, further exemplify the

Classification Types	Model Name	Performance Metrics		
		Accuracy	Macro Precision	Macro Recall
Binary Classification	TF-IDF + SVM	62.8	62.4	58.3
	TF-IDF + RandomForest	62.7	61.4	59.3
	TF-IDF + XGBoost	63.7	66.8	58.0
	LSTM	64.0	65.0	64.0
	LSTM + Attention	68.0	67.0	65.0
	CNN + LSTM	69.7	68.0	68.0
	XLM-Roberta	74.2	73.6	73.8
	BanglaBERT	79.1	79.7	77.1
Multiclass Classification	TF-IDF + SVM	60.8	59.5	30.1
	TF-IDF + RandomForest	60.3	53.1	32.9
	TF-IDF + XGBoost	61.1	60.5	29.6
	LSTM	62.7	55.2	46.7
	LSTM + Attention	59.4	44.0	39.2
	CNN + LSTM	55.4	41.2	40.0
	XLM-Roberta	69.4	37.6	43.2
	BanglaBERT	74.1	70.7	52.4

Table 5: Performance of different models in Error Classification

power of deep learning in enhancing classification accuracy.

6.3 Corrector Model

The performance of the error corrector model is reported in Table 6 where BanglaT5 and BanglaT5-Small were experimented. Both models perform better in the dataset. BanglaT5 gives 1% improvement rather than BanglaT5 small.

Best Predicted

Comment : কাউকে কষ্ট দিয়ে কেউ কখনো সুখী হতে পাড়ে না

Predicted : কাউকে কষ্ট দিয়ে কেউ কখনো সুখী হতে পারে না

Ground Truth : কাউকে কষ্ট দিয়ে কেউ কখনো সুখী হতে পারে না

Worst Predicted

Comment : জাতীয় পাখি দোয়েল কিন্তু দোয়েল নেই চিরাকানায়

Predicted : জাতীয় পাখি দোয়েল কিন্তু দোয়েল নেই চিরাকানায়

Ground Truth : জাতীয় পাখি দোয়েল কিন্তু দোয়েল নেই চিড়িয়াখানায়

Figure 4: Best predicted and worst predicted input

We obtained better scores in ROUGE-1 and ROUGE-L because the dataset we created consisted of single-word errors mostly. Due to this reason, the best 5 predicted sentences of the dataset have a ROUGE-L score of 1.0 and the worst 5 have ROUGE-L scores between the range of 0.2667 and 0.7500. In Figure 4, we can see how sentences with multiple errors performed poorly. More insights on the ROUGE-L scores can be found in Appendix A.3.

In training the BanglaT5 (Bhattacharjee et al., 2022) model, it took 2.25 minutes per epoch. The average inference time on the test dataset was about 0.2614 seconds. We used another pretrained model BanglaT5 Small for training on the dataset which took almost 0.79 minutes per epoch. The average inference time was about 0.1281 seconds which is almost half of the inference time of BanglaT5 model.

Model	Rouge-1	Rouge-2	Rouge-L
BanglaT5	0.8461	0.4430	0.8459
BanglaT5 Small	0.8343	0.4246	0.8344

Table 6: ROUGE Scores(F1)

When comparing the two models numerically, BanglaT5 consistently outperforms BanglaT5-small in all three Rouge metrics: Rouge1, Rouge2, and RougeL. However, the differences between the

models are relatively small, with BanglaT5 having only a slight edge in terms of these specific evaluation scores.

7 Conclusion

In this study, we embarked on a comprehensive journey to address the critical challenge of Bangla text correction leveraging both traditional machine learning and deep learning techniques along with Transformer models. A pivotal milestone was the creation of a novel dataset from Youtube comments that was meticulously curated and annotated. The dataset serves as the cornerstone for our investigation.

We conducted a rigorous evaluation of machine learning models and deep learning models including transformer models for binary and multiclass error-category classification. The standout performance of BanglaBert showcased its ability to navigate complex linguistic semantics. Additionally, the experimental results underscore the potential of BanglaT5 for improving the accuracy and robustness of correction systems in Bangla user-generated text. BanglaBERT achieves accuracy of 79.1% and 74.1% for binary and multiclass error classification while the BanglaBERT is fine-tuned and tested with our proposed dataset. Moreover, BanglaT5 achieves the best Rouge-L score (0.8459) while BanglaT5 is fine-tuned and tested with our corrected ground truths. Our findings underscored the transformative potential of deep learning models and emphasized the importance of dataset curation. The proposed dataset stands as a unique resource set apart from its predecessors, offering a representation of language use in online settings that are more aligned with the language patterns of Bangla speakers in digital communication.

Limitations

The primary constraint of this study lies in the size of the dataset. While being valuable for Bangla textual error detection and correction tasks, it remains insufficient for broader applications such as classification, complex NLP tasks, and large-scale error correction. Additionally, it would have been advantageous to have more incorrect samples compared to correct ones for enhanced model training. We have excluded comments with an excessive number of emojis, potentially leading to the loss of crucial context in informal communication. we will consider incorporating emojis and special symbols in our future data collection endeavor. Moreover, The dataset's focus remains rooted in the specific linguistic context of Bangladesh. It may not comprehensively represent the linguistic patterns and variations found in other regions where Bangla is spoken.

Future Plan

We look ahead to exploring advanced NLP techniques with an expanded dataset containing more errorful samples to enhance correction systems in Bangla user-generated text. It may have the potential to address a previously underrepresented aspect of Bangla language correction, filling a gap in traditional language model training, especially for generative tasks. Our future plans also involve broadening the scope to accommodate variations in Bangla language as spoken in different regions. We also would like to incorporate the Elo rating system in our experiments.

Ethical Considerations

BaTEClaCor dataset is licensed under CC -BY-NC 4.0 (Creative Commons Attribution). It is important to note that the comments are solely collected for research purposes, in compliance with YouTube's Terms of Service. The anonymity of the commenters was rigorously maintained, with no personal information related to the commenters being captured or stored.

Acknowledgements

This project has been jointly sponsored by Independent University, Bangladesh and the ICT Division of the Bangladesh Government.

References

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.

Seyyed Mohammad Hossein Dadgar, Mohammad Shirzad Araghi, and Morteza Mastery Farahani. 2016. A novel text mining approach based on tf-idf and support vector machine for news classification. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, pages 112–116. IEEE.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Tanni Mittra, Sadia Nowrin, Linta Islam, and Deepak Chandra Roy. 2019. A bangla spell checking technique to facilitate error correction in text entry environment. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–6. IEEE.

MF Mridha, Md Abdul Hamid, Md Mashod Rana, Md Eyaseen Arafat Khan, Md Masud Ahmed, and Mohammad Tipu Sultan. 2019. Semantic error detection and correction in bangla sentence. In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pages 184–189. IEEE.

Zhang Qi. 2020. The text classification of theft crime based on tf-idf and xgboost model. In *2020 IEEE International conference on artificial intelligence and computer applications (ICAICA)*, pages 1241–1246. IEEE.

Chowdhury Rafeed Rahman, MD Rahman, Samiha Zakir, Mohammad Rafsan, and Mohammed Eunus Ali. 2022. Bspell: A cnn-blended bert based bengali spell checker. *arXiv preprint arXiv:2208.09709*.

HAZ Shahgir and Khondker Salman Sayeed. 2023. Bangla grammatical error detection using t5 transformer model. *arXiv preprint arXiv:2303.10612*.

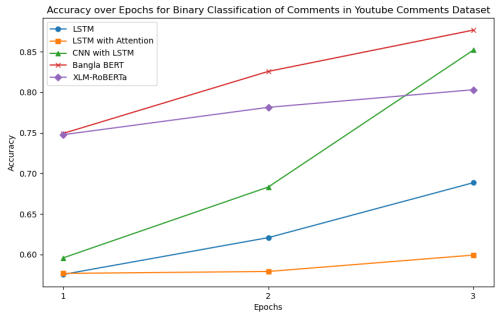
Md Habibur Rahman Sifat, Chowdhury Rafeed Rahman, Mohammad Rafsan, and Hasibur Rahman. 2020. Synthetic error dataset generation mimicking bengali writing pattern. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 1363–1366. IEEE.

Nilam Nur Amir Sjarif, Nurulhuda Firdaus Mohd Azmi, Suriayati Chuprat, Haslina Md Sarkan, Yazriwati Yahya, and Suriani Mohd Sam. 2019. Sms spam message detection using term frequency-inverse document frequency and random forest algorithm. *Procedia Computer Science*, 161:509–515.

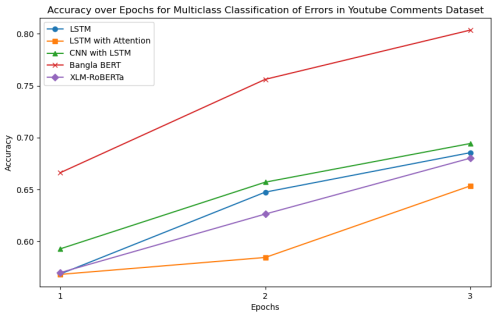
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

A Accuracy & Loss Plots

A.1 Accuracy Plots



Binary Classification Accuracy Plot



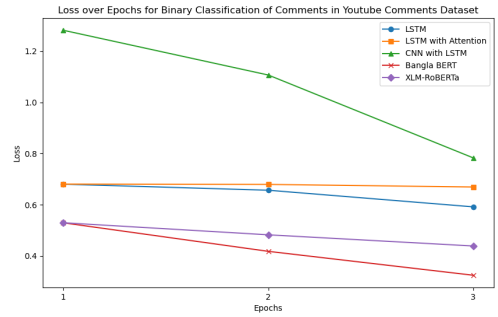
Multiclass Classification Accuracy Plot

Figure 5: Accuracy Plots of Different Models

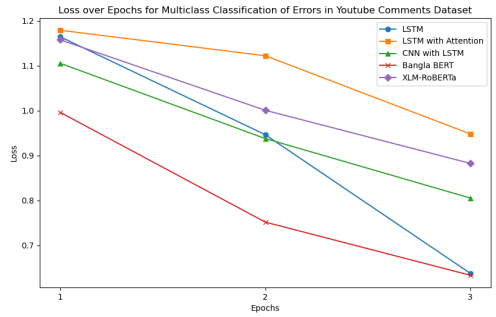
The accuracy plots show, in case of binary classification of comments, we see that BanglaBERT outperformed all the other models with about 88 percent accuracy towards the 3rd epoch. We can also see that,CNN with LSTM had a steady increase of accuracy per epoch i.e from 59.6% in the 1st epoch to 85.2% by the end of 3rd epoch . We see that LSTM with attention had less improvement over the epochs.

In case of multiclass classification of errors, we see that BanglaBERT has better accuracy than other models which is almost 80.35% . CNN with LSTM also gets around 70% accuracy by the end of 3rd epoch. Both LSTM and LSTM with attention's accuracy has minimal improvement over the epochs.

A.2 Loss Plots



Binary Classification Loss Plot



Multiclass Classification Loss Plot

Figure 6: Loss Plots of Different Models

The loss plots show, during the training session of models for binary classification of comments, CNN with LSTM's loss decreasing significantly after each epoch. The change in loss was almost similar for LSTM with Attention and LSTM. A minor reduction in loss was observed for BanglaBERT by the end of 3rd epoch.

Furthermore, during the training session of models for multiclass classification of comments, we see BanglaBERT's loss decreasing significantly in every epoc upto 63 percent after the 3rd epoch. We also see the same for LSTM where there is a significant decrease of loss. The most minimal loss was observed here for LSTM with attention.

A.3 Corrector Model Prediction Analysis

A.3.1 Top 5 Best Predicted Outputs

From top 5 best predicted outputs, we see that common single-word errors were predicated properly which is indicated by the ROUGE-L score of 1.0.

Comment	Predicted	Ground Truth	Rouge-L
ভাই **কোই** পাওয়া যাবে দাম কত টাকা	ভাই **কই** পাওয়া যাবে দাম কত টাকা	ভাই **কই** পাওয়া যাবে দাম কত টাকা	1.0
একদম **আমাকে** সাথে যা যা হয় হুবহু মিলে যায়	একদম **আমার** সাথে যা যা হয় হুবহু মিলে যায়	একদম **আমার** সাথে যা যা হয় হুবহু মিলে যায়	1.0
বাংলাদেশের মানুস জিয়া কে ভালবাসে	বাংলাদেশের মানুষ জিয়া কে ভালবাসে	বাংলাদেশের মানুষ জিয়া কে ভালবাসে	1.0
খুব ভালো লাগলো থ্যাংক ইউ আপনাকে	খুব ভালো লাগলো ধন্যবাদ আপনাকে	খুব ভালো লাগলো ধন্যবাদ আপনাকে	1.0
একটা **মানুস** আরেকটা মানুষকে কেমনে ফেলে চলে যায় এটা তো আমার জানা নাই	একটা **মানুষ** আরেকটা মানুষকে কেমনে ফেলে চলে যায় এটা তো আমার জানা নাই	একটা **মানুষ** আরেকটা মানুষকে কেমনে ফেলে চলে যায় এটা তো আমার জানা নাই	1.0

Figure 7: The Top 5 Best Predicted Outputs

A.3.2 Top 5 Worst Predicted Outputs

Comment	Predicted	Ground Truth	Rouge-L
অসাধারণ একটি ডকুমেন্টারি চ্যানেল	অসাধারণ একটি ডকুমেন্টারি চ্যানেল	অসাধারণ একটি তথ্যচিত্র চ্যানেল	0.7419
ডায়লোগ গুলো বেশি বেশি হইছে	ডায়লোগ গুলো বেশি বেশি হইছে	সংলাপ গুলো বেশি বেশি হইছে	0.7407
ভাইরে ভাই কি এক্সপ্রেসন	ভাইরে ভাই কি এক্সপ্রেসন	ভাইরে ভাই কি অভিব্যাক্তি	0.6667
এমন ভিডিও অরো চাই প্লিজ	এমন ভিডিও অরো চাই প্লিজ	এমন ভিডিও আরো চাই প্লিজ	0.5821
লিজেন্ড কে হারিয়ে ফেরেছি আমরা	লিজেন্ড কে হারিয়ে ফিরেছি আমরা	কিংবদন্তীকে কে হারিয়ে ফেলেছি আমরা	0.5240

Figure 8: The Top 5 Worst Predicted Outputs

There were certain words which were inadequately present in the dataset. Due to which the prediction scores tend to fall for such samples containing those words. We can see that the range of ROUGE-L score lies between 0.5240 and 0.7419 for the worst 5 predicted outputs.