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

Autumn 2023

Prepared by:

Syed Mohaiminul Haque
ID: 1830032

Nabilah Tabassum Oshin
ID: 1830668

Farhan Noor Dehan
ID: 1920269

Department of Computer Science and Engineering
Independent University, Bangladesh

Supervised by:

AKM Mahbubur Rahman
Associate Professor
Department of Computer Science and Engineering
Independent University, Bangladesh

Attestation

We are aware of the fact that plagiarism is strictly prohibited, and it conflicts with our university's rules and regulations. We guarantee the authenticity of our work. We have used some copyrighted materials and models of others in our project, which have been properly cited following international standards and proper guidelines.

Author Name:

. . . . . .

Signature:

. . . . . .

Author Name:

. . . . . .

Signature:

. . . . . .

Author Name:

. . . . . .

Signature:

. . . . . .

Evaluation Committee

Supervisor

Name: _____ Signature: _____

Examiner 1

Name: _____ Signature: _____

Examiner 2

Name: _____ Signature: _____

Examiner 3

Name: _____ Signature: _____

Acknowledgement

We would like to express our sincere gratitude to our respected supervisor, AKM Mahbubur Rahman, Independent University, Bangladesh. His dedicated guidance, insightful suggestions, remarkable patience, and generous investment of his valuable time played a pivotal role in the successful completion of our project. Our professor's expertise and mentorship greatly enhanced the quality and depth of our work, and we are truly appreciative of his unwavering support throughout our academic journey.

Additionally, our heartfelt thanks go to the CCDS Lab (Center for Computational and Data Sciences) for their extraordinary help and consistent support. The collaborative and resourceful environment provided by CCDS Lab significantly contributed to the development and execution of our project. Moreover, the encouragement and assistance have been instrumental in shaping our academic and research fields. We are grateful for the opportunities and resources extended to us during our research, which undoubtedly enriched our learning experience.

We acknowledge and appreciate the contributions of all individuals who supported us, both directly and indirectly, during this endeavor.

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. The existing Bangla datasets are either synthetically produced or derived from newspapers, lacking the day-to-day patterns of informal communication. To bridge this gap, we introduce a meticulously categorized real-life dataset encompassing $\approx$ 25K authentic Bangla comments from Youtube, a commonly used social media platform. Applying various machine learning and deep learning models, including transformer models, we aimed to detect, categorize, and correct textual errors to contribute to the preservation and authenticity of the Bangla language. Moreover, we adapted a graph-based transformer model, BanglaBertGCN, for error classification. Through rigorous comparative analysis of distinct models, our study highlights BanglaBertGCN's superiority in error-category classification and underscores the effectiveness of BanglaT5 for text correction. BanglaBertGCN achieves accuracy of 85.6% and 75.8% for binary and multiclass error-category classification. Moreover, BanglaT5, fine-tuned and tested with our corrected ground truths, achieves the best Rouge-L score (0.8962). Furthermore, through a comparative analysis on the performance of BanglaT5 and ChatGPT, a widely used large language model, using zero-shot prompting, revealed that BanglaT5 outperformed ChatGPT in error correction by 29.32%. This finding further emphasizing the impact of our dataset on model efficacy.

Contents

List of Figures

List of Tables

1 Introduction

The Bangla language, also known as Bangla, boasts a rich history deeply rooted in the broader South Asian region. It is the spoken and written form of expression for approximately 230 million people worldwide, making it one of the most widely spoken languages globally. According to data from the CIA World Factbook, Bangla ranks as the sixth most spoken language on the planet .[5] Yet, its significance goes beyond sheer numbers, for Bangla is renowned for its intricate and distinctive style. This language isn't just a medium of communication; it's a carrier of culture, history, and literature. However, in the contemporary world of communication, particularly on digital platforms like social media, the fluidity of text entry often leads to deviations from the language's original form. These deviations can challenge the integrity of the language and its roots. The intricacies of the Bangla script, comprising 50 letters, including 11 vowels and 39 consonants, symbolize the challenge posed by the language's script in the digital realm .[25] These complexities often surface in the form of typographical errors, ultimately affecting the standard of the language and its cohesion within the digital landscape.

Among the plethora of Bangla letters, it's crucial to acknowledge that certain characters introduce a level of intricacy in the act of writing. This complexity is pivotal because it forms the very foundation of potential discrepancies that can arise between what's written and what's spoken in the Bangla language. It's imperative to delve into the specifics, and this is where the illustrative insights presented in Figure 1 shed light on the interchanging of letters that share phonetic similarities in Bangla. This linguistic phenomenon involving phonetically similar letters leads to their interchangeable use, particularly in the pronunciation of words. In practicality, this interchangeability significantly contributes to the occurrence of errors within words, consequently affecting the language's originality and coherence. These errors pose a substantial challenge, especially concerning the language's inherent richness, both culturally and linguistically, and its seamless transition into the digital realm. Thus, comprehending these intricacies within the Bangla script becomes crucial for safeguarding the language's standardness and preserving its unique cultural and linguistic heritage.

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

Figure 1: Illustrative Instances of Various Error Types in Bangla Text, Encompassing Word and Letter Discrepancies.

To enhance the preservation of the Bangla language, we created a comprehensive dataset consisting of real-life Bangla comments from YouTube. This dataset reflects day-to-day communication patterns. Using various machine learning and deep learning models,

including transformer models, we put tremendous effort into detecting, categorizing, and correcting errors. We further explored the capabilities of BanglaBertGCN, a graph-based transformer model. Comparing different models, our analysis highlighted its superiority in error-category classification. Besides, we found the effectiveness of BanglaT5 for text correction. In a comparative analysis with ChatGPT, a popular large language model, BanglaT5 trained on our dataset outperformed, emphasizing the dataset's impact on model efficacy.

1.1 Problem Statement

In the vast landscape of online platforms, where digital interactions unfold with increasing frequency, it's strikingly evident that users have embraced an informal variant of the Bangla language. This informal language variant emerges as a distinctive form characterized by the pervasive influence of regional speech patterns, a profound affinity for local dialects, and the abundant use of colloquial expressions commonly found among residents of specific geographic areas. The amalgamation of these factors shapes and defines the language's informal version in an online context. In essence, it deviates significantly from the standard Bangla, which forms the backbone of the language's formal framework. This deviation is a testament to the dynamic nature of communication within the digital sphere. Within this landscape, brevity, speed, and the freedom to express oneself informally tend to take precedence over traditional linguistic norms. As a result, we observe an intriguing transformation in the language's form and character. This evolution stands as a reflection of the ever-changing linguistic landscape in the realm of digital communication, offering a unique perspective on the continuous ebb and flow of linguistic trends.

1.2 Motivation & Purpose

The impetus behind the creation of a novel dataset for Bangla text error correction stems from a critical evaluation of existing datasets. These datasets, often sourced from formal platforms like newspapers or synthetically generated, exhibit a glaring gap in representing the linguistic nuances prevalent in everyday online interactions. Recognizing this deficiency, our endeavor to craft a new dataset turned towards the vibrant realms of a commonly used social media platform like YouTube. Beyond a mere compilation of linguistic samples, this dataset is a deliberate response to the ever-evolving digital landscape, capturing the dynamic language shifts and errors among Bangla-speaking internet users in Bangladesh.

The motivation is to correct those errors and informal deviations of the language to maintain linguistic standards. Our motivation extends beyond linguistic precision. It embodies a commitment to encapsulate the essence of contemporary communication in the digital era, where language continually adapts and transforms. In our exploration of creating a dataset that resonates with the intricacies of modern language use, we sought inspiration from the diverse linguistic expressions found on widely-used platforms like YouTube. The decision to shift focus from formal sources like newspapers to dynamic online spaces arose from a profound understanding of the evolving nature of communication. Formal sources often fail to encapsulate the informal, regionally influenced, and context-dependent language variations that characterize online interactions. Therefore, our new dataset is not just a repository of linguistic artifacts; it is a dynamic representation of the living language as it breathes and evolves in the digital sphere.

1.3 Our Contributions

We took the initiative to introduce a groundbreaking dataset for Bangla text error correction named **BaTEClaCor**, A Novel Dataset for **Ba**ngla **T**ext **E**rror **Cla**ssification and **Cor**rection. Our new dataset provides a comprehensive representation of the everyday informal and formal interactions of Bangla language users on diverse online platforms. This dataset stands as a testament to our commitment to ensuring that the field of Bangla language correction keeps pace with the ever-evolving digital linguistic landscape, addressing the specific nuances of language errors encountered within the context of contemporary online communication.

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:

- **Creation of a new dataset:** An expansive and authentic dataset comprising $\approx 25,000$ of diverse Bangla comments from YouTube videos has been created with tremendous effort. The dataset can enhance the generation capability of transformer-based models by providing valuable insights into the informal and regionally influenced Bangla language.
- **Error Classification:** Performance analysis of several advanced machine learning and deep learning models, including transformer models BanglaBERT[2], LSTM[12], XLM-RoBERTa[7], and graph-based transformer model BanglaBertGCN[10], 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.
- **Error Correction:** BanglaT5[4] and BanglaT5 Small[4] were analyzed for performance and utilized in correcting various categories of textual errors, including phonetic and grammatical errors. This analysis involved both fine-tuning and testing with our proposed dataset.
- **Comparative Analysis with ChatGPT**: Recognizing the growing prominence of generative models like ChatGPT, we further evaluated our dataset's impact by testing BanglaT5 (trained with BaTeClaCor) and ChatGPT on correcting Bangla comments. The analysis revealed the unique strengths of BanglaT5, particularly its nuanced understanding of informal and regionally influenced Bangla, leading to superior error correction performance compared to ChatGPT. This comparison underscores the crucial role of the BaTeClaCor dataset in fine-tuning and enhancing the capabilities of language models for specific tasks.

These contributions represent a significant leap forward in the realm of online linguistic interactions. With their notable impact, they set the stage for the cultivation of a more precise and digitally literate environment that caters to the unique needs and preferences of Bangla speakers. By nurturing meaningful communication and fostering a profound understanding in the digital realm, our research endeavor takes substantial strides towards enriching the linguistic tapestry of the online Bangla community. In doing so, we contribute to the creation of an online space that thrives on linguistic precision, coherence, and the cultivation of a deep sense of mutual understanding among its diverse users.

In **Section 2**, we conducted a literature review about diverse approaches in Bangla text error classification datasets. We discussed our criteria for extracting the comments to produce a dataset and the structure of the metadata in detail in **Section 3**. In **Section**

4, a description of baseline models has been provided that were used in benchmarking the dataset. Hyperparameter settings for different machine learning models and transformer models along with evaluation metrics for baseline model performance has been discussed briefly in **Section 5**. Results and analysis of the baseline model performance based on our produced dataset has been provided in **Section 6**.

2 Literature Review

Numerous endeavors have been initiated to enhance the refinement of Bangla text correction, despite the language's classification as low-resource. These efforts span diverse directions, which can be divided into two main areas: dataset development and the application of various models.

2.1 Existing Datasets for Bangla Text Classification and Correction:

- H.A.Z. Sameen presented a Bangla error correction dataset encompassing training samples of 9385 sentence pairs, while the testing set involved 5,000 test sentences. Each sentence in the training set has grammatical errors intentionally highlighted with a special symbol.[20]
- Tanni Mittra created a dataset comprising 50,000 pairs of Bangla words. Each pair consists of a correctly spelled word and its corresponding misspelled variation. Notably, a thorough Bangla word dictionary, encompassing approximately 600,000 words, was meticulously compiled from diverse sources, including online repositories, newspapers, social networking sites, and Bangla blogs.[15]
- Chowdhury Rafeed utilized a synthetic dataset derived from the Prothom-Alo 2017 online newspaper for training, while testing involved 6,300 errorful sentences from the Nayadiganta online newspaper, each of which was meticulously annotated.[17]
- Md. Habibur Rahman Sifat utilized a Bangla corpus containing 6.5 million sentences. The research focused on frequently occurring words within the corpus, extracting 8,637 words appearing over 1000 times each. These words formed the foundation for the subsequent error generation analysis.[21] This synthetic dataset of misspelled words was used for training spell-checking models to handle authentic errors in Bangla text.

2.2 Baseline Models for Bangla Text Classification and Correction:

- H.A.Z. Sameen introduced a pioneering methodology for Bangla grammatical error detection through the utilization of a T5 Transformer model.[20] The paper also analyzed the errors detected by the model and discussed the challenges of adapting a translation model for grammar detection tasks.

- Tanni Mittra proposed n-gram models for spell-checking [15]. N-grams are created for each candidate word, including both the correct and misspelled versions. The models compare the n-grams of candidate words against the reference dictionary and calculate their probabilities. The candidate word with the highest probability and closest match to the dictionary is identified as the correct spelling.

- Chowdhury Rafeed presented BSpell, an innovative Bangla spell checker that leverages a combination of Convolutional Neural Networks (CNNs) and the Bidirectional Encoder Representations from Transformers (BERT) model to achieve accurate spell checking in Bangla text [17].

- Md. Habibur Rahman Sifat introduces a novel error generation model for simulating realistic spelling mistakes in Bangla text. Unlike deterministic approaches, it employs a stochastic algorithm based on Bangla writing patterns and the QWERTY keyboard layout to probabilistically create various word variations. While not delving into algorithm-specifics or exact probabilities, the model generates errors such as phonetic substitutions, insertions, deletions, and QWERTY-specific mistakes [21].

- Farhan Noor Dehan demonstrated the superior performance of BanglaBertGCN over traditional and transformer-based models in Bangla text classification significantly[10]. This achievement is attributed to the amalgamation of extensive pretraining, fine-tuning, and transductive learning in BanglaBertGCN. The model leverages large-scale pretraining through a BERT model for generating representations of document nodes in a text graph. These representations serve as inputs to a Graph Convolutional Network (GCN), where document representations are iteratively updated based on graph structures. The final representations are then forwarded to a softmax classifier for predictions, showcasing the synergistic strengths of pre-trained and graph models.[14]

Although these studies collectively contribute significantly to the advancement of Bangla text correction techniques, encompassing various methodologies and datasets, they also exhibit drawbacks in error categorization and granularity, hindering nuanced understanding and tailored correction techniques for Bangla grammatical errors. Focusing on word-level errors may lack comprehensiveness, while synthetic training data raises concerns about representativeness and adaptability to real-world language usage. Reliance on newspapers may introduce biases towards formal language, potentially compromising authenticity in representing informal online language patterns. Additionally, concentration on frequently occurring words might unintentionally skew the dataset, emphasizing the stochastic nature of real-world language errors.

3 Introducing BaTeClaCor Dataset

Introducing a novel dataset will lay the foundation for a significant advancement in Bangla Natural Language Processing (NLP). Through the thoughtful combination of both error-free and errorful comments spanning various genres, the dataset offers a comprehensive insight into the intricacies of real-world language usage and prevalent typing errors. Positioned as a valuable resource, its potential extends to fostering the development and refinement of typing error detection models, thereby enhancing the linguistic quality and overall effectiveness of online communication in Bangla.

3.1 Source of Data Samples

The primary conduit for our data samples is YouTube, a social platform that boasts staggering popularity in Bangladesh, encompassing approximately 34.50 million Bangladeshi users [9]. This selection positions YouTube as a microcosm of the linguistic diversity thriving within the country. The platform acts as a melting pot, attracting users from diverse backgrounds with varying levels of literacy and exhibiting distinct linguistic patterns. Leveraging YouTube's API, our web scraping process specifically targeted randomly listed videos, as highlighted in Figures 2 and 3, each accumulating over 500k views from November 2008 to January 2024, to ensure a substantial number of comments for data retrieval, as videos with considerable viewership are more likely to have high user engagement and comments, providing a richer dataset for analysis. This deliberate randomness in the selection process safeguards against potential biases, ensuring our dataset is a rich tapestry of diverse linguistic expressions and errors. To maintain objectivity, we systematically collected around 60 comments per video, minimizing potential biases and obtaining substantial data for analysis while maintaining a delicate balance in the dataset size. This specific criterion ensures that the dataset remains meaningful, avoiding unnecessary padding and aligning perfectly with the needs of machine learning and deep learning models. Our approach optimizes the efficiency and utility of the dataset, providing a robust foundation for future research and advancements in Bangla language processing.

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.

Algorithm 1 Pseudocode for Comment Scrapping

```
Input: API_KEY = Youtube's API ,
       video_list = ["video_id1", "video_id2", ...,"video_idN"]
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 Web Scraping YouTube Video Details: This figure presents a pseudocode implementation for extracting crucial video metadata from YouTube, encompassing Video ID, Duration, Channel Name, and Comments. The procedure involves inputting an API Key and a Video List, with the enforced limitation of retrieving up to 60 comments per video containing more than 3 words.

3.2 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. Comprising an extensive collection of $\approx 25,000$ comments, the dataset has undergone scrupulous filtering, ensuring its exclusivity to content composed solely in Bangla letters. Noteworthy is the deliberate inclusion of comments featuring emojis, recognizing their potential to provide essential contextual information. The comprehensive nature of the dataset is laid bare in Table 3.1, where out of the 25,105 entries, 15,556 represent errorful samples labelled as 1, while the remaining 9549 showcase error-free comments labelled as 0.

Label	**No. of Comments**
1	15556
0	9549

Table 3.1: Distribution of Labels in the Dataset: Comparison of Error-Containing (Labelled as 1) and Correct Samples (Labelled as 0).

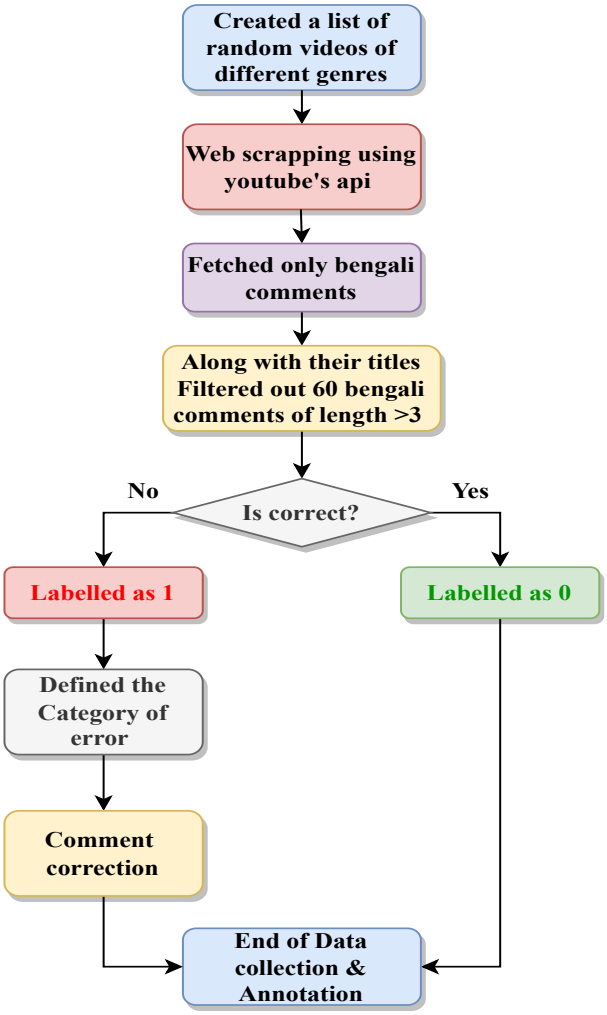


Figure 3: Flowchart of Data Collection and Annotation Process: A list of random videos spanning different genres is generated. A YouTube API key is acquired for web scraping purposes, enabling the extraction of Bangla comments alongside their corresponding titles. Subsequently, comments with a length greater than three words are filtered out, resulting in a collection of 60 comments per video. Each comment is then assessed; if deemed correct, it is labeled as 0 and stored. Otherwise, it is marked as erroneous, with its specific error category defined. Corrected comments are then appended to the dataset for further analysis

Delving into the rich diversity encapsulated within the BaTEClaCor dataset, it becomes evident that its composition transcends mere numerical representation. Figure 4 outlines the distribution of comments across various video genres. The inclusion of genres such as News, Entertainment, Politics, Sports, and the intriguing Miscellaneous category is a thoughtful orchestration. Each genre serves as a microcosm of societal discourse, allowing individuals to express their opinions and ideas through comments. Notably, the genres of News, Entertainment, Politics, and Sports serve as the pillars of societal communication, reflecting the multifaceted nature of discourse within the Bangla context. The Miscellaneous category, however, extends beyond the conventional boundaries set by the more conventional genres. Embracing a myriad of topics, including Lifestyle, Philosophy, Nature, and others, this category mirrors the diverse interests and passions of the Bangladeshi people. By encapsulating a broad spectrum of subjects, the BaTEClaCor dataset not only augments the utility of error detection models but also becomes a reflection of the nuanced and multifaceted tapestry of Bangla language usage in the digital landscape.

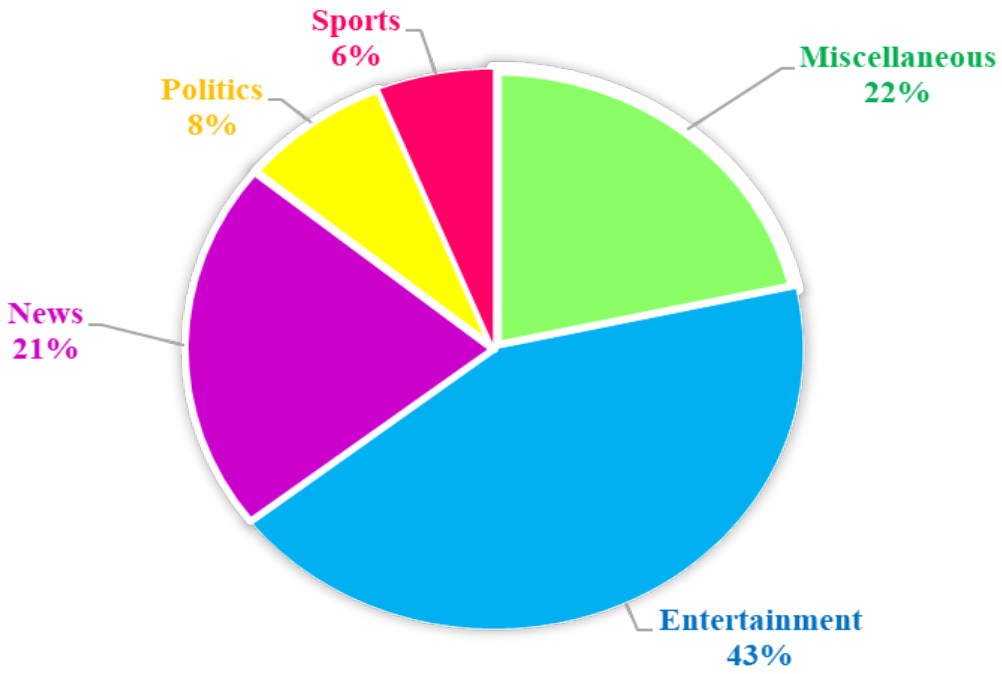


Figure 4: Genre-wise Distribution of Samples: The dataset consists of samples categorized into five main genres: Politics, Entertainment, Miscellaneous, Sports, and News. The Entertainment genre constitutes the largest proportion, accounting for 43% of the dataset. Additionally, the News and Miscellaneous genres represent 21% and 23% of the dataset, respectively. The Politics and Sports genres comprise 8% and 6% of the dataset, respectively.

In Table 3.2, 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.

Error Category	**No. of Comments**
Spelling	10116
Code Switching	2893
Grammatical	2083
Multiple Errors	441

Table 3.2: Distribution of Comments by Error Category in BaTECalCor Dataset.

3.2.1 Spelling

Spelling, being the most commonly occurring category of errors, encompasses instances of incorrect spellings. Moreover, the Bangla language, with its phonetically similar letters, often becomes susceptible to frequent typing errors. Regionally varied informal colloquial expressions or dialects introduce another layer to this category. To maintain precision and adherence to the correct linguistic standards, the report annotates the standard forms of these regionally diverse expressions and dialectal words. In the given statement "খমোতায় বসে খেলা হয় না", there is a spelling error where the term "খমোতায়" should be replaced with the more appropriate term "ক্ষমতায়". The corrected sentence would read as follows: "ক্ষমতায় বসে খেলা হয় না".

3.2.2 Grammatical

Grammatical errors denote mistakes related to the structural and syntactical aspects of the Bangla language. This category encompasses a spectrum of errors, including those associated with incorrect tense usage and improper pronoun placement. Addressing these nuances is crucial for ensuring linguistic accuracy and adhering to the standard grammatical norms of the Bangla language. In the given statement "আল্লাহ তুমি রহম করেন", a grammatical error is identified. The corrected version is as follows: "আল্লাহ তুমি রহম করো".

3.2.3 Code-Switching

Code-switching, often referred to as the mixing of English and Bangla within a single comment, is a phenomenon known as **Banglish**. These instances may not constitute conventional text errors in terms of comprehension or meaning. However, during annotation, the English words present in the comment are replaced by correct and suitable Bangla words while ensuring that the overall meaning remains unchanged. This categorization aims to maintain linguistic authenticity by preserving the essence of the Bangla language, ensuring adherence to standard and widely accepted linguistic norms.In the provided instance "মাহামুদুল্লাহর মতো প্লেয়ার আর হবে না ♡", there is an instance of code-switching. A more formal expression, maintaining linguistic consistency, would be: "মাহামুদুল্লাহর মতো খেলোয়াড় আর হবে না ♡"

3.2.4 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. In the given statement "পাবলিকের দোলায় খেলে সব টিক হয়ে যাবে", there are multiple errors identified (code-switching and spelling). A revised and more academically appropriate version is: "জনগণের ধোলাই খেলে সব ঠিক হয়ে যাবে".

Each record in the dataset features vital information such as Comment, Video ID, Video Title, Channel Name, and Time of Publishing. Each comment is annotated by Genre, Label, Error Category, and Correct Comment. We can see sample data in Table 3.3 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.

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.

Comment	আমরা রিয়াদ মাশারাফিকে আবার দেকতে চাই
Video ID	VWvWUdmO-0I
Video Title	মাশরাফী এখনো বাংলাদেশের ক্যাপ্টেন। অন্য ভূমিকায় ফিরিয়ে আনা হোক তাকে
Channel Name	ON FIELD
Time of Publishing	2023-07-07T15:53:09Z
Genre	Sports
Label	1
Error Category	Spelling
Correct Comment	আমরা রিয়াদ মাশারাফিকে আবার দেখতে চাই

Table 3.3: Sample Data: An excerpt from the appended dataset is presented, showcasing various attributes including Comment, Video ID, Video Title, Channel Name, Time of Publishing, Genre, Label (1 is the label indicating an error, 0 for opposite one), Error Category (categorized as Spelling, Grammatical, Code-switching, or Multiple Errors), and Corrected Comment.

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

In this research, we undertook a binary classification task to determine the presence or absence of errors in textual content. Additionally, we extended our analysis to encompass a multi-class classification framework, delineating errors into five distinct categories: Spelling, Grammatical, Code-switching, Multiple Errors, and Correct. Furthermore, we incorporated the utilization of different variants of the error correction model for the correction of errors within the dataset.

4.1 Classification Models

Our investigation involved the exploration of various machine-learning models, deep-learning architectures, transformer models, and graph-based approaches.

4.1.1 Machine Learning Models

- **TF-IDF + SVM:** The TF-IDF + SVM methodology stands out as an effective approach for text classification, leveraging the synergies between Term Frequency-Inverse Document Frequency (TF-IDF) [19] and the Support Vector Machine (SVM) [11] algorithm. This intricate process begins with the nuanced task of tokenizing the text, breaking it down into its constituent words. The subsequent computation of TF-IDF scores for each word produces a robust feature vector that encapsulates the intrinsic importance of each term across the entire dataset. The TF-IDF score, a key component of this method, is defined as:

$$\text{TF-IDF}(w, d) = \text{TF}(w, d) \cdot \text{IDF}(w) \tag{4.1}$$

  Here, $\text{TF}(w, d)$ denotes the term frequency of the word w within a specific document d, while $\text{IDF}(w)$ signifies the inverse document frequency of the term.

  The ensuing feature vectors (x) serve as potent inputs for training the Support Vector Machine (SVM) model. The SVM model, renowned for its prowess in classification tasks, predicts the class (y) using the following equation:

$$y = \text{sign}(w^T x + b) \tag{4.2}$$

  In this equation, y denotes the predicted class, determined by the sign of the dot product between the weight vector (w) and the feature vector (x), augmented by the bias term (b). The SVM model's ability to discern classes is facilitated by identifying the hyperplane that optimally separates different classes within the feature space [8]. This amalgamation of TF-IDF and SVM thus forms a robust classification framework, adept at navigating the intricacies of text-based datasets. The feature vectors derived from TF-IDF capture the nuanced semantics, allowing the SVM

model to discern patterns and make accurate predictions, making this approach particularly well-suited for text classification tasks.

- **TF-IDF + Random Forest:** The TF-IDF + Random Forest methodology presents a robust framework for text classification, leveraging the synergy between Term Frequency-Inverse Document Frequency (TF-IDF) [19] and the Random Forest algorithm [18]. The initial step involves the creation of feature vectors using TF-IDF, encapsulating the semantic relevance of words within the text. These feature vectors serve as inputs for training a Random Forest model, a powerful ensemble learning algorithm.

  The Random Forest model is composed of multiple decision trees, each contributing to the collective prediction of the document's class (y). The prediction mechanism involves considering the class with the highest cumulative probability across all decision trees in the ensemble. Mathematically, this can be expressed as:

$$y = \text{argmax}_c \sum_{t=1}^{T} p(c|x_t) \tag{4.3}$$

  Here, T represents the total number of decision trees within the Random Forest model, c signifies the class label, and x_t denotes the feature vector under consideration. The probability $p(c|x_t)$ is calculated individually by each decision tree in the ensemble, reflecting the likelihood of the document belonging to a particular class.

  The strength of the Random Forest model lies in its ability to mitigate overfitting and enhance generalization. By aggregating predictions from multiple decision trees, the model achieves a robust and accurate classification, capable of handling complex relationships within the data. This approach is particularly advantageous in text classification tasks where the presence of diverse linguistic patterns requires a versatile and adaptive learning algorithm [22].

- **TF-IDF + XGBoost:** The TF-IDF + XGBoost methodology unfolds as a sophisticated and effective strategy for text classification, amalgamating the power of Term Frequency-Inverse Document Frequency (TF-IDF) [19]feature vectors with the robust XGBoost model.[6] This approach strategically leverages XGBoost's proficiency in deciphering intricate relationships inherent in textual data. The predictive mechanism of the XGBoost model is intricately characterized by the summation of predictions originating from multiple decision trees, embodying the ensemble learning paradigm:

$$g(x) = \sum_{t=1}^{T} \beta_t f_t(x) \tag{4.4}$$

  In this equation, $\boldsymbol{g(x)}$ denotes the predicted value for the input feature vector x. The contribution of each tree to the prediction is encapsulated by $\boldsymbol{\beta_t f_t(x)}$, where $\boldsymbol{\beta_t}$ signifies the coefficient for the t-th tree, and $\boldsymbol{f_t(x)}$ represents the prediction of the t-th tree. The ensemble learning capabilities of XGBoost empower the model to predict the class of the text by selecting the class associated with the highest predicted value [16]. This intuitive and powerful strategy enables the model to make informed predictions, drawing upon the collective knowledge embedded within the diverse decision trees.

- **TF-IDF + Naive Bayes:** The TF-IDF + Naive Bayes approach represents a powerful paradigm for text classification, synergizing the strengths of Term Frequency-Inverse Document Frequency (TF-IDF) [19] and the Naive Bayes algorithm.[1] The methodology initiates with the segmentation of the text into individual words, followed by the computation of TF-IDF scores for each word. The TF-IDF score, a fundamental element of this approach, is defined as:

$$\text{TF-IDF}(w, d) = \text{TF}(w, d) \cdot \text{IDF}(w) \tag{4.5}$$

Here, $\text{TF}(w, d)$ stands for the term frequency of the word w within a specific document d, while $\text{IDF}(w)$ signifies the inverse document frequency of the term.

The computed TF-IDF scores contribute to the creation of feature vectors (x), which serve as inputs for training the Naive Bayes model. The Naive Bayes algorithm, based on probabilistic principles, predicts the class (y) using Bayes' theorem:

$$P(y|x) = \frac{P(x|y) \cdot P(y)}{P(x)} \tag{4.6}$$

The prediction involves selecting the class with the maximum probability. The Naive Bayes model assumes independence between features, simplifying the calculation of conditional probabilities.

The amalgamation of TF-IDF and Naive Bayes forms a robust classification framework adept at handling text-based datasets. The TF-IDF scores encapsulate the semantic importance of words, allowing the Naive Bayes model to discern patterns and make accurate predictions. This approach is particularly well-suited for text classification tasks due to its simplicity, efficiency, and ability to handle high-dimensional feature spaces [26].

4.1.2 Deep Learning Models

- **LSTM:** Long Short-Term Memory (LSTM)[12], a pivotal component in natural language processing, excels at capturing intricate dependencies within sequential data. In the realm of text classification, LSTM processes an input sentence $S = \{x_1, x_2, ..., x_n\}$ from the dataset, where x_i represents individual words. The input sentence undergoes an initial embedding layer, yielding embedded representations $E = \{e_1, e_2, ..., e_n\}$. These embedded representations serve as inputs to the LSTM model, producing hidden representations $H = \{h_1, h_2, ..., h_n\}$.

LSTM's strength lies in its ability to comprehend contextual information across the entire sequence. This is achieved through the intricate interplay of three gates: the input gate (i_t), forget gate (f_t), and output gate (o_t). These gates, governed by sigmoid and tanh activation functions, regulate the flow of information within the LSTM cell. The computation for each gate is defined as follows:

$$\begin{aligned} i_t &= \sigma(W_{ii} \cdot x_t + b_{ii} + W_{hi} \cdot h_{t-1} + b_{hi}) \\ f_t &= \sigma(W_{if} \cdot x_t + b_{if} + W_{hf} \cdot h_{t-1} + b_{hf}) \\ o_t &= \sigma(W_{io} \cdot x_t + b_{io} + W_{ho} \cdot h_{t-1} + b_{ho}) \end{aligned} \tag{4.7}$$

Here, x_t represents the input at time step t, h_{t-1} denotes the hidden state from the previous time step, and σ signifies the sigmoid activation function. The weighted inputs and biases for each gate are denoted by W and b, respectively.

The LSTM cell computes the cell state (c_t) and hidden state (h_t) at each time step using the input, forget, and output gates:

$$\begin{aligned} c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_{ic} \cdot x_t + b_{ic} + W_{hc} \cdot h_{t-1} + b_{hc}) \\ h_t = o_t \cdot \tanh(c_t) \end{aligned} \tag{4.8}$$

The final prediction is obtained by passing the last layer's hidden representation (h_t) through a linear layer and applying the Softmax activation function:

$$\hat{y} = \text{Softmax}(W_{\text{c}} \cdot h_t + b) \tag{4.9}$$

This comprehensive LSTM model integrates these gates, enabling it to capture and utilize contextual information efficiently for text classification tasks. The architecture's adaptability to diverse sequential patterns makes it a powerful tool in natural language processing [12].

- **LSTM with Attention:** Long Short-Term Memory [12] with Attention mechanism [27] is an advanced neural network architecture that combines the strengths of LSTM in capturing sequential dependencies with the attention mechanism's ability to dynamically focus on specific parts of the input sequence. The model begins by generating hidden states (h_i) from the input sequence using the LSTM layer. These hidden states encapsulate the contextual information from the entire sequence.

  The attention mechanism is then employed to calculate attention scores (α_i) for each hidden state. The attention scores are computed using the following equation:

$$\alpha_i = \text{Softmax}(W_{hi} + b) \tag{4.10}$$

  In this equation, W_{hi} and b denote the weight matrix and bias term, respectively. The Softmax activation function is applied to normalize the attention scores, ensuring that they represent valid probabilities.

  Subsequently, the context vector (c_i) for each sentence is computed by taking a weighted sum of the hidden states ($\hat{h}_j$) across all time steps (j), where the weights are determined by the attention scores (α_{ij}):

$$c_i = \sum_{j=1}^{n} \alpha_{ij} \hat{h}_j \tag{4.11}$$

  This attention mechanism allows the model to dynamically assign importance to different parts of the input sequence, enabling it to focus on the most relevant information for the given task. The resulting context vectors (c_i) enriched with the attention mechanism's insights are then utilized for the subsequent classification process.

  The LSTM-Attention model's capability to adaptively prioritize specific elements within the input sequence contributes to its enhanced performance in tasks requiring nuanced understanding and context-aware predictions [24].

- **CNN-LSTM:** The CNN with LSTM architecture is a hybrid model that seamlessly integrates Convolutional Neural Network (CNN)[13] and Long Short-Term Memory (LSTM)[12] components to capitalize on their complementary strengths.

The model initiates with the input sequence x undergoing convolutional processing through a CNN, resulting in the generation of feature maps denoted as f [13]. The computation of feature maps at each time step t is expressed by the convolutional operation:

$$f_t = \text{CNN}(x_t) \tag{4.12}$$

Here, x_t represents the input sequence at time step t, and the CNN operation captures essential patterns and spatial hierarchies within the input data. The resulting feature maps f_t serve as enriched representations that retain crucial information for subsequent processing.

The subsequent step involves leveraging these feature maps (f) in conjunction with the LSTM model to capture both spatial and temporal dependencies. The LSTM computes hidden states h using the following recurrent computation:

$$h_t = \text{LSTM}(f_t, h_{t-1}) \tag{4.13}$$

In this equation, h_{t-1} signifies the hidden state from the previous time step. The integration of CNN and LSTM components in this architecture enhances the model's ability to capture intricate patterns in both spatial and temporal dimensions. This makes the CNN with LSTM architecture particularly well-suited for tasks requiring a nuanced understanding of input sequences, where the combination of convolutional and recurrent operations allows the model to learn hierarchical representations and long-term dependencies simultaneously.

4.1.3 Transformer Models

- **BanglaBERT:** BanglaBERT[2], a transformer-based model, exhibits exceptional proficiency in text classification tasks through a comprehensive multi-stage process.[2] Initially pre-trained on extensive Bangla language corpora, BanglaBERT captures intricate contextual information using transformer layers.

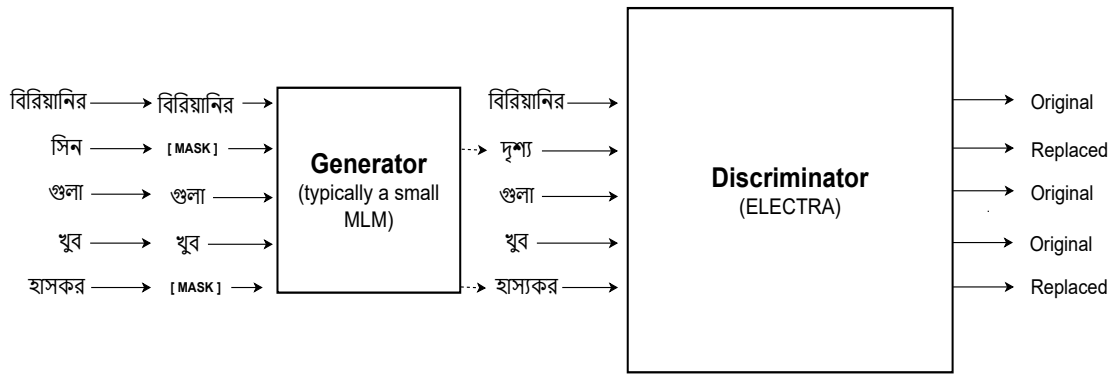


Figure 5: Block Diagram of BanglaBERT: The diagram illustrates the architecture of BanglaBERT. BanglaBERT is trained on the Electra model framework, which enhances efficiency and performance. This diagram provides an overview of the various components and layers involved in the BanglaBERT model, highlighting its structure and key elements.

For an input sentence S, we obtain $S = \{t_1, t_2, \ldots, t_n\}$ after passing the sentence into the BanglaBERT tokenizer, where t_i represents the i-th token.After passing the sentence S through a BanglaBERT model, we obtain contextual representations for each token t_i, denoted as $H = \{h_1, h_2, \ldots, h_n\}$, where h_i represents the contextual representation of token t_i. In this case, we consider the last layer hidden representations of the BanglaBERT encoder. To represent an input sentence, the model leverages a special token, often denoted as [CLS], which encapsulates the entire sentence's semantics. This token's representation, denoted as h_{CLS}, plays a pivotal role in the subsequent classification process by following method.

$$z = W_2 \cdot (\text{ReLU}(W_1 \cdot h_{\text{CLS}} + b_1)) + b_2 \tag{4.14}$$

Here W_1 and W_2 are the learnable weights and z is the logits value.

- **XLM-RoBERTa:** XLM-RoBERTa, another formidable transformer-based model, shares the foundational principles of BanglaBERT but extends its capabilities through cross-lingual pre-training. The model is initially trained on a diverse range of languages, allowing it to comprehend and generalize linguistic patterns across multiple language domains. This cross-lingual pre-training enhances XLM-RoBERTa's ability to handle diverse linguistic nuances and nuances present in various languages [7].

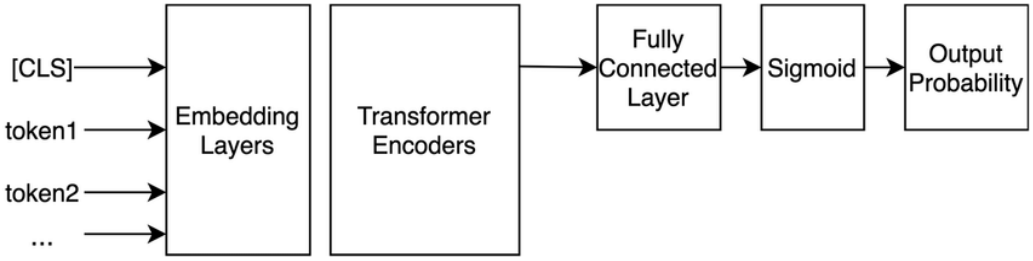


Figure 6: Architecture of XLM-Roberta: This diagram depicts the architectural design of XLM-Roberta, a multilingual pre-trained language model. The diagram offers a comprehensive visualization of the model's internal components and connections, facilitating a better understanding of its functioning and utility in multilingual natural language processing tasks.

Similar to BanglaBERT, XLM-RoBERTa utilizes the [CLS] token to represent entire sentences, ensuring a fixed-size representation h_{CLS} that encapsulates the semantic richness of the input. We also follow the same equation described in Eqn 4.14 for doing the classification.

4.1.4 Transformer-based GCN Model

BertGCN [14], an innovative fusion of BERT-based language models and Graph Convolutional Networks (GCN), revolutionizes natural language processing by leveraging contextual embeddings and graph-based structures. This novel approach enhances the understanding of complex relationships within textual data, empowering more accurate and nuanced language representations for various applications, from sentiment analysis to information retrieval.[10, 14] The construction of the graph is crucial, laying the foundation for fundamental operations such as message passing and aggregation in the graph-based framework.

Textual data is converted into a graph structure where unique words and documents become nodes (V), interconnected by edges (E). This graph, denoted as $G = (V, E)$, serves as the foundation for subsequent graph-based operations. Models in this framework utilize an adjacency matrix (A), sized $N \times N$ for N nodes, which captures relationships by incorporating edge weights derived from word co-occurrence and TF-IDF for documents. Formally, the adjacency matrix

$$A_{i,j} = \begin{cases} \text{PMI}(i,j), & \text{if } i,j \text{ are words} \\ \text{TF-IDF}(i,j), & \text{if } i \text{ is doc \& } j \text{ is word} \\ 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{4.15}$$

Here, PMI (Pointwise Mutual Information) values represent the co-occurrence between words and TF-IDF values capture the importance of words within documents.

Document embeddings are generated by the BERT model, forming the initial node feature matrix X. This matrix is the input for the GCN model, where each layer's output is computed as:

$$L^{(i)} = \rho\left(\tilde{A}L^{(i-1)}W^{(i)}\right) \tag{4.16}$$

Here, $L^{(i)}$ represents the output feature matrix of the i-th GCN layer, ρ is an activation function, $\tilde{A}$ is the normalized adjacency matrix, and $W^{(i)} \in R^{d_{i-1} \times d_i}$ is a weight matrix for the i-th layer.

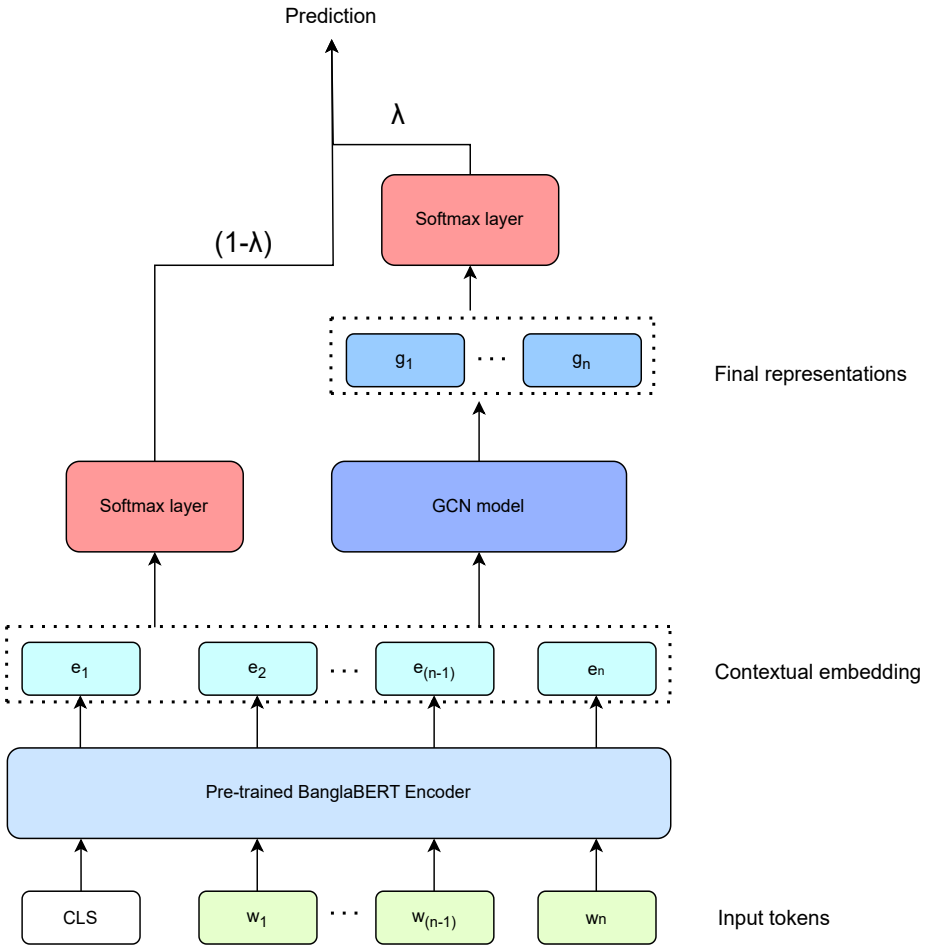


Figure 7: Block Diagram of BanglaBertGCN: This schematic illustrates the architectural framework of BanglaBertGCN, an advanced variant of the BanglaBERT model enhanced with Graph Convolutional Networks (GCNs). By integrating GCNs into the BanglaBERT architecture, BanglaBertGCN leverages graph-based representations to capture contextual relationships among words and sentences in Bangla text.

In practice, enhancing BanglaBertGCN involves incorporating an auxiliary classifier directly on BERT embeddings. This auxiliary classifier, activated using softmax, is formed by feeding document embeddings (denoted by X) to a dense layer. The final training objective linearly interpolates predictions from BanglaBertGCN and BERT:

$$Z = \lambda * Z_{\text{GCN}} + (1 - \lambda) * Z_{\text{BERT}} \tag{4.17}$$

Here, Z is the final prediction, λ determines the balance between BanglaBertGCN and BERT predictions, and Z_{GCN} and Z_{BERT} represent predictions from the GCN and BERT models, respectively.

4.2 Error Corrector Model

BanglaT5[3] leverages the powerful sequence-to-sequence (seq2seq)[23] architecture for textual error correction tasks, transforming input Bangla text into accurate and semantically sound outputs. The input sequence is represented as $S = \{t_1, t_2, ..., t_n\}$, where each t_i symbolizes the i-th token (word or subword) in the Bangla sentence. To unlock the inherent meaning within these tokens, BanglaT5 employs a token embedding layer, converting each t_i into a numerical representation e_i. This transformation captures the semantic essence of each token and its relationships with others, laying the foundation for understanding the broader context of the sentence.

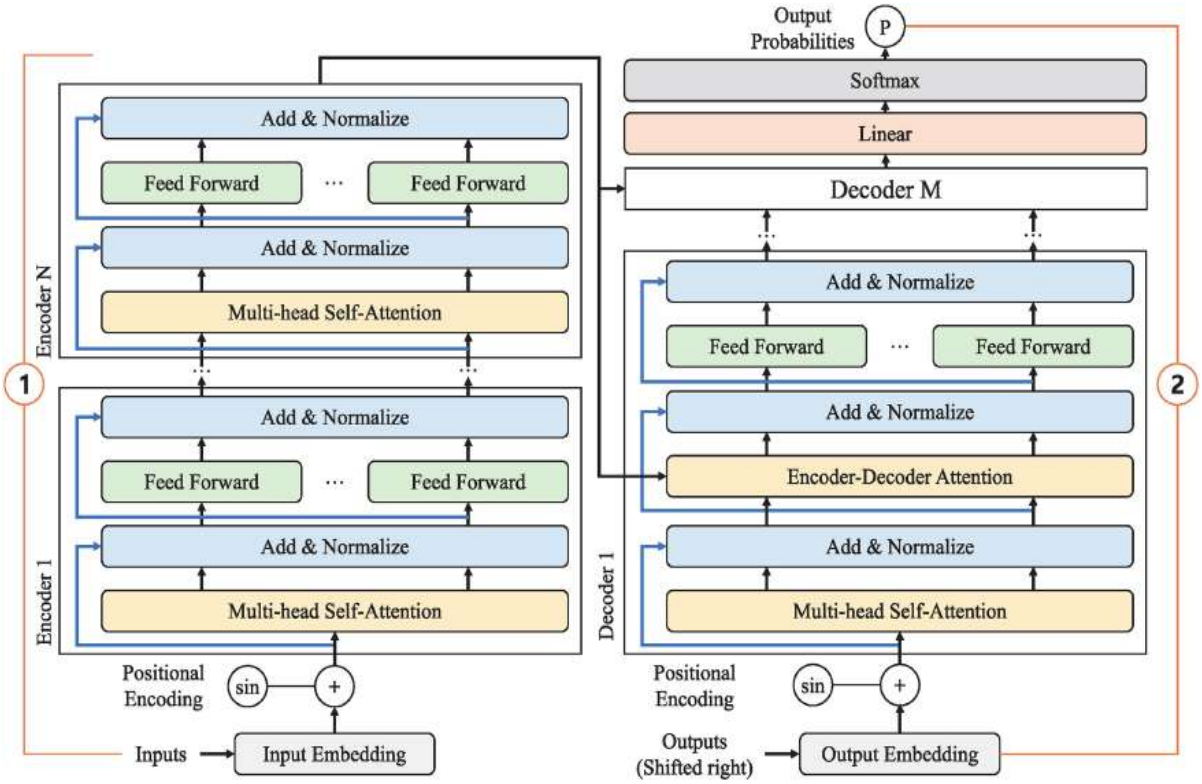


Figure 8: Architecture of BanglaT5: This diagram presents the architectural layout of BanglaT5, an innovative model inspired by the Transformer-based T5 architecture, specifically tailored for Bangla language processing tasks. BanglaT5 incorporates pre-training on diverse Bangla text corpora, followed by fine-tuning on task-specific datasets. This diagram offers a comprehensive visualization of BanglaT5's modular components and intricate connections, providing insights into its advanced mechanisms for generating and comprehending Bangla text.

Armed with the token embeddings $E = \{e_1, e_2, ..., e_n\}$, BanglaT5 delves into the encoder, a multi-layered transformer structure. Within each layer, self-attention is calculated to get the contextual representations:

$$\text{Self-Attention}(Q, K, V) = \text{softmax}(Q^T K / \sqrt{d_k}) \cdot V \tag{4.18}$$

Here, $Q = W_Q X$, $K = W_K X$, $V = W_V X$ are query, key, and value matrices respectively which are extracted from X and W_Q, W_K, W_V are the learnable matrices. This equation calculates the attention score between different parts of the input sequence, essentially finding the relevance between one token and another to understand the overall meaning. By attending to these relationships, the encoder gains a deeper comprehension of the context and meaning embedded within the input sentence. Then the decoder generates the corrected output sequence. It receives the encoder's final hidden representation

(H_e), a concentrated summary of the input's meaning, as its guiding light. Equipped with this knowledge, the decoder initiates a step-by-step token generation process. At each step, it leverages self-attention to understand the relationships within the partially generated output sequence, ensuring coherence and logical flow.

$$\text{Encoder-Decoder Attention}(Q_d, K_e, V_e) = \text{softmax}(Q_d^T K_e / \sqrt{d_k}) \cdot V_e \tag{4.19}$$

Here, Q_d represents the decoder's current query (seeking relevant information), K_e and V_e represent the keys and values from the encoder's hidden representation (the input's contextual meaning). This attention mechanism allows the decoder to selectively focus on specific parts of the input that are most relevant to the word it's currently generating, ensuring context-awareness and accuracy in the output. The decoder diligently continues producing outputs until a special "end-of-sequence" token signifies completion. The final output sequence, denoted as $S' = \{t'_1, t'_2, ..., t'_m\}$, represents the corrected text. This is the culmination of the seq2seq process within BanglaT5, where the model's understanding of the input and its ability to generate contextually relevant text come together to produce remarkable results.

While the seq2seq architecture forms the bedrock of BanglaT5's text processing abilities, it's important to note that BanglaT5 itself is a large language model. This means it goes beyond the core seq2seq framework, incorporating additional layers and fine-tuning specifically tailored to the task at hand. Whether it's textual error correction, question answering, or text generation, these enhancements empower BanglaT5 to excel in diverse language processing endeavors.

Although the full BanglaT5 model exhibits impressive power in Bangla text correction, its smaller version, BanglaT5 small, offers a compelling alternative. This streamlined version sacrifices some complexity for faster processing and lower resource requirements, making it ideal for situations where efficiency is paramount. In essence, BanglaT5 small emerges as a versatile and efficient option for Bangla text correction, particularly when swift execution and resource optimization are primary concerns.

5 Experimental Design

5.1 Experimental Settings

Our initial focus was on text preprocessing to ensure data quality, encompassing basic steps such as removing punctuation and Bangla stop words.

5.1.1 Binary Classification

For binary classification, the baseline models were fine-tuned and tested using the parameter values given in table 5.1.

Model	Parameter Name	Value
TF-IDF + SVM	C	1.0
	kernel	'rbf'
	degree	4
	gamma	'scale'
TF-IDF + Random Forest	max features	5000
	n_estimators	200
	min_samples_leaf	1
TF-IDF + XG-Boost	max features	10000
	colsample_bytree	0.7
	max_depth	6
	n_estimators	700
	subsample	0.7
TF-IDF + Naive Bayes	max features	10000
	alpha	0.1
LSTM	LSTM layer units	128
LSTM with Attention	LSTM layer units	64
LSTM + CNN	Convolutional filters	128
	Convolutional kernel size	5
	LSTM layer units	64
BanglaBert	Learning rate	1e-5
XLM-Roberta	Learning rate	1e-5
BanglaBERTGCN	GCN Layers	3
	Hidden dimension	200
	Learning rate of GCN	1e-3

Table 5.1: Configuration of Baseline Models (Binary Classification): This table outlines the hyper parameter settings for various baseline models utilized in binary classification tasks.

5.1.2 Multiclass Classification

For multiclass classification, the baseline models were fine-tuned and tested by the parameter values given in table 5.2.

Model	Parameter	Value
TF-IDF + SVM	C	1.0
	kernel	'rbf'
	degree	4
	gamma	'scale'
TF-IDF + Random Forest	max_features	10000
	n_estimators	700
	min_samples_split	5
	min_samples_leaf	2
TF-IDF + XG-Boost	max_features	10000
	colsample_bytree	0.7
	max_depth	4
	n_estimators	500
	subsample	0.7
TF-IDF + Naive Bayes	max_features	10000
	alpha	0.1
LSTM	LSTM layer units	100
	Recurrent_dropout	0.2
LSTM with Attention	LSTM layer units	64
LSTM + CNN	Convolutional filters	128
	Convolutional kernel size	5
	LSTM layer units	64
BanglaBert	Learning rate	1e-5
XLM-Roberta	Learning rate	1e-5
BanglaBERTGCN	GCN Layers	3
	Hidden dimension	200
	Learning rate of GCN	1e-3

Table 5.2: Configuration of Baseline Models (Multi-class Classification): This table outlines the hyper parameter settings for various baseline models utilized in multi-class classification tasks.

These models, each tailored with specific hyperparameters, were designed to comprehensively address the binary and multiclass classification task, leveraging a diverse set of approaches from traditional machine learning to advanced deep learning architectures.

5.1.3 Error Corrector Model

The error corrector model was fine-tuned and tested by the parameter values given in table 5.3.

Model Variant	Parameter	Value
BanglaT5 small	Batch size	16
	Learning rate	2e-5
	Gradient accumulation steps	6
BanglaT5 large	Batch size	16
	Learning rate	2e-5
	Gradient accumulation steps	6

Table 5.3: Configuration of Corrector Model Variants: This table presents the hyper parameter settings for different variants of the Corrector Model, including BanglaT5 small and BanglaT5 large.

These parameters were fine-tuned to strike a balance between computational efficiency and model performance. The iterative nature of gradient accumulation steps aids in stabilizing the training process, facilitating more robust and effective learning patterns. Through this thorough exploration of configurations, we sought to harness the full potential of these models, ensuring their optimal performance in handling complex Bangla text data for a variety of applications.

5.2 Evaluation Metrics

In this study, we employed different evaluation metrics to measure the performance of different models. The following evaluation metrics :

Macro Precision (P_M)

Macro Precision is a metric used to evaluate the precision of a classification model on multiple classes. It calculates the average precision across all classes, considering both true positive and false positive instances. The formula is given by the sum of true positives divided by the sum of true positives and false positives for each class.

$$P_M = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FP_i} \tag{5.1}$$

Macro Precision (P_M) is a metric used to assess the precision of a classification model when dealing with multiple classes. In the equation, N represents the total number of classes, TP_i denotes the true positives for class i, and FP_i represents the false positives for class i. The formula computes the average precision across all classes by summing the true positives and dividing by the sum of true positives and false positives for each class.

Macro Recall (R_M)

Macro Recall assesses the ability of a classification model to capture all relevant instances across multiple classes. It calculates the average recall across all classes, taking into account true positives and false negatives. The formula involves the sum of true positives divided by the sum of true positives and false negatives for each class.

$$R_M = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FN_i} \tag{5.2}$$

Macro Recall (R_M) evaluates the ability of a classification model to capture all relevant instances across multiple classes. Similar to Macro Precision, N is the total number of classes, TP_i signifies true positives for class i, and FN_i represents false negatives for class i. The formula calculates the average recall across all classes by summing the true positives and dividing by the sum of true positives and false negatives for each class.

Accuracy (Acc)

Accuracy is a widely used metric to measure the overall correctness of a classification model. It calculates the ratio of correctly predicted instances (true positives and true negatives) to the total number of instances. The formula consists of the sum of true positives and true negatives divided by the sum of true positives, true negatives, false positives, and false negatives.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.3}$$

Accuracy (Acc) is a widely used metric to measure the overall correctness of a classification model. The equation includes terms such as TP for true positives, TN for true negatives, FP for false positives, and FN for false negatives. The accuracy is computed by summing the true positives and true negatives and dividing by the sum of true positives, true negatives, false positives, and false negatives.

Rouge-1 (R_1)

Rouge-1 is a metric commonly used in natural language processing to evaluate the similarity between the output of an automatic system (e.g., text summarization) and a reference summary. It focuses on unigrams and calculates the precision of overlapping unigrams between the reference and system-generated summaries.

$$R_1 = \frac{\sum_{n=1}^{N} \min(\text{count}(\text{ref}, n), \text{count}(\text{system}, n))}{\sum_{n=1}^{N} \text{count}(\text{ref}, n)} \tag{5.4}$$

Rouge-1 (R_1) is a natural language processing metric assessing unigram (word) overlap between a reference summary and a system-generated summary. The equation involves N as the total number of unigrams, and $\text{count}(\text{ref}, n)$ and $\text{count}(\text{system}, n)$ represent the counts of unigram n in the reference and system-generated summaries, respectively.

Rouge-2 (R_2)

Rouge-2 extends the concept of Rouge-1 to consider bigrams. It evaluates the precision of overlapping bigrams between the reference and system-generated summaries. This provides a more detailed analysis of the similarity between the two sets of n-grams.

$$R_2 = \frac{\sum_{n=2}^{N} \min(\text{count}(\text{ref}, n), \text{count}(\text{system}, n))}{\sum_{n=2}^{N} \text{count}(\text{ref}, n)} \tag{5.5}$$

Rouge-2 (R_2) extends the concept of Rouge-1 to evaluate bigram overlap. The equation includes N as the total number of bigrams, and $\text{count}(\text{ref}, n)$ and $\text{count}(\text{system}, n)$ denote the counts of bigram n in the reference and system-generated summaries, respectively.

Rouge-L (R_L)

Rouge-L measures the longest common subsequence between the reference and system-generated summaries. It considers the length of the longest sequence of words that appear in the same order in both summaries. The formula involves the ratio of the length of the longest common subsequence to the length of the reference summary.

$$R_L = \frac{\sum_{n=1}^{N} \min(\text{count}(\text{ref}, n), \text{count}(\text{system}, n))}{\sum_{n=1}^{N} \text{count}(\text{ref}, n)} \tag{5.6}$$

Rouge-L (R_L) measures the longest common subsequence between a reference summary and a system-generated summary. The formula incorporates N as the total length of the longest common subsequence, and $\text{count}(\text{ref}, n)$ and $\text{count}(\text{system}, n)$ signify the counts of words in the reference and system-generated summaries, respectively. 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 Results and Analysis

Classification Types	Model Name	Performance Metrics		
		Accuracy	Macro Precision	Macro Recall
Binary Classification	TF-IDF + RandomForest	60.9	57.0	56.1
	TF-IDF + XGBoost	64.3	61.0	58.2
	TF-IDF + SVM	66.1	63.7	59.9
	TF-IDF + Naive Bayes	67.4	72.7	58.1
	LSTM	66.0	64.0	65.0
	LSTM + Attention	67.0	65.0	61.0
	CNN + LSTM	73.0	73.0	67.0
	XLM-Roberta	79.6	79.8	77.9
	BanglaBERT	80.1	79.8	78.5
	BanglaBERTGCN	**85.6**	**84.8**	**85.6**
Multiclass Classification	TF-IDF + Naive Bayes	49.1	35.1	28.3
	TF-IDF + RandomForest	50.3	31.3	28.5
	TF-IDF + SVM	51.5	41.3	30.4
	TF-IDF + XGBoost	63.4	59.8	57.1
	CNN + LSTM	55.5	39.8	38.7
	LSTM + Attention	59.6	41.9	43.5
	LSTM	59.4	44.0	39.2
	XLM-Roberta	73.7	55.0	55.5
	BanglaBERT	75.1	62.7	53.9
	BanglaBERTGCN	**75.8**	**60.5**	**75.8**

Table 6.1: Classification Performance Comparison of Baseline Models on BaTeClCor Dataset. The table presents performance metrics including accuracy, macro precision, and macro recall for various classification models applied to the BaTeClCor dataset. Models are evaluated under binary and multiclass, utilizing diverse methodologies such as TF-IDF with different classifiers, LSTM, convolutional neural networks (CNN), and pre-trained transformer-based models like XLM-Roberta and BanglaBERT. Notably, BanglaBERTGCN, a Graph based transformer model demonstrates superior performance, outperforming other models across all classification types.

6.1 Binary Classification

Our investigation into binary classification employed a diverse spectrum of machine learning and deep learning models, delving into both traditional approaches and cutting-edge techniques. This section delves into the key findings and insights gleaned from each category of models.

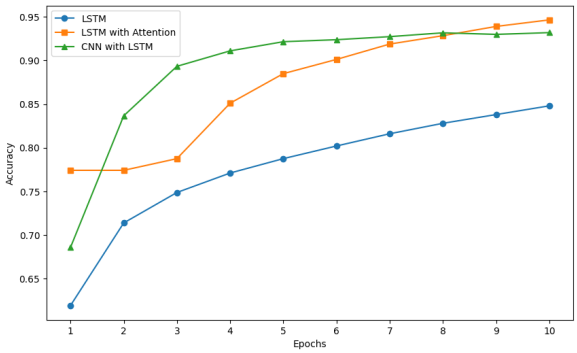
6.1.1 Machine Learning Models:

From Table 6.1, it is evident that among the machine learning models, TF-IDF + Naive Bayes emerges as the standout performer, surpassing TF-IDF + SVM by approximately 1.33%, TF-IDF + XG-Boost by around 3.14%, and TF-IDF + Random Forest by a notable margin of approximately 6.51%. The preeminence of Naive Bayes in this context can be attributed to its inherent probabilistic nature, which adeptly navigates the intricacies of the Bangla text dataset, capturing nuanced linguistic patterns. On the competitive front, TF-IDF + SVM demonstrates commendable accuracy, leveraging its strength in finding optimal hyperplanes within high-dimensional spaces. However, TF-IDF + XG-Boost, despite its robust boosting approach, experiences a marginal shortfall, shedding light on the intricate adaptability required for models to align seamlessly with the nuanced characteristics of Bangla text. Notably, TF-IDF + Random Forest, with its ensemble complexity, registers the lowest accuracy among the models. This discrepancy may stem from the ensemble's intricacies not being fully harnessed within the constraints of the given dataset size, ultimately favoring the simpler yet effective models like Naive Bayes and SVM.

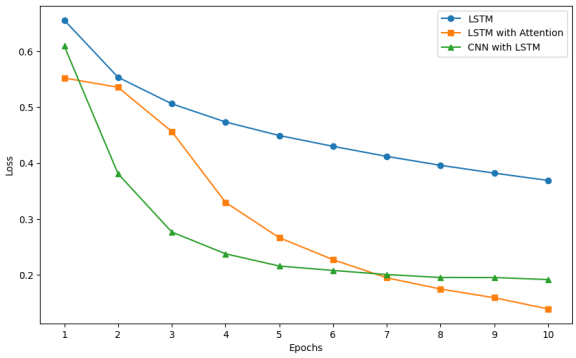
6.1.2 Deep Learning Models:

In the training phase, LSTM models exhibit consistent training accuracy gains of approximately 1.5% to 2.5% for binary classification. LSTM with an attention mechanism showcases a 1% to 2.5% increase in accuracy. CNN with LSTM models demonstrate accuracy improvements ranging from 6% to 10% over the 10 epochs, highlighting their ability to capture nuanced patterns and adapt effectively, as shown in figure 9. Besides, The LSTM models display steady reductions in loss, with percentage improvements ranging from approximately 4% to 9% across the 10 epochs. LSTM with attention mechanism demonstrates more substantial decreases, showing percentage improvements from around 32% to 55%. The CNN with LSTM models exhibit consistent declines in loss, with improvements ranging from approximately 14% to 27%.

From Table 6.1, It is evident that the LSTM model achieved superior accuracy compared to LSTM with attention, showcasing a notable improvement in capturing sequential dependencies within Bangla comments. Although both models demonstrated similar macro precision, the LSTM outperformed LSTM with attention by approximately 2.5%, emphasizing the efficacy of its sequential processing. The CNN + LSTM model showcased an improvement of around 6.7% in accuracy compared to LSTM, and an 8.7% boost over LSTM with attention. This suggests that the combined convolutional and sequential processing in the CNN+LSTM architecture significantly contributed to the superior error detection in Bangla comments. The results emphasize the effectiveness of integrating convolutional layers to capture both sequential and local patterns, thereby enhancing the overall performance for this specific task.



Accuracy Plot

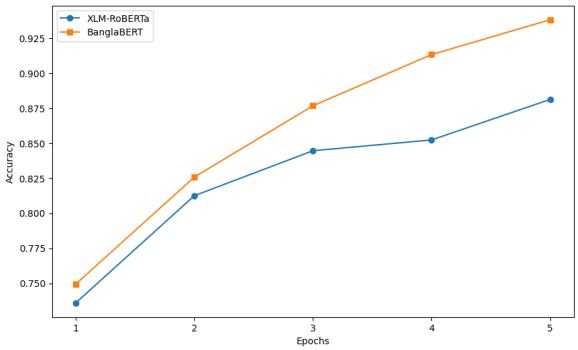


Loss Plot

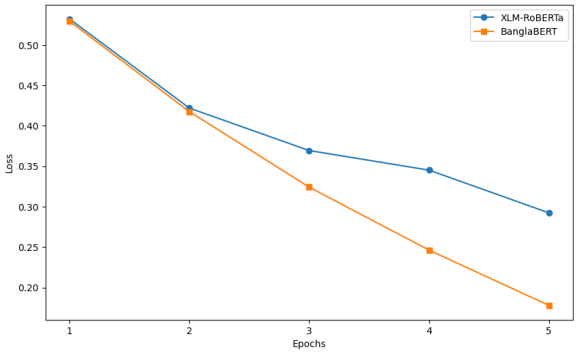
Figure 9: Training Progress of Deep Learning Models (Binary Classification): The plot depicts the evolution of accuracy and loss metrics for LSTM with attention alongside LSTM and CNN with LSTM. Remarkably, LSTM with attention gradually outperformed both LSTM and CNN with LSTM in terms of accuracy, showcasing a steady increase over time until surpassing their accuracy levels. Conversely, LSTM with attention exhibited a more pronounced decrease in loss compared to the other models, highlighting its efficient convergence during training.

6.1.3 Transformer Models:

In the training phase shown in figure 10, XLM-RoBERTa consistently performs, with accuracy gains of approximately 1.5% to 2% over the 5 epochs for binary classification. BanglaBERT demonstrates accuracy improvements ranging from 6% to 8%, showcasing its robust performance. XLM-RoBERTa and BanglaBERT models consistently reduce losses, with improvements of approximately 8% to 45% and 16% to 33%, respectively. This indicates effective loss minimization across epochs for both transformer models in binary classification.



Accuracy Plot



Loss plot

Figure 10: Training Progress of Transformer Models (Binary Classification): The plot illustrates the training dynamics of XLM-Roberta alongside BERT. Notably, XLM-Roberta demonstrates superior performance, consistently outperforming BERT in accuracy throughout the training process. This superiority is reflected in its steady increase in accuracy over time, ultimately surpassing BERT's performance.

Table 6.1 shows that among the transformer models, BanglaBERT showcased commendable accuracy, outperforming both ML and DL models. The accuracy of BanglaBERT was approximately 13.4% higher than the LSTM model, emphasizing the effectiveness of transformer-based architectures in capturing complex contextual information for Bangla comments. Additionally, BanglaBERT exhibited a macro precision approximately 13.2%

higher than LSTM and a macro recall approximately 18.9% higher than LSTM, highlighting its robustness in handling both precision and recall aspects.

Similarly, XLM-RoBERTa, another transformer-based model, displayed remarkable accuracy, surpassing the LSTM model by approximately 12.6%. The macro precision of XLM-RoBERTa was approximately 11.0% higher than LSTM, and the macro recall was approximately 12.7% higher than LSTM, underscoring the consistent superiority of transformer models in achieving higher accuracy and capturing nuances within Bangla comments.

The transformer models' proficiency in handling complex contextual information, coupled with their ability to learn intricate patterns, positions them as compelling choices for binary classification tasks.

6.1.4 Graph Based Model:

In the training phase, BanglaBERTGCN model exhibits consistent improvements in accuracy, showing significant gains from the first to the second epoch and maintaining a strong upward trend into the third epoch highlighting the model's effectiveness and robust learning capabilities.

Besides, BanglaBertGCN stands out as the clear champion, surpassing traditional machine learning and deep learning models. Its superiority is evident through significant performance improvements compared to the top-performing models Naive Bayes(ML model), CNN+LSTM (DL model), and BanglaBanglaBERT (Transformer model) for BaTeClaCor as shown in Table 6.1.

BanglaBertGCN vs. TF-IDF + Naive Bayes:

- **Accuracy:** BanglaBertGCN demonstrates a significant 17.16% accuracy advantage, highlighting its superior ability to correctly categorize text samples. While Naive Bayes offers simplicity and interpretability, BanglaBertGCN's transformer architecture excels at capturing complex relationships within text, resulting in more accurate classifications.

- **Precision** Both models exhibit respectable precision, with BanglaBertGCN leading by 5.56%. This indicates their effectiveness in identifying true positives, and BanglaBertGCN's contextual understanding provides a slight edge.

- **Recall:** BanglaBertGCN shines in recall, boasting a remarkable 43.77% improvement over Naive Bayes. This means BanglaBertGCN misses significantly fewer relevant positive examples, leveraging its ability to consider long-range dependencies and complex word relationships often overlooked by Naive Bayes.

BanglaBertGCN vs CNN+LSTM:

- **Accuracy:** BanglaBertGCN maintains a notable 7.64% accuracy advantage, showcasing its superior categorization ability. The powerful transformer architecture, trained on extensive data, allows BanglaBertGCN to capture complex contextual relationships within the text, overcoming the challenges faced by traditional deep learning models like CNN+LSTM.

- **Precision:** Both models display commendable precision, with BanglaBertGCN slightly ahead by 0.07%. Both effectively identify true positives, but BanglaBertGCN's contextual understanding provides a slight edge.

- **Recall:** BanglaBertGCN excels in recall, boasting a significant 22.58% improvement over CNN+LSTM. This indicates that BanglaBertGCN misses significantly fewer relevant positive examples, thanks to its incorporation of graph structures, enabling effective modeling of relationships within the text—an ability absent in CNN+LSTM.

BanglaBertGCN vs.BanglaBert:

- **Accuracy:** BanglaBertGCN maintains supremacy with a 3.99% accuracy lead, showcasing its superior contextual understanding for more accurate classifications. While BanglaBert performs well, BanglaBertGCN's ability to consider global relationships within text gives it a competitive edge.

- **Precision:** Both models closely match in precision, with BanglaBert trailing by only 0.07%. This reflects their effectiveness in identifying true positives, but BanglaBertGCN's recall advantage makes it the standout performer.

- **Recall:** BanglaBertGCN distinguishes itself with a commanding 9.88% lead in recall, capturing considerably fewer relevant positive examples than BanglaBert. This advantage likely stems from BanglaBertGCN's use of graph structures, enabling effective modeling of relationships between entities—an aspect lacking in BanglaBert.

6.2 Multiclass Classification

Expanding upon our exploration, we extend the analysis to tackle the more intricate realm of multiclass classification. We leverage the same array of machine learning and deep learning models as listed in the Binary Classification section.

6.2.1 Machine Learning Models:

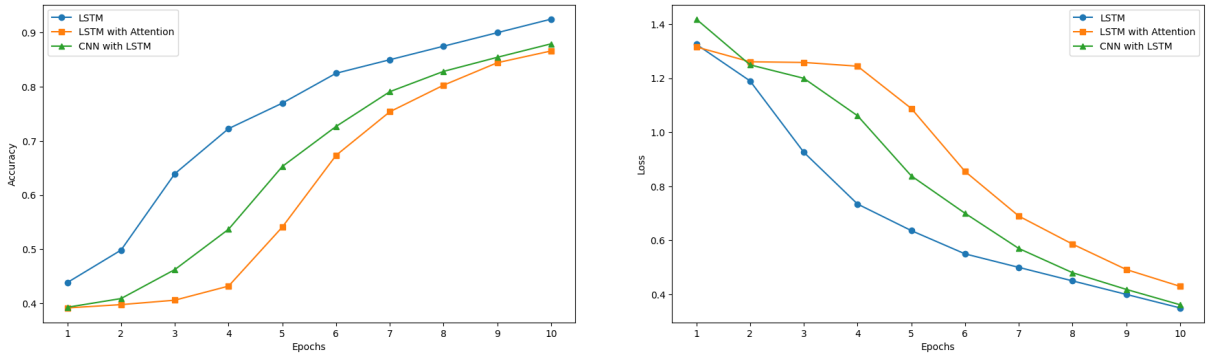
The machine learning models exhibited distinct performances, with notable differences in accuracy, macro precision, and macro recall multiclass classification. The XGBoost model emerged as the top performer among the considered models, demonstrating an accuracy approximately 13.0% higher than the SVM model. This substantial improvement suggests that the ensemble learning capabilities of XGBoost contribute significantly to enhancing the classification of errors in Bangla comments. Additionally, XGBoost exhibited a macro precision approximately 19.6% higher than SVM and a macro recall approximately 26.8% higher than SVM, emphasizing its superiority in achieving higher precision and recall.

Random Forest, despite its capacity for handling complex relationships within data, showcased lower accuracy than XGBoost, but still surpassed SVM. The accuracy of Random Forest was approximately 3.0% higher than that of SVM, affirming its competency in Bangla error classification. The macro precision of Random Forest was approximately 6.3% higher than SVM, and the macro recall was approximately 8.5% higher than SVM, underscoring its ability to capture nuanced patterns within the data.

Naive Bayes, on the other hand, displayed comparatively lower accuracy than both XGBoost and Random Forest. The accuracy of Naive Bayes was approximately 14.0% lower than XGBoost and approximately 1.8% lower than SVM. Despite this, Naive Bayes demonstrated a macro precision approximately 7.6% higher than SVM and a macro recall approximately 0.6% higher than SVM, suggesting its potential for achieving higher precision in multiclass error classification scenarios.

6.2.2 Deep Learning Models:

In the training phase shown in figure 11, LSTM models show varying improvements from 3% to 6%. CNN with LSTM models demonstrates accuracy improvements ranging from 6% to 10% over the 10 epochs, highlighting their ability to capture nuanced patterns and adapt effectively. Besides, the CNN with LSTM model exhibits consistent declines in loss, with improvements ranging from approximately 14% to 27%. LSTM with attention mechanism demonstrates more substantial decreases, showing percentage improvements from around 32% to 55%.



Accuracy Plot Loss Plot

Figure 11: Training Progress of Deep Learning Models (Multiclass Classification): The visualization depicts the training progress of LSTM alongside other models. Notably, LSTM emerges as the top performer, consistently outshining the other models in terms of accuracy. Moreover, LSTM exhibits a more substantial decrease in loss compared to the other models, indicating its superior convergence during training.

From Table 6.1, It is evident that the deep learning models displayed diverse performances, with LSTM emerging as the most effective among the considered models. The LSTM model achieved an accuracy that was approximately 3.1% higher than LSTM with Attention and approximately 4.1% higher than LSTM + CNN. Notably, LSTM demonstrated a macro precision approximately 4.2% higher than LSTM with Attention and approximately 4.8% higher than LSTM + CNN, underscoring its ability to achieve higher precision in classifying errors in Bangla comments. Additionally, LSTM showcased a macro recall approximately 4.0% higher than LSTM with Attention and approximately 2.1% higher than LSTM + CNN, emphasizing its proficiency in capturing relevant information across all classes.

LSTM with Attention, while exhibiting slightly lower accuracy and macro recall compared to LSTM, demonstrated a macro precision that was approximately 1.3% lower than LSTM. The introduction of attention mechanisms in LSTM with Attention was expected to enhance the model's ability to capture intricate relationships within the input sequence. However, the results suggest that, for this specific task, the additional complexity might not have provided substantial benefits.

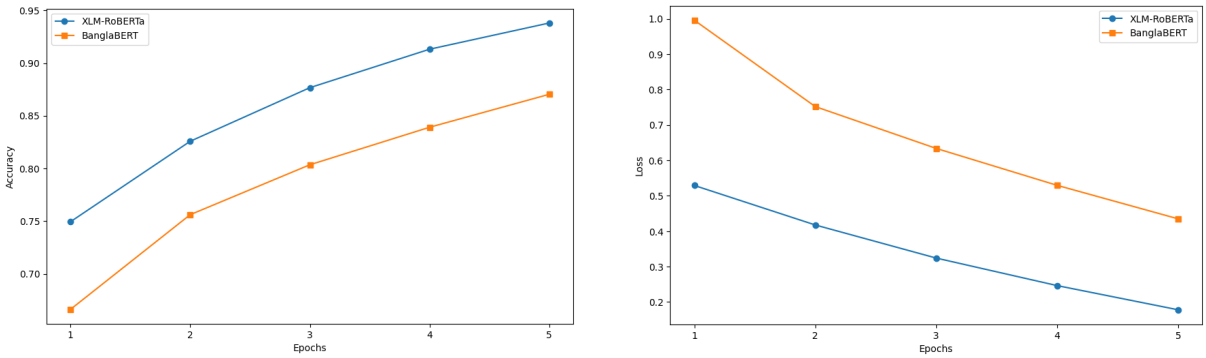
LSTM + CNN, despite incorporating convolutional layers for spatial hierarchies, showcased the lowest accuracy among the considered models. The accuracy of LSTM + CNN was approximately 4.1% lower than LSTM and approximately 1.1% lower than LSTM with Attention. This result suggests that the additional convolutional layers might not have contributed significantly to improving the model's performance in multiclass classification scenarios.

The standalone LSTM model outperformed LSTM with Attention and LSTM + CNN. Its higher accuracy, macro precision, and macro recall highlight its effectiveness in han-

dling the complexities of multiclass error detection in the context of Bangla YouTube comments. The choice of the most suitable model depends on the specific priorities of the classification task, considering factors such as precision, recall, and overall accuracy.

6.2.3 Transformer Models:

In the training phase, XLM-RoBERTa achieves accuracy gains of around 1% to 2%, while BanglaBERT shows more substantial improvements, ranging from 9% to 11%, emphasizing its effectiveness in capturing intricate patterns across various classes. Besides, both models demonstrate reductions in loss, with XLM-RoBERTa showing improvements from around 38% to 66%, and BanglaBERT ranging from approximately 56% to 69%. This indicates effective loss minimization across epochs for both transformer models, as shown in figure 12.



Accuracy Plot Loss plot

Figure 12: Training Progress of Transformer Models (Multiclass Classification): The plot illustrates the training dynamics of XLM-Roberta alongside BERT. Notably, XLM-Roberta demonstrates superior performance, consistently outperforming BERT in terms of accuracy throughout the training process. This superiority is reflected in its steady increase in accuracy over time, ultimately surpassing BERT's performance.

Table 6.1 shows that once again, transformer models, BanglaBERT and XLM-RoBERTa, outshined their machine learning and deep learning counterparts. BanglaBERT, in particular, demonstrated superior performance, achieving an accuracy approximately 21.6% higher than the best-performing deep learning model, LSTM. The macro precision of BanglaBERT was approximately 7.2% higher than LSTM, underscoring its ability to provide more accurate and balanced predictions across multiple classes. Moreover, the macro recall of BanglaBERT was approximately 2.2% higher than LSTM, highlighting its proficiency in capturing relevant instances from all classes.

XLM-RoBERTa, while not surpassing BanglaBERT, still showcased remarkable performance, outperforming all considered machine learning and deep learning models. Its accuracy was approximately 19.6% higher than LSTM and approximately 13% higher than LSTM + CNN. The macro precision of XLM-RoBERTa was approximately 7% higher than LSTM, suggesting its enhanced ability to make precise predictions across diverse classes. Additionally, XLM-RoBERTa demonstrated a macro recall approximately 0.4% higher than LSTM, emphasizing its efficacy in capturing instances from all classes.

The exceptional performance of transformer models, particularly BanglaBERT, underscores their capability to leverage contextual information and linguistic nuances effectively. These models, pretrained on extensive corpora, can capture intricate patterns in Bangla YouTube comments, making them powerful tools for multiclass error identification. The

choice between BanglaBERT and XLM-RoBERTa can be based on specific task priorities, considering factors such as precision, recall, and overall accuracy.

6.2.4 Graph Based Model:

In the training phase, BanglaBERTGCN shows noticeable percentage improvements from the first to the second epoch and continued advancement into the third epoch, highlighting the model's effectiveness and robust learning capabilities. Besides, the loss grew by about 15.97% in the last epoch. These trends highlight potential challenges in convergence and may prompt further investigation into optimizing the model's training dynamics.

Once again, BanglaBertGCN stands out as the clear champion, surpassing traditional machine learning and deep learning models. Its superiority is evident through significant performance improvements compared to the top-performing models TF-IDF + XGBoost (ML model), LSTM (DL model), and BanglaBERT (Transformer model) for BaTeClaCor as shown in Table 6.1.

BanglaBertGCN vs. TF-IDF + XGBoost:

- **Accuracy:** BanglaBertGCN demonstrates a substantial 12.33% lead, underscoring its adept handling of multi-class problems. This notable advantage is attributed to BanglaBertGCN's powerful transformer architecture, allowing for a more nuanced understanding of complex relationships within text—crucial for accurate multi-class classification. In contrast, TF-idf + XGBoost lacks this sophisticated architecture, impacting its accuracy in diverse categorization tasks

- **Precision:** BanglaBertGCN takes the lead by exceeding TF-idf + XGBoost precision by 7.67%. This significant margin is influenced by BanglaBertGCN's attention mechanism, allowing it to focus on relevant information for each class. TF-idf + XGBoost lacks this mechanism, contributing to its comparatively lower precision in identifying true positives for various categories.

- **Recall:** BanglaBertGCN outperforms TF-idf + XGBoost by an impressive 28.63% in recall. This substantial difference is attributed to BanglaBertGCN's incorporation of graph structures, enabling it to model relationships between entities and words. TF-idf + XGBoost lacks this capability, affecting its ability to comprehensively recall relevant positive examples across all classes.

BanglaBertGCN vs LSTM:

- **Accuracy:** BanglaBertGCN holds a substantial 16.19% advantage, showcasing its superior capability in multi-class classification tasks. This advantage is rooted in BanglaBertGCN's powerful transformer architecture and its ability to capture both local and global contextualized information effectively. In contrast, LSTMs, while adept at sequence modeling, can struggle with the intricacies of multiple categories.

- **Precision:** BanglaBertGCN leads by 17.05%, showcasing its improved ability to correctly assign each class label. This precision is influenced by BanglaBertGCN's transductive learning approach, allowing it to generalize effectively from a large pretraining dataset. LSTMs may lack this advantage, impacting their precision in handling diverse categories.

- **Recall:** BanglaBertGCN outperforms LSTM by an impressive 33.89%, indicating its remarkable ability to capture all relevant positive examples across all classes. This comprehensive recall is linked to BanglaBertGCN's attention mechanism and incorporation of graph structures, which facilitate a better understanding of relationships between entities and words. LSTMs may struggle with these aspects, impacting their recall in the multi-class scenario.

BanglaBertGCN vs.BanglaBert:

- **Accuracy:** BanglaBertGCN edges out Banglabert by 0.67%, suggesting its slightly finer handling of multi-class complexities. This marginal lead can be attributed to BanglaBertGCN's transductive learning and pretrained approach, enabling it to adapt more effectively to diverse classification tasks. Banglabert may lack this adaptability, affecting its accuracy in a variety of categories.

- **Precision:** Banglabert takes a lead by 2.19%, suggesting sharper precision for specific classes. However, it's essential to consider the overall performance metrics where BanglaBertGCN excels. Banglabert's precision advantage may be specific to certain categories, while BanglaBertGCN's comprehensive approach allows it to maintain a competitive edge across diverse classes.

- **Recall:** BanglaBertGCN reclaims the crown with a significant 21.80% advantage in recall. This superiority is tied to BanglaBertGCN's powerful transformer architecture, attention mechanism, and incorporation of graph structures. These factors enable it to comprehensively capture relevant examples from all classes, aspects that Banglabert may lack, impacting its recall in diverse categorization scenarios.

Throughout the comparisons, the unique features of BanglaBertGCN, including its powerful transformer architecture, attention mechanism, incorporation of graph structures, and transductive learning, contribute to its superior performance in accuracy, precision, and recall across diverse multi-class classification tasks.

6.3 Error Corrector Model

The performances of the error corrector models are reported in Table 6.2 where BanglaT5 and BanglaT5-Small were experimented with. These two transformer-based models, despite sharing a common linguistic foundation, exhibit discernible variations in their performance metrics. Analyzing the ROUGE scores, it becomes evident that BanglaT5 surpasses BanglaT5 small in several key aspects. From the top 5 best-predicted outputs of BanglaT5 in Figure 13, we see that common single-word errors were predicated properly which is indicated by the ROUGE-L score of 1.0. For ROUGE-1 F1, BanglaT5 outperforms BanglaT5 small by approximately 7.24%, showcasing a superior ability to capture unigram precision and recall. The margin widens when considering ROUGE-2 F1, where BanglaT5 exhibits a substantial improvement of 20.11% over BanglaT5 small, emphasizing its proficiency in capturing bigram-level linguistic nuances. Additionally, for ROUGE-L F1, BanglaT5 demonstrates a 7.41% enhancement, signifying its robustness in preserving the longest common subsequence between the reference and generated summaries. These percentage differences underscore BanglaT5's prowess in generating more accurate, cohesive, and contextually relevant summaries compared to its smaller counterparts.

BanglaT5 small			
ROUGE Metric	Precision	Recall	F1
ROUGE-1	0.8357	0.8343	0.8343
ROUGE-2	0.7073	0.7052	0.4246
ROUGE-L	0.8361	0.8342	0.8344
Total Inference Time	**108.00 seconds**		
Average Inference Time	**0.1281 seconds**		

BanglaT5			
ROUGE Metric	Precision	Recall	F1
ROUGE-1	0.8979	0.8925	**0.8947**
ROUGE-2	0.8182	0.8126	**0.5100**
ROUGE-L	0.8993	0.8940	**0.8962**
Total Inference Time	1093.16 seconds		
Average Inference Time	0.3515 seconds		

Table 6.2: Error Correction Performance of BanglaT5 Small and BanglaT5. The table presents the performance metrics of two variants of BanglaT5 (BanglaT5 Small and BanglaT5) in error correction tasks. Evaluation is conducted using ROUGE metrics, precision, recall, and F1 score. Notably, BanglaT5 achieves higher ROUGE scores across all metrics compared to BanglaT5 Small, indicating its superior performance in error correction. Additionally, the table includes total and average inference times for both models, showcasing their computational efficiency.

Shifting the focus to inference times, the efficiency of BanglaT5 small becomes apparent. It outpaces BanglaT5 significantly in terms of speed, being approximately 174.40% faster in average inference time. This remarkable efficiency positions BanglaT5 small as an optimal choice for scenarios where computational resources are a critical consideration. The faster inference times of BanglaT5 small can be attributed to its streamlined architecture and optimized processing, allowing for swift execution of natural language processing tasks. While BanglaT5 excels in capturing intricate linguistic nuances, BanglaT5 small showcases commendable efficiency in terms of inference times. The choice between these models depends on the specific requirements of a task, considering factors such as the

need for high-quality linguistic analysis or resource-efficient processing in the context of natural language processing applications.

6.4 Error Analysis of Baseline Models

In the error analysis of our binary classification models, we identified certain limitations impacting their performance. The machine learning model, utilizing TF-IDF with other classifiers but lacking a sophisticated representation of semantic relationships, exhibited results that fell slightly short of the remarkably high standards achieved by some other models. The reliance on TF-IDF alone made the model struggle to capture the intricate nuances of errors dependent on contextual semantics, leading to occasional misclassifications. Besides, the deep learning models faced challenges related to limited model interpretability, making it challenging to understand and address specific issues affecting their predictions. Despite the overall high performance of BanglaBERT and XLM-Roberta, there was a nuanced disparity in achievement, as their performance, though excellent (above 89%), did not reach the exceptionally elevated levels of accuracy achieved by some other models. In multiclass classification, the observed performance variation, with models performing approximately 5-15% worse than binary classification, can be attributed to the imbalanced distribution of error types in the dataset.

Comment	Predicted	Ground Truth	Rouge-L
নাটক টি সত্যি মন ছুয়ে গেছে❤😢😢😢😢	নাটক টি সত্যি মন ছুঁয়ে গেছে❤😢😢😢😢	নাটক টি সত্যি মন ছুঁয়ে গেছে❤😢😢😢😢	1.0
ভাইরে যেটা খামু সেখানে ভেজাল	ভাইরে যেটা খাবো সেখানে ভেজাল	ভাইরে যেটা খাবো সেখানে ভেজাল	1.0
হাসতে হাসতে পেট ব্যাথা হয়ে গেছে 😂😂	হাসতে হাসতে পেট ব্যথা হয়ে গেছে 😂😂	হাসতে হাসতে পেট ব্যথা হয়ে গেছে 😂😂	1.0
বিরিয়ানির সিন গুলা খুব হাসকর,😁😁😂😂	বিরিয়ানির দৃশ্য গুলা খুব হাস্যকর,😁😁😂😂	বিরিয়ানির দৃশ্য গুলা খুব হাস্যকর,😁😁😂😂	1.0
সুস্ট বিচারের দাবী জানাচ্ছি 😢😢	সুষ্ঠ বিচারের দাবী জানাচ্ছি 😢😢	সুষ্ঠ বিচারের দাবী জানাচ্ছি 😢😢	1.0

Figure 13: Top 5 Best Predicted Outputs. The figure displays a selection of predicted outputs generated by the model. Each prediction is compared against the ground truth. These examples provide insight into the model's ability to accurately interpret and generate text in the given context. The accompanying 1.0 score indicates perfect match between the ground truth and predicted outputs.

The imbalanced distribution posed a challenge for the models to equally represent and identify various error categories, leading to a decrease in overall performance. In the binary classification models, focused on distinguishing between correct and incorrect samples, may have achieved higher accuracy by primarily focusing on the majority class of spelling errors. This imbalance underscores the importance of carefully considering dataset composition and class distribution, particularly in scenarios where certain error types are more prevalent than others, and may have dominated the learning process.

Comment	Predicted	Ground Truth	Rouge-L
একেই বলে ফ্রী এন্ড ফেয়ার সিলেকশন 😄🤣😁।	একেই বলে ফ্রী এন্ড ফেয়ার সিলেকশন 😄🤣😁।	একেই বলে মুক্ত এবং সুষ্ঠ নির্বাচন 😄🤣😁।	0.53
ধনোডাদ এত ভালো ভাইডিও ডিওয়ার জইনও	ধনোডা এত ভালো ভাইডিও দেখার দরকার	ধন্যবাদ এত ভালো ভিডিও দেওয়ার জন্য	0.66
একজন জোকারি আরেক জোকারের মর্ম বোঝে	একজন জোকারি আরেক জোকারের মর্ম বোঝে	একজন ঠাট্টবাজই আরেক ঠাট্টবাজের মর্ম বোঝে	0.67
ভাই মোভি টার নাম কি	ভাই মোভি টার নাম কি	ভাই চলচ্চিত্র টার নাম কি	0.72
টেষ্ট করে দেখতে হবে দেখছি	টেষ্ট করে দেখতে হবে দেখছি	স্বাদ নিয়ে দেখতে হবে দেখছি	0.73

Figure 14: Top 5 Worst Predicted Outputs. The figure illustrates a collection of predicted outputs generated by the model. Each prediction is contrasted against the ground truth. The accompanying scores may indicate the degree of mismatch between the ground truth and predicted outputs, providing valuable feedback for model evaluation and development.

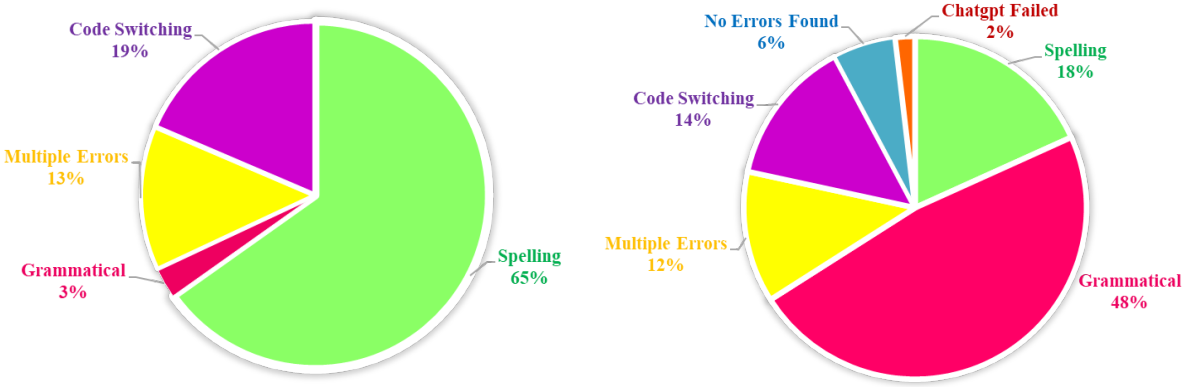
For the same reason, the BanglaT5 model predicted the correct output of the incorrect comments mostly from the Spelling error category as shown in figure 13. This category had the majority of samples compared to code-switching, grammatical, or multiple error categories. This imbalance manifested in lower Rouge scores for several samples in the corrector models. As shown in figure 14, the range of ROUGE-L score lies between 0.53 and 0.73 for the worst 5 predicted outputs.

6.5 Can Chatgpt Detect and Correct Bangla Text Efficiently?

In recent times, the advancements in large language models (LLMs), exemplified by ChatGPT, have been remarkable, showcasing heightened capabilities in handling diverse linguistic tasks, including error correction within multilingual contexts. The iterative updates and refinements to models like ChatGPT underscore a trajectory of enhanced proficiency in rectifying errors across a spectrum of languages. This evolving landscape suggests a promising avenue for LLMs, demonstrating their increasing aptitude for refining and rectifying errors in diverse multilingual text scenarios.

In light of these developments, our focus shifts to investigating the performance of ChatGPT, a prominent LLM, specifically in the realm of Bangla text error correction. This exploration aims to provide insights into the adaptability and efficacy of such models when applied to the unique linguistic nuances and challenges present in Bangla language data. Out of our test samples, we specifically selected 3,000 errorful comments where each error category was allocated 20% of the total samples denoted in table 3.2, ensuring a balanced representation across categories for evaluating the predictions of both the BanglaBERTGCN and ChatGPT models in error classification, BanglaT5 and ChatGPT models in error correction. This investigation endeavors to evaluate the effectiveness of the ChatGPT-3.5-turbo model in addressing error classification and correction challenges within the Bangla language. Through the utilization of a procured API, we systematically extracted errorful sentences from our designated test dataset, subjecting them to comprehensive evaluation alongside outcomes from our indigenous BanglaBERTGCN and BanglaT5 model.

6.5.1 Error Classification



Ground Truth

Predicted by ChatGPT

Figure 15: Category Distribution: The figure presents the distribution of categories in the ground truth data compared to the category distribution generated by ChatGPT. The comparison offers insights into ChatGPT's performance in categorizing data across different categories, highlighting areas where the model may exhibit biases or discrepancies compared to the ground truth

After thorough team discussion and considering different prompt designs, it was decided to consider the following zero-shot prompt for error classification:

> "You are a Bangla text error detection tool that can identify textual errors in a Bangla text out of the 4 categories which are Spelling, Grammatical, Code Switching, and Multiple Errors. Code-switching means the mix of Bangla and English words written in Bangla letters. Multiple errors mean if more than one category of error is present among the other 3 categories mentioned. You must identify the error category among the 4 categories in the Bangla sentence. The Bangla sentence is: "Error Sentence". You need to comprehend the sentence as a whole before identifying any errors. Keep your response limited to only the Error category within the tag <c> Your predicted category </c>. Please follow the format of the output with the tags."

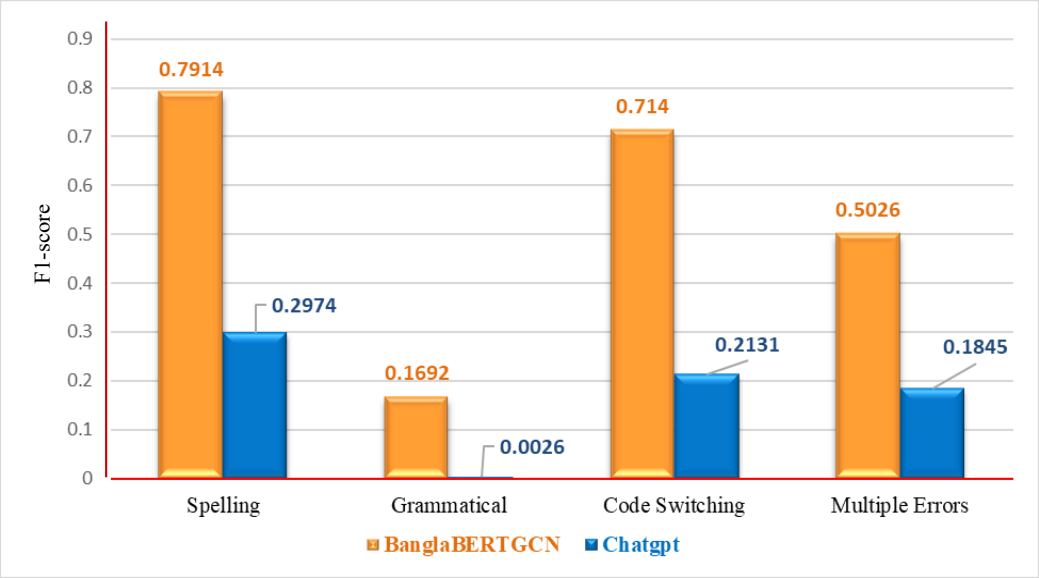


Figure 16: F1 Score Comparison of Multiclass Classification (BanglaBERTGCN vs. ChatGPT): The figure presents the F1 scores achieved by BanglaBERTGCN and ChatGPT in multiclass classification across different error categories. This comparison highlights the relative strengths and weaknesses of BanglaBERTGCN and ChatGPT in accurately classifying errors in Bangla comments.

Figure 15 displays the labeled categories considered as ground truth, alongside those predicted by ChatGPT. Despite ChatGPT's prevalent prediction of grammatical errors across the majority of samples, its accuracy in comparison to the ground truth was notably lacking. Besides, for approximately 6% of the errorful samples, ChatGPT predicted the samples to have no errors despite their containing errors, indicating its inability to detect or comprehend certain error types. Furthermore, in approximately 2% of the errorful samples, ChatGPT encountered difficulty in predicting any error category, suggesting limitations in its error categorization capabilities.

In error classification in Bangla comments, a closer examination illuminates the factors contributing to BanglaBERTGCN's supremacy. Notably, BanglaBERTGCN's exceptional performance can be attributed to its comprehensive training by BaTEClaCor dataset. From table 16, BanglaBERTGCN demonstrates superiority in spelling error detection, scoring 63.29% higher F1 than ChatGPT, likely due to extensive training on Bangla text. BanglaBERTGCN performed better than ChatGPT in understanding

Bangla grammar, scoring 99.74% higher in identifying grammatical errors compared to ChatGPT's significantly lower score of 0.0026. BanglaBERTGCN excels in code-switching detection with a 70.42% higher F1 score, leveraging its bilingual proficiency. In contrast, ChatGPT's limitations in bilingual understanding and pattern recognition hinder its performance. BanglaBERTGCN also shows proficiency in handling texts with multiple errors, achieving a remarkable 92.56% higher F1 score. Conversely, ChatGPT struggles in such scenarios due to its lack of advanced error recognition mechanisms. Despite ChatGPT's general-purpose approach, its inadequacies in Bangla-specific training and advanced error recognition significantly impair its performance compared to BanglaBERTGCN.

6.5.2 Error Correction

After thorough team deliberation and exploration of various prompt designs, it was determined to utilize the following zero-shot prompt for error correction:

The following prompt was considered:

> "You are a Bangla text error correction tool that can correct textual errors in a Bangla text. You must correct any textual errors in the Bangla sentence. The Bangla sentence is: "Error Sentence". You need to comprehend the sentence as a whole before identifying and correcting any errors step by step while keeping the original sentence structure unchanged as much as possible. Keep your response limited to only your corrected output results within the tag <output> Your Corrected Version </output>. Please follow the format of the output and their tags."

ChatGPT may face challenges in accurately correcting errors in sentences that involve "Banglish" or code-switching, a linguistic phenomenon where speakers switch between two or more languages, such as Bangla and English. The model may struggle with Bangla text in zero-shot prompting, which refers to providing the model with tasks or prompts without specific training in that language. The difficulties arise from the fact that ChatGPT has primarily been trained in English text and may not possess a robust understanding of Bangla. As a result, when attempting to correct errors or predict outcomes in Bangla sentences, the model might provide suggestions with changed spellings and incorrect words, leading to altered meanings. These challenges stem from the inherent limitations in the model's training data, which may not adequately cover the nuances of Bangla language structures and variations. Additionally, the model's lack of explicit training in Bangla can contribute to inaccuracies in handling tasks related to 'Banglish' or code-switching.

Chatgpt			
ROUGE Metric	Precision	Recall	F1
ROUGE-1	0.7197	0.7144	0.7157
ROUGE-2	0.5340	0.5340	0.5312
ROUGE-L	0.7171	0.7119	0.7132

Table 6.4: Error Correction Performance of ChatGPT. The table displays the error correction performance of ChatGPT, evaluated using ROUGE metrics including precision, recall, and F1 score for different n-gram orders (ROUGE-1, ROUGE-2) and the longest common subsequence (ROUGE-L). The scores indicate ChatGPT's performance in correcting errors in Bangla comments.

	Comment	Rouge-L
Original	কি বলবো সত্যি অসাধারণ রান্না আমি অসাধারণ রান্না আমি আজকেই ট্রাই করবো❤	-
Ground Truth	কি বলবো সত্যি অসাধারণ রান্না আমি অসাধারণ রান্না আমি আজকেই চেষ্টা করবো❤	-
ChatGPT Prediction	কি বলবো সত্যি অসাধারণ রান্না আমি অসাধারণ রান্না আমি আজকেই ট্রাই করবো❤	0.77
BanglaT5 Prediction	কি বলবো সত্যি অসাধারণ রান্না আমি অসাধারণ রান্না আমি আজকেই চেষ্টা করবো❤	1
Original	বউ এর বান্ধবীর কাছে ফোন দেওয়ার পর হাসতে হাসতে পেট ব্যাথা হয়ে গেছে।	-
Ground Truth	বউ এর বান্ধবীর কাছে ফোন দেওয়ার পর হাসতে হাসতে পেট ব্যথা হয়ে গেছে।	-
ChatGPT Prediction	বউর বন্ধু কোল করার পরে হাসাটা হাসাটা পেট ব্যাথা হয়ে গেল।	0.69
BanglaT5 Prediction	বউ এর বান্ধবীর কাছে ফোন দেওয়ার পর হাসতে হাসতে পেট ব্যথা হয়ে গেছে।	1
Original	আমি সুযোগ পেলেই এই ছবিগুলো দেখি।বিশ্বাস করেন আমার শরীররের প্রত্যেকটি লোম দারিয়ে যায়।	-
Ground Truth	আমি সুযোগ পেলেই এই ছবিগুলো দেখি।বিশ্বাস করেন আমার শরীরের প্রত্যেকটি লোম দাঁড়িয়ে যায়।	-
ChatGPT Prediction	আমি সুবিধা পেলেই এই ছবিগুলো দেখি| বিশ্বাস করেন যে আমার শরীরের প্রত্যেকটি লম্বা দাড়িয়ে যায়।	0.64
BanglaT5 Prediction	আমি সুযোগ পেলেই এই ছবিগুলো দেখি।বিশ্বাস করেন আমার শরীরের প্রত্যেকটি লোম দাঁড়িয়ে যায়।	1
Original	মুন্সিগঞ্জের ছেলে হয়ে গর্ভিত আমি😊	-
Ground Truth	মুন্সিগঞ্জের ছেলে হয়ে গর্বিত আমি😊	-
ChatGPT Prediction	মুন্সিগঞ্জের ছেলে হয়ে গর্ভধারিতা আমি😊	0.83
BanglaT5 Prediction	মুন্সিগঞ্জের ছেলে হয়ে গর্বিত আমি😊	1
Original	আসল দৃশ্য দেখালেন না। তখন না মানুষ বুজত জীন কি জিনিস	-
Ground Truth	আসল দৃশ্য দেখালেন না। তখন না মানুষ বুঝতো জীন কি জিনিস	-
ChatGPT Prediction	আসল দৃশ্য দেখানো হলো না তখন কেউই বুঝেনি কী জিনিস জীন	0.36
BanglaT5 Prediction	আসল দৃশ্য দেখালেন না। তখন না মানুষ বুঝত জীন কি জিনিস	0.99

Table 6.3: Error Correction Performance of ChatGPT and BanglaT5. The table presents the error correction performance comparison between ChatGPT and BanglaT5, assessed using ROUGE-L metric scores. Each row corresponds to a specific input text, providing the original text, ground truth corrected text, ChatGPT's prediction, BanglaT5's prediction, and their respective ROUGE-L scores.

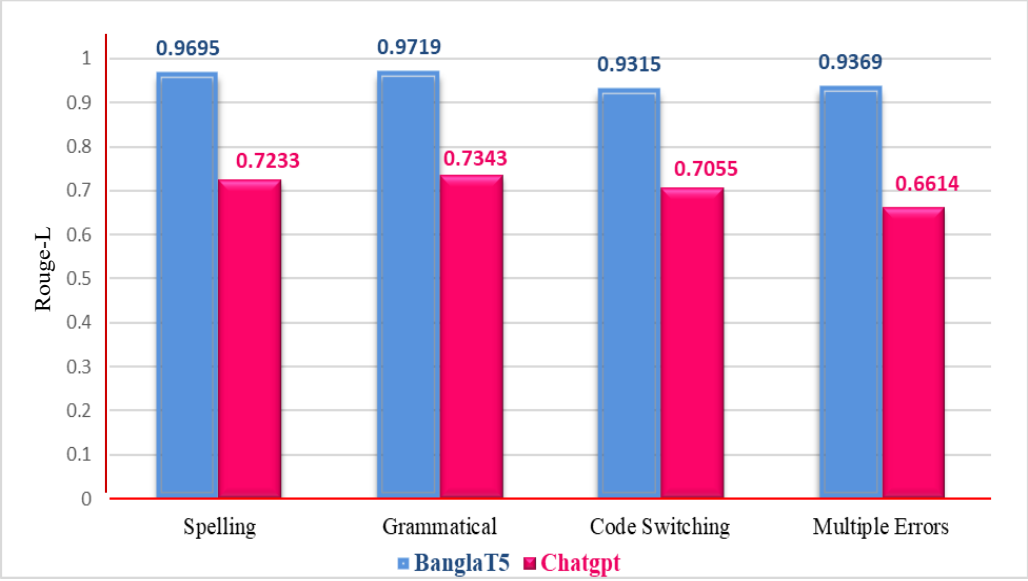


Figure 17: Error Correction Performance Comparison (BanglaT5 vs. ChatGPT) based on ROUGE-L Score. The figure depicts the performance comparison between BanglaT5 and ChatGPT in error correction in Bangla comments, evaluated using the ROUGE-L metric score. Each model's ability to classify text data across multiple classes is assessed, with higher ROUGE-L scores indicating better alignment with the ground truth.

Table 6.4 shows the Rouge scores yielded by ChatGPT in error correction for the 3000 errorful comments. For a more comparative analysis, we investigated Chatgpt's performance with the trained BanglaT5 model's performance by the Rouge-L F1 scores for every error category. It is evident from figure 17 that BanglaT5 asserts its superiority over the general-purpose LLM ChatGPT across all error categories. This dominance stems from the synergy of BanglaT5's inherent capabilities and the focused training provided by our BaTEClaCor dataset.

In spelling error correction, BanglaT5 excels, leveraging its exposure to diverse writing styles during training to grasp subtle spelling rules and detect deviations in Bangla text with remarkable accuracy, achieving a 27.09% higher Rouge-L score than ChatGPT. Moreover, in handling code-switching, BanglaT5 inherent bilingual proficiency and potential exposure to code-mixed content in the BaTEClaCor dataset, culminating in a commanding 40.53% higher Rouge-L score compared to ChatGPT, which grapples with its limitations in bilingual comprehension and specialized training. Though the rouge-L score of BanglaT5 in grammatical error correction is low due to lesser samples present for this error category in the BaTEClaCor dataset as mentioned in section 6.4, it still surpasses ChatGPT's performance by 24.48% higher Rouge-L score.

BanglaT5 demonstrates proficiency in addressing texts fraught with multiple errors, leveraging its ability to analyze sentence structure and word relationships to identify and rectify various error types simultaneously, resulting in a 25.18% higher Rouge-L score compared to ChatGPT. Nonetheless, it's imperative to acknowledge ChatGPT's shortcomings in this domain, attributed to its lack of specific Bangla language training, hindering its grasp of Bangla grammar, spelling, and code-switching nuances. Additionally, the absence of targeted training on datasets like BaTEClaCor impedes its acquisition of domain-specific expertise, further widening the performance gap.

Our findings indicate that the BanglaT5 model outperformed the ChatGPT model by **29.32%**. This observation underscores the contrast in performance between ChatGPT across languages, excelling notably in English but displaying suboptimal results in languages like Bangla where resources are limited. Consequently, our dataset emerges

as a valuable resource in this context. Furthermore, we posit that error correction models trained on language-specific datasets exhibit superior performance compared to those trained on generic datasets.

7 Conclusion and Future Works

7.1 Conclusion

In this extensive study, we embarked on a comprehensive journey to tackle the formidable challenge of Bangla text correction, employing a sophisticated amalgamation of traditional machine learning, deep learning techniques, and Transformer models. A defining milestone in our research was the meticulous creation and annotation of a novel dataset derived from YouTube comments. This dataset, meticulously curated to capture the nuances of Bangla language usage, forms the bedrock of our investigative endeavors.

Our approach involved a thorough evaluation of a diverse array of machine learning models, deep learning models, and Transformer models for both binary and multiclass error-category classification. Notably, the exceptional performance of BanglaBertGCN and BanglaBERT stood out, underscoring their adeptness in navigating the intricate semantics of the Bangla language. Furthermore, the experimental results shed light on the promising potential of BanglaT5 in enhancing the accuracy and robustness of correction systems in the context of Bangla user-generated text.

BanglaBertGCN, after fine-tuning and testing with our meticulously crafted dataset, achieved remarkable accuracies of **85.6%** and **75.8%** for binary and multiclass error classification, respectively. The prowess of BanglaT5 was highlighted through its attainment of the highest Rouge-L score **0.8962** when fine-tuned and tested with our corrected ground truths. Moreover, our experiment with ChatGPT, a widely used Language Model (LLM), reaffirms the superiority of BanglaT5, which demonstrated a RougeL score **29.32%** higher than ChatGPT. This validates the utility of our dataset in effectively enhancing error correction models.

Beyond the numerical achievements, our findings underscore the transformative capabilities of deep learning models and accentuate the pivotal role of dataset curation in the success of language correction systems. The uniqueness of our proposed dataset lies not only in its comprehensive representation of language use in online settings but also in its alignment with the nuanced language patterns of Bangla speakers in digital communication. This dataset thus emerges as a valuable resource, distinct from its predecessors, contributing significantly to the advancement of Bangla text correction research. As we conclude this study, the amalgamation of cutting-edge models, meticulous dataset creation, and insightful findings positions our research at the forefront of endeavors to enhance the linguistic accuracy of Bangla text in digital communication.

7.2 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. 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.

7.3 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.

7.4 Impact On The Society

7.4.1 Sustainability Of The Work

Ensuring the sustainability of this research involves several key considerations. Firstly, ongoing efforts to maintain and expand the dataset, while addressing its limitations, are essential. This includes increasing the dataset size, incorporating more errorful samples, and broadening its scope to encompass variations in Bangla language usage across different regions. Additionally, ethical considerations regarding data collection and usage must be upheld to maintain the integrity of the research and protect the privacy of individuals contributing to the dataset. Collaborative partnerships and knowledge sharing within the research community can also contribute to the long-term sustainability of efforts aimed at advancing Bangla text correction.

7.4.2 Social and Environmental Effects And Analysis

The development and implementation of improved Bangla text correction systems can have both social and environmental effects. Socially, clearer and more accurate communication enhances user experiences in digital spaces, promotes digital inclusivity, and preserves cultural identity. Environmentally, efficient digital communication practices resulting from enhanced language correction systems may lead to reduced energy consumption and resource usage associated with online interactions. However, further analysis is needed to fully understand the environmental implications and optimize sustainability strategies in the context of digital language technologies. Ongoing interdisciplinary research and collaboration can help assess and mitigate any potential negative impacts while maximizing the positive societal and environmental effects of this work.

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 the Bangla language as spoken in different regions.

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. 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.

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.

Impact On The Society

Sustainability Of The Work

Ensuring the sustainability of this research involves several key considerations. Firstly, ongoing efforts to maintain and expand the dataset, while addressing its limitations, are essential. This includes increasing the dataset size, incorporating more errorful samples, and broadening its scope to encompass variations in Bangla language usage across different regions. Additionally, ethical considerations regarding data collection and usage must be upheld to maintain the integrity of the research and protect the privacy of individuals contributing to the dataset. Collaborative partnerships and knowledge sharing within the research community can also contribute to the long-term sustainability of efforts aimed at advancing Bangla text correction.

Social and Environmental Effects And Analysis

The development and implementation of improved Bangla text correction systems can have both social and environmental effects. Socially, clearer and more accurate communication enhances user experiences in digital spaces, promotes digital inclusivity, and preserves cultural identity. Environmentally, efficient digital communication practices resulting from enhanced language correction systems may lead to reduced energy consumption and resource usage associated with online interactions. However, further analysis is needed to fully understand the environmental implications and optimize sustainability strategies in the context of digital language technologies. Ongoing interdisciplinary research and collaboration can help assess and mitigate any potential negative impacts while maximizing the positive societal and environmental effects of this work.

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 the Bangla language as spoken in different regions.

Bibliography

[1] Bayes, T. Naive bayes classifier. *Article Sources and Contributors* (1968), 1–9.

[2] Bhattacharjee, A., Hasan, T., Ahmad, W., Mubasshir, K. S., Islam, M. S., Iqbal, A., Rahman, M. S., and Shahriyar, R. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022* (Seattle, United States, July 2022), Association for Computational Linguistics, pp. 1318–1327.

[3] Bhattacharjee, A., Hasan, T., Ahmad, W., and Shahriyar, R. 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* (2023), pp. 714–723.

[4] Bhattacharjee, A., Hasan, T., Ahmad, W. U., and Shahriyar, R. Banglanlg: Benchmarks and resources for evaluating low-resource natural language generation in bangla. *arXiv preprint arXiv:2205.11081* (2022).

[5] Central Intelligence Agency. CIA World Factbook.

[6] Chen, T., and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (2016), pp. 785–794.

[7] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 8440–8451.

[8] Dadgar, S. M. H., Araghi, M. S., and Farahani, M. M. 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)* (2016), IEEE, pp. 112–116.

[9] DataReportal. Digital 2022: Bangladesh, 2022.

[10] Dehan, F., Fahim, M., Ali, A. A., Amin, M. A., and Rahman, A. Investigating the effectiveness of graph-based algorithm for bangla text classification. In *Proceedings of the First Workshop on Bangla Language Processing (BLP-2023)* (2023), pp. 104–116.

[11] Evgeniou, T., and Pontil, M. Support vector machines: Theory and applications. In *Advanced Course on Artificial Intelligence*. Springer, 1999, pp. 249–257.

[12] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

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

[14] Lin, Y., Meng, Y., Sun, X., Han, Q., Kuang, K., Li, J., and Wu, F. Bertgcn: Transductive text classification by combining gcn and bert. *arXiv preprint arXiv:2105.05727* (2021).

[15] Mittra, T., Nowrin, S., Islam, L., and Roy, D. C. 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)* (2019), IEEE, pp. 1–6.

[16] Qi, Z. 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)* (2020), IEEE, pp. 1241–1246.

[17] Rahman, C. R., Rahman, M., Zakir, S., Rafsan, M., and Ali, M. E. Bspell: A cnn-blended bert based bengali spell checker. *arXiv preprint arXiv:2208.09709* (2022).

[18] Rigatti, S. J. Random forest. *Journal of Insurance Medicine 47*, 1 (2017), 31–39.

[19] Robertson, S. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation 60*, 5 (2004), 503–520.

[20] Shahgir, H., and Sayeed, K. S. Bangla grammatical error detection using t5 transformer model. *arXiv preprint arXiv:2303.10612* (2023).

[21] Sifat, M. H. R., Rahman, C. R., Rafsan, M., and Rahman, H. Synthetic error dataset generation mimicking bengali writing pattern. In *2020 IEEE Region 10 Symposium (TENSYMP)* (2020), IEEE, pp. 1363–1366.

[22] Sjarif, N. N. A., Azmi, N. F. M., Chuprat, S., Sarkan, H. M., Yahya, Y., and Sam, S. M. Sms spam message detection using term frequency-inverse document frequency and random forest algorithm. *Procedia Computer Science 161* (2019), 509–515.

[23] Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems (NIPS)* (2014), 3104–3112.

[24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems 30* (2017).

[25] Wikipedia. Bengali alphabet, Jan. 2024.

[26] Yoo, J.-Y., and Yang, D. Classification scheme of unstructured text document using tf-idf and naive bayes classifier. *Advanced Science and Technology Letters 111*, 50 (2015), 263–266.

[27] Zhou, X., Wan, X., and Xiao, J. Attention-based lstm network for cross-lingual sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (2016), pp. 247–256.