Neural Computing and Applications
https://doi.org/10.1007/s00521-023-09211-7

ORIGINAL ARTICLE

Panini: a transformer-based grammatical error correction method for Bangla

Nahid Hossain[1] · Mehedi Hasan Bijoy[2] · Salekul Islam[1] · Swakkhar Shatabda[1]

Received: 11 July 2023 / Accepted: 21 October 2023

Abstract

The purpose of the Bangla grammatical error correction task is to spontaneously identify and correct syntactic, morphological, semantic, and punctuation mistakes in written Bangla text using computational models, ultimately enhancing language precision and eloquence. The significance of the task encompasses bolstering linguistic acumen, fostering efficacious communication, and ensuring utmost lucidity and meticulousness in written expression, thereby mitigating the potential for obfuscation or dissemination of fallacious connotations. Prior endeavors have centered around surmounting the constraints inherent in rule-based and statistical methods through the exploration of machine learning and deep learning methods, aiming to enhance accuracy by apprehending intricate linguistic patterns, comprehending contextual cues, and discerning semantic nuances. In this study, we address the absence of a baseline for the task by developing a large-scale parallel corpus comprising 7.7M source-target pairs and exploring the untapped potential of transformers. Alongside the corpus, we introduce a Vaswani-style efficient monolingual transformer-based method named Bangla grammatical error corrector, Panini by leveraging transfer learning, which has become the state-of-the-art method for the task by surpassing the performance of both BanglaT5 and T5-Small by 18.81% and 23.8% of accuracy scores, and 11.5 and 15.6 of SacreBLEU scores, respectively. The empirical findings of the method substantiate its superiority over other approaches when it comes to capturing intricate linguistic rules and patterns. Moreover, the efficacy of our proposed method has been compared with the Bangla paraphrase task, showcasing its superior capability by outperforming the previous state-of-the-art method for the task as well. The BanglaGEC corpus and Panini, along with the baselines of BGEC and the Bangla paraphrase task, have been made publicly accessible at https://tinyurl.com/BanglaGEC.

Keywords Transformer · Panini · Grammar error correction · Bangla paraphrase · Transfer learning

1 Introduction

The grammatical error correction (GEC) task aims to autonomously identify and rectify errors in written texts, encompassing grammar, syntax, punctuation, and language rules, with the purpose of optimizing overall precision, coherence, and readability to facilitate effective written communication. Within the specific context of the Bangla Grammatical Error Correction (BGEC), the task involves automatically detecting and correcting grammatical errors in Bangla text. The significance of BGEC lies in its capacity to enhance communication, foster language acquisition, elevate writing excellence, safeguard the Bangla language, and escalate efficacy in handling Bangla text, benefiting individuals, educators, and organizations alike. Despite significant advancements in GEC for high-resource languages [1–4], the development of accurate and effective GEC systems for low-resource

Nahid Hossain and Mehedi Hasan Bijoy have contributed equally to this work.

One of the earliest linguists and grammararians, Bangla grammar follows the rules set by Panini.

✉ Swakkhar Shatabda
swakkhar@cse.uiu.ac.bd

Nahid Hossain
nahid@cse.uiu.ac.bd

Mehedi Hasan Bijoy
mhb6434@gmail.com

Salekul Islam
salekul@cse.uiu.ac.bd

[1] Computer Science and Engineering, United International University, Dhaka 1212, Bangladesh

[2] Computer Science and Engineering, Bangladesh University of Business and Technology, Dhaka 1216, Bangladesh

Published online: 04 December 2023

languages like Bangla remains a challenge. The intricacy of the Bangla language, characterized by its complex morphology, diverse verb forms, and tangled sentence structures, renders it one of the most formidable endeavors in Bangla Natural Language Processing. Moreover, the paucity of a publicly available large-scale corpus for Bangla grammatical error correction poses another inevitable constraint in the pursuit of developing exceedingly accurate models.

In the past decade, considerable research has been undertaken to address the BGEC task through rule-based [5–18] and statistical [19–30] approaches. The limitations of rule-based methods in the BGEC task, including the lack of flexibility, challenges in rule creation, limited coverage, inability to handle ambiguity, difficulty in handling language variation, and limited error detection capabilities, have prompted researchers to investigate data-driven statistical approaches as a means to overcome these limitations and enhance accuracy. Nevertheless, statistical methods also exhibit inherent limitations when it comes to contextual understanding, disentangling ambiguity, excessive reliance on handcrafted features, adaptability to novel languages, handling out-of-vocabulary words, grappling with data sparsity challenges, and limited error detection capabilities. These constraints have spurred the exploration of sophisticated methods such as machine learning [31, 32] and deep learning [33–42], aimed at ameliorating accuracy and fortifying the resilience of error correction systems. In recent years, the application of deep learning techniques has showcased promising accomplishments in the BGEC task owing to their enhanced capacity in capturing intricate linguistic patterns, deciphering context dependencies, and discerning semantic nuances. Lately, transformer-based methods have exhibited remarkable prowess in several Bangla natural language processing tasks, including machine translation [43], sentiment analysis [44], and spelling error correction [45], to name a few. A Vaswani et al. [46] style transformer architecture is utilized in our proposed method which is bifurcated into two integral components: the encoder and the decoder. Within the encoder module, there are pivotal elements including positional encoding, multi-head self-attention mechanism, layer normalization, residual connections, and feedforward neural networks. Likewise, the decoder module encompasses positional encoding, masked multi-head self-attention, layer normalization, residual connections, and feedforward neural networks. The mainstream structure of transformers typically involves stacking multiple encoder and decoder blocks. To the best of our knowledge, no transformer-based baseline for the BGEC task has been proposed yet. Henceforth, we endeavor to harness the formidable capabilities of transformers and embark upon an exploration of their untapped potential in the realm of BGEC.

Several constraints associated with the BGEC task have been identified, particularly concerning the dearth of a large-scale parallel corpus and the utilization of transformer-based monolingual methods. The objective of this study is to surmount the recognized limitations and provide a solid foundation for further advancements in the field by establishing a comprehensive baseline, thereby paving the way for future research endeavors. To do so, an extensive parallel corpus has been developed by carefully crafting a diverse set of Bangla grammar rules. Moreover, a monolingual transformer-based model named Panini has been proposed for the BGEC task. Additionally, a scrutiny was conducted to ascertain whether the performance of the monolingual transformer model is improved by the transfer learning technique. To this end, we initially train the model on a Bangla paraphrase dataset [47] and then transfer the acquired knowledge while addressing the BGEC task. In short, our proposed Panini accepts a grammatically erroneous sentence as input, which is subsequently tokenized using a pre-trained tokenizer [47]. The tokens are then fed into the encoder component of the model, where they undergo transformations resulting in a sequence of continuous representations. Following this, the decoder component integrates the output response from the encoder along with the output from its previous time step to generate the grammatically correct sentence.

The contributions of this article are summarized below:

- We propose a large-scale parallel corpus for the BGEC task, which comprises approximately 7.74M source-target instances. It has been created by carefully crafting a diverse set of intricate grammar rules, thus making Bangla a resourceful language for the task.
- A state-of-the-art monolingual transformer-based model named Panini has been introduced, exemplifying advancements in the BGEC task compared with other transformer-based baselines including BanglaT5 and T5-Small, thereby potentially heralding more sophisticated automated grammatical error correction.
- The impact of the training corpus size on the potency of the proposed method Panini in rectifying grammatical errors in Bangla has been investigated.
- The efficacy of transfer learning from the Bangla paraphrase task in the domain of BGEC has been meticulously scrutinized.
- The empirical outcomes of the proposed Panini have been juxtaposed with the baselines of the Bangla Paraphrase task, showcasing its supremacy in the task by surpassing the previous state-of-the-art performance with approximately 3.5 times fewer parameters, therefore attesting to its superior capabilities across different Bangla Natural Language Processing (BNLP) tasks.

The remaining sections of this paper are structured as follows: Sect. 2 provides a comprehensive review of

contemporary research in the domain of grammatical error correction, shedding light on the prevailing hurdles encountered specifically in the context of Bangla. In Sect. 3, we explicate the meticulous process employed for the creation of a large-scale parallel corpus, outlining the step-by-step procedure in a systematic manner. Subsequently, Sect. 4 elucidates the methodology and architectural design of our proposed monolingual transformer-based method. Next, Sect. 5 showcases the experimental setup and results along with the evaluation metrics employed to assess the performance of the model. Finally, Sect. 6 summarizes the results, implications, and potential future directions for this research.

2 Related work

A considerable amount of research has been carried out on correcting grammatical errors in the Bangla language. While the development of the Bangla GEC task has indeed gained steep attention since the late 2000s, it is evident that notable standards have yet to be achieved. The existing methods can broadly be classified into four primary groups including rule-based [5–18], statistical [19–30], machine-learning-based [31, 32], and deep-learning-based [33–42]. We observed that deep-learning-based and rule-based approaches became prominent and obscure after 2018, respectively.

2.1 Rule-based methods

The most commonly used rule-based schemes for Bangla grammatical error correction include context-free grammar (CFG) [5, 8, 16], context-sensitive grammar (CSG) [9, 14], head-driven phrase structure grammar (HPSG) [7], string matching algorithm [10], and Viterbi algorithm [17]. Among rule-based approaches, [5, 6, 8, 12], and [16] utilize the formalism of CFG by defining a set of valid grammar rules and determining whether a given sentence conforms to these rules or not. In particular, Purohit et al. [8] identify several features of Bangla words and further develop a set of semantic features for different word categories with the help of CFG to tackle the Bangla GEC task. A CFG-based predictive parser has been proposed by [5] and [6], which is implemented following a top-down fashion to avoid the left recursion issue of the CFG by left factoring, for Bangla grammar error correction. In 2016, Rabbi et al. [12] introduced a parsing method, to resolve the intricate and ambiguous Bangla grammar, by employing a shift-reduce parser through constructing a parse table based on the LR strategy. Recently, [16] utilized both CFG and CYK parsing algorithms for Bangla GEC and found that although the CFG-based parser performed better, the CYK-based parser worked faster. Another parser has been proposed by [9] which incorporates both CFG and CSG rules to parse Bangla complex and compound sentences semantically. However, Alamgir and Arefin [14] propose a CSG-based parser that prioritizes the intonation or mood of a sentence over its structure. Besides, [7, 10], and [17] bring forward a Bangla grammar checker using HPSG, string matching algorithm, and Viterbi algorithm. Unlike CFG-based parsers, the HPSG-based one [7] can detect syntactic and semantic errors in a sentence by utilizing the POS tags of words. Karim et al. [10] also utilize POS tags to determine sentence types, followed by validation of their structure through a string-matching algorithm. Furthermore, an augmented phrase structured grammar (APSG) rule-based semantic analyzer was proposed for scrutinizing the legitimacy of simple, complex, and compound Bangla sentences in 2018 [15]. A more recent study by Faisal et al. [18] presented a rule-based method for identifying grammatical errors in Bengali sentences employing only POS tags. First, they classified words into one of seven POS tags and then checked whether the resulting tag combination followed any of their manually written rules.

2.2 Statistical methods

In the case of statistical methods, the n-gram language model [20, 24, 29, 30] is found to be the most widely used where a few approaches utilized the frequency of words [21, 22] and term frequency-inverse document frequency (TF-IDF) [25] to fix Bangla grammatical mistakes. For instance, Kundu et al. [20] come up with a natural language generation (NLG)-based approach for Bangla GEC, which first transforms the input sentence into word vectors. These vectors are then fed into a bi-gram language model to determine whether the sentence is grammatically correct or not. Rana et al. [27] and Hossain et al. [30] have also introduced methods that combine bigram and trigram models to tackle Bangla homophone errors in real-world text and Bangla GEC, respectively. A similar method has been presented by Mridha et al. [28], which is the coalescence of bigram and trigram models, for addressing sentence-level missing word errors. Recently, a higher order n-gram ($n = 6$) model has been proposed to cluster Bangla words considering their contextual and semantic similarity [26]. However, to resolve the zero probability issue in the n-gram model, some approaches utilize smoothing techniques such as Witten–Bell [23, 24] and Kneser–Ney [29]. In 2020, Rahman et at. [29] experimented with both Witten–Bell and Kneser–Ney smoothing approaches to deal with missing words in the corpus. Their empirical outcomes manifested that Kneser–Ney outperforms Witten–Bell. Moreover, [21] and [22] have brought forward a Bangla grammar checker that counts the frequency of words. While a graph-based edge-weighting method has been described in [22]

to measure the semantic similarity between two words, a confidence score filter has been delineated in [21] to elect an appropriate sample from the outcomes. Lately, Nipu and Pal [25] proposed another Bangla grammar checker utilizing a vector space model (VSM) with TF-IDF scores.

2.3 Machine learning and deep learning-based methods

Due to recent advancements in Bangla NLP, machine learning [31, 32] and deep learning [33, 37, 39, 41, 42]-based approaches have become prominent in the Bangla GEC task because of their impressive performance. Especially, deep learning-based methods have received significant attention for their versatility in handling different types of grammar errors. In 2013, Kundu et al. [31] introduced a method that uses the K-Nearest Neighbors (k-NN) algorithm to correct Bangla grammatical errors. In addition, they introduced an active-learning-based novel complexity estimation matrix (CMM) for quantifying the grammatical intricacy of a sentence. A more recent study by Mridha et al. [32] presented a Naive Bayes classifier to address the same task to a broader extent, as it incorporates both typographical and grammatical errors. However, a word embedding-based tactic has been described in [34, 38], and [41] to grasp the semantic meaning, followed by cosine similarity to measure the semantic similarity of words. In [34], the authors use a pre-trained word2vec model with an embedding size of 300, which was trained on Bangla Wikipedia texts, to find semantic textual similarity. Pandit et al. [38] investigated a path-based and a distributional model for calculating semantic similarity in Bangla. Their experimental results favored the distributional model, which employed word2vec, over the path-based one. Furthermore, Iqbal et al. [41] inspected word2vec, GloVe, and FastText to calculate semantic similarity and found that FastText with a continuous bag-of-words outperformed word2vec and GloVe. Recent studies have utilized recurrent neural networks (RNN) to keep pace with advancements in NLP [33, 37, 39, 40]. Rakib et al. [36], for instance, used a Gated Recurrent Unit (GRU), a type of RNN cell, on an n-gram dataset to anticipate the next most appropriate word in Bangla sentences in 2019. In the same year, Islam et al. [37] employed long short-term memory (LSTM), another type of RNN cell, in the sequence-to-sequence model to generate coherent and grammatically correct Bangla sentences. Likewise, in 2021, the LSTM RNN cell is utilized by Chowdhury et al. [40] to determine the suitability, adjacency, and anticipation of simple Bengali sentences. Recently, Anbukkarasi and Varadhaganapathy [42] proposed a GRU-based grammar checker for Tamil which is a low-resource language. Furthermore, several studies have presented strategies that leverage the advantages of bidirectional LSTM RNNs [33, 35, 39]. Islam et al. [33] introduced a seq2seq model for both correcting and auto-completing Bangla sentences. Even though they employed a bi-LSTM RNN in the encoder, a conventional LSTM RNN with attention is used in the decoder part of the seq2seq model. Abujar et al. [35] and Noshin et al. [39] proposed bi-LSTM-RNN-based methods for predicting the next word in the sequence and correcting real-word errors in Bangla, respectively. Lately, the T5 model has been utilized by [48] for Bangla grammatical error correction, where it underwent fine-tuning on a tiny corpus consisting of only 9385 sentences, which is not enough for such a model. Nevertheless, we encountered the unattainable reproducibility of this endeavor, rendering the findings inconclusive.

2.4 Drawbacks of different methods

In brief, rule-based approaches are limited to a few rules when detecting Bangla grammatical errors, and their performance largely depends on POS tags. While these approaches are capable of efficiently correcting syntactic errors, they have limitations in their ability to address semantic errors. They are neither language-independent nor easy to maintain and update. In the case of statistical methods, reliable performance depends heavily on the quality of the corpus. Therefore, it is essential to have a balanced corpus for these methods to be effective. However, statistical methods struggle with correcting complex errors and have limitations in handling domain-specific language. Similarly, the machine learning-based approaches vastly rely on annotated training data, and they also have several limitations such as a lack of interpretability and performance degradation in noisy environments. On the flip side, deep learning-based methods outperform other approaches given sufficient training data.

However, the biggest barrier to the development of Bangla GEC is the scarcity of publicly available large-scale parallel corpora for the task. To address this challenge, we first develop a large-scale parallel corpus for the Bangla GEC task and make it publicly available. To the best of our knowledge, we are the first to propose a transformer-based method for Bangla grammatical error correction.

3 Corpus creation

The scarcity of parallel corpora is the most significant barrier to the development of effective natural language processing systems for grammatical error correction. In recent years, the availability of parallel corpora for some languages like English has significantly improved [4], but the situation is not the same for languages like Bangla due to limited resources. Therefore, we take the initiative to make Bangla a resourceful language for the GEC task by developing a large-scale parallel corpus. To do so, we identify seven primary

types of Bangla grammatical mistakes, including errors in verb inflection, number (bochon), word choice (homonym), sentence structure, punctuation, the agreement between subject and verb, and sentence fragments because of a missing subject or verb. Furthermore, sentence fragments can be further classified into four categories: subject missing, verb (from the dictionary) missing, auxiliary verb missing, and main verb missing. The grammatical errors we incorporated into our corpus are described below.

- **Verb Inflection.** It refers to a set of letters that correlate one word to another, particularly a noun or pronoun to its corresponding verb or adjective, in a sentence. It varies depending on the changes in number (bochon). Using verb inflection incorrectly can disrupt the relationship between a verb and its associated noun within a sentence. For example: (correct) মন্দিরের সামনে একটি বিশাল দিঘি আছে। → (erroneous) মন্দিরের সামনে একটি বিশাল দিঘি আছেন।.
- **Number (Bochon).** In Bengali grammar, the act of determining the quantity of nouns and pronouns is known as number (bochon). There are two types of numbers including singular and plural. This type of error occurs when the number of a noun or pronoun does not match the number of the verb, adjective, or article used in the sentence. There could be three variants of errors in number (bochon): (i) using a singular verb with a plural subject, or vice versa, (ii) using a singular article or adjective with a plural noun, or vice versa, and (iii) using a singular pronoun to refer to a plural noun or vice versa. It is worth mentioning that these errors can make a sentence difficult to understand or change its semantic meaning altogether. Therefore, it's important to pay attention to the correct use of singular and plural forms in Bengali grammar to ensure clear and effective communication. For example: (correct) তবে বিজ্ঞানীরা শুধু এতেই সন্তুষ্ট থাকলেন না। → (erroneous) তবে বিজ্ঞানীদের শুধু এতেই সন্তুষ্ট থাকলেন না।.
- **Word Choice (Homonym Error).** A homonym error is a mistake where two or more words sound the same but have different meanings. The wrong choice of words in a sentence essentially leads to confusion and miscommunication, especially in written language. Also, the mistake in choosing appropriate homonyms brings up the compatibility issue of the sentence. For example: (correct) এই নিয়মগুলির লক্ষ্য ছিল কালোদের ভোটের অধিকার থেকে বঞ্চিত করা। → (erroneous) এই নিয়মগুলির লক্ষ ছিল কালোদের ভোটের অধিকার থেকে বঞ্চিত করা।.
- **Sentence Structure.** It occurs when the arrangement of words in a sentence is incorrect, which consequently makes the sentence grammatically incorrect. It often changes the intended meaning and makes the sentence difficult to understand. For example: (correct) কেউ বাইরে যেতে পারবে না। → (erroneous) কেউ বাইরে পারবে যেতে না।.
- **Punctuation.** Punctuation marks are symbols that are used in writing to clarify the meaning and structure of a sentence. Some common punctuation marks in Bangla include full stop or period (), comma (,), semicolon (;), colon (ঃ), question mark (?), and exclamation mark (!). This type of error occurs when the punctuation marks are not used or are used incorrectly in a sentence. These errors affect the clarity and meaning of a sentence. For example: (correct) এরপর চুক্তির বাস্তবায়ন প্রক্রিয়া একটি বিরতির মধ্যে পড়ে↦ (erroneous) এরপর চুক্তির বাস্তবায়ন প্রক্রিয়া একটি বিরতির মধ্যে পড়ে?.
- **Subject-Verb Agreement.** This type of error occurs when there is a mismatch in person and number between the subject and verb. In Bangla grammar, person refers to the grammatical category that indicates the relationship between the speaker and the subject. Bangla grammar recognizes three persons - first person, second person, and third person. To ensure correct grammar in Bangla, the verb must agree with the subject in both person and number. For instance, if the subject is singular, the verb must be singular as well, and if the subject is plural, the verb should be plural. Similarly, if the subject is in the first person, the verb must also be in the first person, and so on. For example: (correct) তিনিনিজে যে মুক্তির স্বাদ পেয়েছেন, তা সবাইকে দিতে চান। → (erroneous) তুমি নিজে যে মুক্তির স্বাদ পেয়েছেন, তা সবাইকে দিতে চান।.
- **Sentence Fragments.** In Bangla grammar, a sentence fragment refers to a collection of words that do not constitute a complete sentence or convey a complete idea. These phrases usually lack a subject, a verb, or both, making them unable to function as complete sentences independently. Such a fragment can arise when a writer neglects to provide a complete sentence or excludes necessary components. This may cause ambiguity or misinterpretation in communication, leading to confusion or misunderstanding.
  - **Subject Missing.** This pertains to the circumstance in which the subject is left out or not included. For example: (correct) তিনি জানিয়েছেন অন্যান্য মূল ধারার রাজনৈতিক নেতাদের ক্ষেত্রেও সম্ভবত একই ব্যবস্থা নেওয়া হচ্ছে। → (erroneous) জানিয়েছেন অন্যান্য মূল ধারার রাজনৈতিক নেতাদের ক্ষেত্রেও সম্ভবত একই ব্যবস্থা নেওয়া হচ্ছে।.
  - **Verb (from the Dictionary) Missing.** This refers to a scenario in which a verb is absent from a predetermined list. To address this type of mistake, we com-

pile a list of verbs beforehand from online Bangla dictionaries. For example: (correct) কর্মকর্তারা বলছেন , যেসব বিষয় একজন শিক্ষার্থীর অবশ্যই জানা প্রয়োজন সেসব বিষয় সংক্ষিপ্ত সিলেবাসে রাখা হয়েছে। → (erroneous) কর্মকর্তারা বলছেন , যেসব বিষয় একজন শিক্ষার্থীর অবশ্যই প্রয়োজন সেসব বিষয় সংক্ষিপ্ত সিলেবাসে রাখা হয়েছে।.

- **Auxiliary Verb Missing.** This type of error occurs when the sentence lacks an auxiliary verb. For example: (correct) দ্রুতগতি সম্পন্ন ঘোড়ার পিঠে ছুটে গিয়ে তারা চারদিক থেকে ঘিরে ফেলতো প্রতিপক্ষকে। → (erroneous) দ্রুতগতি সম্পন্ন ঘোড়ার পিঠে ছুটে তারা চারদিক থেকে ঘিরে ফেলতো প্রতিপক্ষকে।.
- **Main Verb Missing.** This error arises when the sentence is missing a main verb. For example: (correct) এজন্য তাদের অর্থ উপার্জনের অন্যান্য উপায় খুঁজতে হবে। → (erroneous) এজন্য তাদের অর্থ উপার্জনের অন্যান্য উপায় হবে।.

3.1 Data sourcing

We source the raw data from a publicly available corpus named BanglaParaphrase [47], which comprises approximately 466k pairs of high-quality synthetic paraphrases in Bangla. These paraphrases are carefully crafted to ensure both semantic coherence and syntactic diversity, thus guaranteeing their superior quality.

3.2 Data augmentation

We introduce the previously discussed ten types of Bangla grammatical errors in the sourced data employing the noise injection technique. To do so, we consider each sentence as a finite set of words denoted as $S = \{W_1, W_2, \ldots, W_{N-1}, W_N\}$ where N is the length of the sentence such that $N \in Z^+$. Furthermore, each word $W_i \in S$ is considered as another finite set of Bangla characters which is represented as $W_i = \{C_1, C_2, \ldots, C_{M-1}, C_M\}$ where M is the length of the word such that $M \in Z^+$ as well. However, four different approaches have been initiated to propagate these ten types of errors due to their complex structures. Our process ensures that each synthetic erroneous sentence has only one mistake. Nevertheless, some sentences contain multiple erroneous words, and we take all of them into account in separate sentences. Therefore, this results in one correct sentence and multiple incorrect versions.

An analogous procedure has been implemented to embed verb inflection, number (bochon), and punctuation errors into a sentence, S. Firstly, a set of suffixes for inflection and number (bochon) errors and a set of Bangla punctuations for punctuation errors are collected, which are delineated as $A = \{a_1, a_2, a_3, \ldots, a_B\}$ where $a_j \in A$ is the jth suffix or punctuation symbol. The items in set A are further grouped into sub-lists based on the similarity of suffixes or punctuations which is denoted as $D_i = [d_1, d_2, d_3, \ldots d_E]$ such that $d_j \in A$ and D_i is the ith sub-list. Next, a dictionary is created incorporating these similar groups, which is described as $F = \{G_1 : D_1, G_2 : D_2, \ldots, G_N : D_N\}$ where G_j is the jth group name and D_j is its corresponding list of similar suffixes or punctuations. Then, we iterate each word of a sentence, $W_i \in S$, and determine whether it is found in the dictionary, F. If $W_i \in F$, we replace W_i with another suffix or punctuation d_j from its corresponding group, D_i, such that $d_j \in A$.

The same list of Bangla homonyms used by [45] is being utilized here. The homonym words' list is defined as $H = [(h_1, p_1), (h_2, p_2), \ldots, (h_K, p_K)]$ where h_j is the jth word of the list and p_j is its respective homonym version. To propagate the error, we iterate each word, W_j, in the sentence, S, and if it is found in the homonym words' list, H, we simply replace the word with its homonym version. Likewise, we accumulate a list of verbs, denoted as $V = [v_1, v_2, v_3, \ldots, v_K]$, from an online dictionary through web-scraping [45] to introduce the missing verb (from the dictionary) error. Again, we go through each word, W_j, in a sentence, S, and remove it upon its appearance in the previously collected verbs' list, such that $W_j \in V$. On the other hand, we introduce errors in sentence structure by randomly exchanging the positions of two words within a sentence.

In order to generate synthetic errors related to subject–verb agreement, missing subjects, missing auxiliary verbs, and missing main verbs, we make use of parts-of-speech (POS) tags obtained from a Bangla language toolkit called bnlp.[1] To begin with, we generate corresponding POS tags for each word, W_i, in the sentence, S, which is denoted as $S_t = \{pt_1, pt_2, \ldots, pt_N\}$. Then, we iterate through the tags set, S_t, and perform the following three operations: if the pos tag indicates a pronoun, auxiliary verb, and main verb, we remove it to introduce a missing subject error, missing auxiliary verb error, and missing main verb error, respectively. However, a slightly different approach is taken to introduce errors in subject–verb agreement. To do so, the subject and verb in the sentence (S) are determined using the POS tag list (S_t). Then, the subject is changed so that it mismatches with the form of the verb. In order to keep the resultant sentence semantically coherent even after introducing the error, a similar dictionary, previously used when introducing errors in verb inflection, number (bochon), and punctuation, has been developed for subjects as well.

3.3 Corpus statistic

The developed Bangla GEC corpus comprises ten distinct types of errors. The errors in verb inflection are found to be

[1] https://github.com/sagorbrur/bnlp.

the most frequent (25.51%) and word choice is found to be the least frequent (2.09%) type of error in the corpus. The error in verb inflection is the second largest type of error, containing 1,804,721 pairs of instances, which is slightly above a quarter. Four out of ten types of errors, including verb inflection, number, errors in sentence structure, and main verb missing errors, comprise 78.60% of total errors in the corpus, while the remaining six comprise the remaining 21.33% of errors. Even though word choice or homonym errors, punctuation errors, subject–verb agreement errors, subject missing errors, missing dictionary verb errors, auxiliary verb missing errors, and main verb missing errors comprise 2.09%, 7.40%, 3.22%, 2.19%, 2.44%, and 3.36% of the corpus, respectively, each type of error contains a substantial amount of instances. For instance, the top three least frequent types of errors are word choice or homonym errors, missing dictionary verb errors, and subject missing errors, each containing 147,737, 172,704, and 197,540 instances, respectively.

The percentage of different error types is justified as none of them were introduced manually. All the instances have been crafted automatically based on the underlying corpus and predefined suffixes which are carefully extracted by experts in the language. Moreover, error related to word choice is the least common in the corpus, which is logical considering the fact that the Bangla language has a relatively small number of homonyms. On the other hand, the most prominent error type in the corpus is related to verb inflection, which is not surprising given the fact that the Bangla language has a wide range of verb inflection suffixes.

4 Methodology

The proposed method is twofold: initially, a transformer-based seq2seq (MarianMT [49]) model, $\mu(.)$, undergoes training on a Bangla Paraphrase task, and subsequently, the acquired knowledge is transferred to an identical model that is tweaked for the Bangla GEC task. In either case, the initial step involves passing an input sentence, $[x_1, x_2, \ldots, x_n]$, through a pre-trained Bangla tokenizer, $\tau(.)$, that transforms the text into numerical data. After the input sentence has been tokenized, it is then fed into the model, $\mu(.)$, to make a prediction. Finally, the model's prediction is assessed using relevant metrics specific to the task at hand. The entire Bangla GEC method is depicted in Fig. 1. In mathematical terms, the entire process can be succinctly summarized as:

$$\hat{y} = \mu(\tau([x_1, x_2, \ldots, x_n])) \quad (1)$$

4.1 Problem formulation

The word-level Bangla grammatical error correction task strives to map an erroneous sequence denoted as $X = [x_1, x_2, \ldots, x_n]$ into the corresponding correct sequence denoted as $Y = [y_1, y_2, \ldots, y_m]$ where X_i and Y_j are the ith and jth word of the erroneous and correct sentences, respectively, such that the lengths $n \in Z^+$ and $m \in Z^+$ but are not necessarily required to be equal. The erroneous sentence, X, is first fed into the pre-trained tokenizer, $\tau(.)$, that tokenizes X which is represented as $X_\tau = [x_{\tau_1}, x_{\tau_2}, \ldots, x_{\tau_n}]$ where x_{τ_i} is the numerical value of ith token if and only if the word is present in the vocabulary, otherwise a unknown ($<unk>$) token. Next, the Bangla grammatical error correction model, denoted as $\mu(.)$, processes the tokenized sentence X_τ and produces a prediction referred to as $\hat{y}$. Lastly, the model's prediction is assessed by means of several evaluation metrics by comparing $\hat{y}$ with corresponding target.

4.2 Panini

It is essentially a tweaked MarianMT [49], which is a Vaswani et al. [46] style seq2seq transformer model, for Bangla GEC Task. MarianMT[49] was selected primarily because it incorporates several low-resource machine translation techniques for grammatical error correction (GEC) tasks and has achieved state-of-the-art results in neural GEC on several benchmarks, such as the CoNLL-2014 and JFLEG test sets. However, the model consists of a stack of 12 encoder and decoder blocks, with each block including self-attention, recurrent connections, and feedforward neural networks. Below are the descriptions of the encoder and decoder blocks.

4.2.1 Encoder

It is responsible for processing the input sequence of tokens, $X_\tau = [x_{\tau_1}, x_{\tau_2}, \ldots, x_{\tau_n}]$, and producing a sequence of hidden states that capture the meaning and context of each token in the input by incorporating positional encoding. Especially, each encoder layer includes a self-attention mechanism that computes attention scores between all pairs of tokens in the input sequence, and a feedforward neural network that applies a nonlinear transformation to the output of the self-attention mechanism. The self-attention mechanism computes a weighted sum of the hidden states for each token in the input sequence, where the weights are based on the similarity between the token and all other tokens in the sequence. This allows the encoder to focus on the most relevant parts of the input sequence for each token, taking into account the

context in which it appears. However, transformer model uses multi-head self-attention to capture multiple relationships, increase expressive power, become robust to variations in data, and address regularization. The formula for calculating each head's self-attention is as follows:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

Here, Q, K, and V are matrices of queries, keys, and values, respectively. The d_k is the dimension of Keys that is utilized to scale the resultant score.

The multi-head self-attention is essentially the amalgamation of each head's outcome which can be represented as,

$$MultiHead(Q, K, V) = Concat(Head_1, Head_2, \ldots, Head_h)W^o \tag{3}$$

$$Head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{4}$$

Here, $Head_i$ is the ith head, and W_i^Q, W_i^K, W_i^V are the corresponding weight metrics of the queries (Q), keys (K), and values (V).

Finally, a feed-forward neural network ($\mathcal{F}(.)$) takes the response of multi-head self-attention followed by a recurrence connection and applies a nonlinear transformation to the output of the self-attention mechanism, which allows the encoder to capture more complex relationships between the tokens in the input sequence.

4.2.2 Decoder

It is responsible for generating the output sequence based on the encoded input sequence generated by the encoder. The decoder is auto-regressive that takes the encoded input sequence and generates the output sequence token-by-token. Each decoder layer comprises of masked multi-head self-attention layer, multi-head attention layer, and feed-forward neural network layer. First, masked self-attention is computed over the target sequence Y. The masked multi-head self-attention layer is similar to the self-attention layer in the encoder but with a mask applied to ensure that the decoder cannot attend to future tokens in the output sequence. This sub-layer allows the decoder to attend to relevant parts of the output sequence generated so far and capture the dependencies between the tokens in the output sequence. Next, attention is computed over the encoded hidden representations H. The multi-head attention layer is responsible for attending to the encoded input sequence generated by the encoder. This sub-layer enables the decoder to incorporate information from the input sequence into the output sequence and produce a translated version of the input sequence. Then, a position-wise feed-forward network is applied to the output representation obtained in the previous step. The feed-forward neural network ($\mathcal{F}(.)$) layer applies a non-linear transformation including residual connections and layer normalization to the output of the attention layers to generate the final output sequence. During training, the decoder uses teacher forcing, where the true previous token is fed as input to the decoder at each time step. During inference, the decoder generates the output sequence token-by-token by recursively predicting the most likely token at each time step based on the previous tokens and the encoded input sequence.

Table 1 Statistic of the Bangla GEC corpus

Error type	#No. of instances	Percentage (%)
Verb inflection	1,804,721	25.51
Number (Bochon)	709,480	10.03
Word choice (homonym error)	147,737	2.09
Sentence structure	1,795,641	25.38
Punctuation	523,255	7.40
Subject–verb agreement	227,534	3.22
Subject missing	197,540	2.79
Verb (from the dictionary) missing	172,704	2.44
Auxiliary verb missing	237,741	3.36
Main verb missing	1,258,072	17.78
	Total = 7,074,425	

5 Experimental analysis

5.1 Bangla GEC corpus

We developed and published a large-scale Bangla GEC parallel corpus containing 7,074,425 (≈7.1M) source-target pairs. It amalgamates 10 distinct types of Bangla grammar errors, as depicted in Table 1. To maintain optimal model performance and avoid unnecessary asymptotic complexity, we limited the maximum sentence length to 50, as we observed that including longer sentences did not yield any improvements in model performance. The frequencies of sentence lengths are illustrated in Fig. 2. We employ the dataset for the Bangla grammar error correction task by partitioning it into training, validation, and test sets, ensuring that all three subsets incorporate the comprehensive range of all 10 error types.

- **Training Set:** There are 5,730,275 (≈5.73M) instances in the training set. This subset is primarily utilized to train the model. The model assimilates information from these instances to discern intricate patterns, establish correlations, and generate accurate predictions.

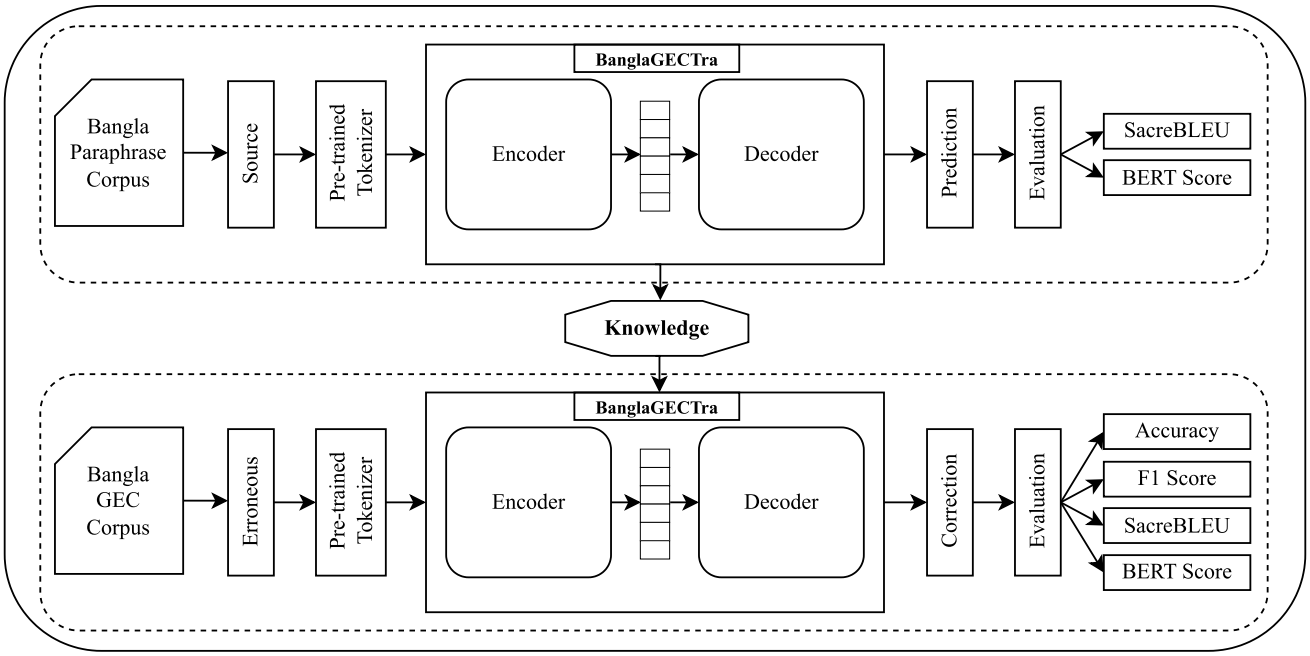

Fig. 1 (Top) The Panini is initially trained on the Bangla paraphrase task. It begins by taking a source sentence as input, which is then tokenized using a pre-trained tokenizer, $\tau(.)$. Finally, the model, $\mu(.)$, generates a prediction. The knowledge acquired during this process is saved for further use. (Middle) We employed transfer learning by initially subjecting the model to pretraining on the Bangla paraphrase task, followed by harnessing the saved weights to enhance both the learning dynamics and the efficacy of the model for the Bangla grammatical error correction task. Here, the term 'knowledge' refers to the insights garnered from the Bangla paraphrase task. (Bottom) The Panini is being trained here to address the BGEC task, harnessing the knowledge accrued from the Bangla paraphrase task via transfer learning. The process commences by embracing an erroneous input sentence, subsequently subjecting it to tokenization using the pre-trained tokenizer, $\tau(.)$. These tokenized inputs are then fed to the model, $\mu(.)$, which proficiently generates the requisite correction

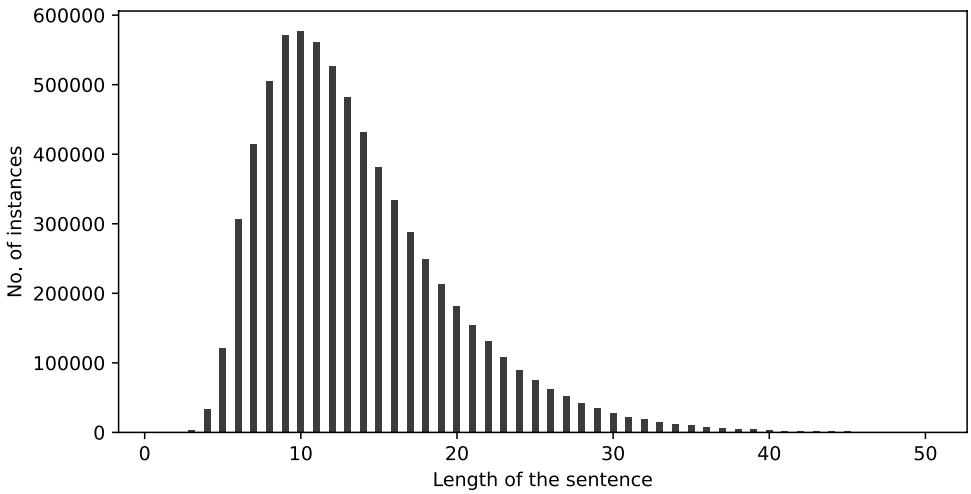

Fig. 2 The instances versus sentence length plot, which provides a visual representation of the sentence length distribution within the BGEC corpus

- **Validation Set:** It comprises 636,702 (≈636.7K) instances which are used to assess the performance and fine-tune the parameters of the trained model.
- **Test Set:** The test set encompasses 707,448 (≈707.4K) instances, which are kept untouched during training or validation to evaluate the generalization capabilities of the model, unveiling an impartial gauge of its efficacy in handling unseen data.

5.2 Baselines

- **Bangla-T5** [47]. This is a large-scale pre-trained language model for the primary purpose of performing the

Bangla paraphrase task. As it is a transformer-based pre-trained sequence-to-sequence model, we further fine-tuned and cross-checked its performance on the Bangla GEC task by transferring knowledge.

- **T5-Small** [50]. It is an efficient version of the T5 language model developed by Google, designed to require fewer resources while still maintaining high performance. We first fine-tuned the model for the downstream Bangla paraphrase task. Then, we tailored the model for the Bangla GEC task while transferring the knowledge gained from the paraphrasing task.

5.3 Performance evaluation

- **Accuracy and F1-Score.** Accuracy and F1 score are two commonly used metrics to evaluate the performance of a model. Accuracy is a measure of how often the model is correct in its predictions. It is defined as the number of correct predictions divided by the total number of predictions. Mathematically, it can be expressed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. While accuracy is a useful metric, it can be misleading in cases where the classes are imbalanced. The F1 score is a more balanced metric that takes into account both precision and recall. Precision measures how many of the positive predictions are correct, while recall measures how many of the positive examples are correctly predicted. The F1 score is the harmonic mean of precision and recall and is calculated as follows:

$$F1\ Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{6}$$

where $precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+FN}$.

- **BERT Score.** It measures the similarity between two pieces of text based on their semantic meaning. The mathematical equation for BERT score is as follows:

$$BERT\ Score = \frac{1}{N} \times \sum_{i}^{N} F1\ Score(BERT_{similarity_i}, BERT_{precision_i}, BERT_{recall_i}) \tag{7}$$

where N is the number of text pairs being evaluated, F1 Score is the harmonic mean of $BERT_{precision}$ and $BERT_{recall}$, $BERT_{similarity_i}$ is the cosine similarity between the embeddings of the two text pieces, $BERT_{precision_i}$ the proportion of words in the first text that have a matching word in the second text, and $BERT_{recall_i}$ is the proportion of words in the second text that have a matching word in the first text.

- **SacreBLEU.** It also measures the similarity between prediction and one or more human-annotated references. It evaluates not only the accuracy of individual words but also the overall fluency and coherence of the translated text. The score ranges from 0 to 100, with higher scores indicating better performance. The mathematical equation for calculating SacreBLEU score is as follows:

$$SacreBLEU = 100 \times e^{(1-BP)} \times \left(\frac{sum\ of\ Ngram\ matches}{total\ Ngrams} \right) \tag{8}$$

$$BP = min(1, e^{(1 - \frac{reference\ length}{candidate\ length})}) \tag{9}$$

where BP or Brevity Penalty is a penalty term that adjusts the score for the length of the candidate predictions relative to the length of the reference annotations, the reference length, and candidate length are the length of the concatenated predictions and candidate annotations, respectively. The sum of n-gram matches indicates the total number of matching n-grams in the candidate annotations and total n-grams refers to the number of n-grams in the candidate predictions.

5.4 Hyperparameters

The default configuration for MarianMT [49] uses six stacked layers for both the encoder and decoder. Likewise, MarianMT's default hidden layer size of 512 is utilized to represent the feature vectors. A learning rate of 5×10^5 is employed with a batch size of 16 during the training process. The model is trained for 30 epochs with the AdamW optimizer for updating the weights of the network to minimize the loss function.

5.5 Transformers on Bangla GEC task

5.5.1 Quantitative result

The quantitative results of different BGEC baselines are demonstrated in Table 2, which explicitly manifests the preeminence of our proposed Panini model over the BanglaT5 and T5-Small models in the BGEC task by attaining an accuracy score of 83.33%, an f1-score of 0.833, a BERT score of 99.43, and a ScarceBLEU score of 95.9. However, the T5-Small model exhibits relatively inferior performance compared to the other models, which is

Table 2 The table of the quantitative results, where the empirical outcomes of our proposed Panini are compared with other transformer-based baselines for the BGEC task

Method	Training	Inference				Param. (M)
	Loss	Accuracy (%)	F1 score	BERT score	ScarceBLEU	
BanglaT5	4.21×10^{-2}	64.52	0.645	97.5	84.4	247.53
T5-Small	4.5×10^{-2}	59.53	0.595	96.73	80.3	60.51
Panini	2.79×10^{-2}	83.33	0.833	99.43	95.9	74.36

Table 3 The qualitative results table that elucidates the effectiveness of Panini and other transformer-based baselines in rectifying Bangla grammatical errors

(Input)	তিনি আরও বলেন, এটা কোন জঙ্গি তৎপরতা নয় বলেই তারা মনে করছেন। (He also said that they think it is not a terrorist activity.)
(BanglaT5)	তিনি আরও বলেন, এটি কোন জঙ্গি তৎপরতা নয় বলেই তারা মনে করছেন। (He also said that they think it is not a terrorist activity.) (✓)
(T5-Small)	তিনি আরও বলেন, এটা কোন জঙ্গি তৎপরতা নয় বলেই তারা মনে করছে । (He also said that they think it is not a terrorist activity.) (×)
(Panini)	তিনি আরও বলেন, এটি কোন জঙ্গি তৎপরতা নয় বলেই তারা মনে করছেন। (He also said that they think it is not a terrorist activity.) (✓)
(Input)	তারা মধ্যে অধিকাংশই সাধারণ মানুষ যারা বাশার আলআসাদের বিরুদ্ধে বিদ্রোহীদের সমর্থন করে, যারা গ্রেপ্তার ও হয়রানির ভয়ে আসাদের সীমাবদ্ধ এলাকায় যাওয়ার পরিবর্তে অবশেষে বিদ্রোহী নিয়ন্ত্রিত এলাকা ইদলিবে চলে যায়, যেখানে বিদ্রোহীরা তাদের অবশিষ্ট শেষ অঞ্চল দখল করে। (Most of them are ordinary civilians who support the rebels against Bashar al-Assad, who opt to relocate to the rebel-controlled areas in Idlib instead of staying in the restricted territories under Assad's control, where the rebels hold their last remaining territory.)
(BanglaT5)	তাদের মধ্যে অধিকাংশই সাধারণ মানুষ যারা বাশার আলআসাদের বিরুদ্ধে বিদ্রোহীদের সমর্থন করে, যারা গ্রেপ্তার ও হয়রানির (Most of them are ordinary people who support the rebels against Bashar al-Assad, who are arrested and harassed) (×)
(T5-Small)	তাদের মধ্যে অধিকাংশই সাধারণ মানুষ যারা বাশার আলআসাদের বিরুদ্ধে বিদ্রোহীদের সমর্থন করে, যারা গ্রেপ্তার ও হয়রানির (Most of them are ordinary people who support the rebels against Bashar al-Assad, who are arrested and harassed) (×)
(Panini)	তাদের মধ্যে অধিকাংশই সাধারণ মানুষ যারা বাশার আলআসাদের বিরুদ্ধে বিদ্রোহীদের সমর্থন করে, যারা গ্রেপ্তার ও হয়রানির ভয়ে আসাদের সীমাবদ্ধ এলাকায় যাওয়ার পরিবর্তে অবশেষে বিদ্রোহী নিয়ন্ত্রিত এলাকা ইদলিবে চলে যায়, যেখানে বিদ্রোহীরা তাদের শেষ অবশিষ্ট অঞ্চল দখল করে। (Most of them are ordinary civilians who support the rebels against Bashar al-Assad, who opt to relocate to the rebel-controlled areas in Idlib instead of staying in the restricted territories under Assad's control, where the rebels hold their last remaining territory.) (✓)
(Input)	ভোর শেষ করে নিচের ডেকে চলে গেলাম। (At the end of the morning, went to the lower deck.)
(BanglaT5)	ভোর শেষ করে আমি নিচের ডেকে চলে গেলাম। (At the end of the morning, I went to the lower deck.) (×)
(T5-Small)	ভোর শেষ করে আমি নিচের ডেকে চলে গেলাম । (At the end of the morning, I went to the lower deck.) (×)
(Panini)	ভোর শেষ করে আমরা নিচের ডেকে চলে গেলাম। (At the end of the morning we went down to the lower deck.) (×)

expected given its substantially lower number of trainable parameters. On the other hand, the BanglaT5 model, which possesses around 3.5 times more parameters than our model, falls significantly short in all evaluation criteria, making it the second-best performer overall. Our Panini outperforms BanglaT5 with a substantial lead, improving the accuracy score by 18.81%, the f1-score by 0.19, the BERT Score by 1.93, and the ScarceBLEU score by 11.5.

5.5.2 Qualitative result

Table 3 exemplifies the qualitative outcomes of BanglaT5, T5-Small, and Panini. It unequivocally illustrates the outstanding performance of Panini compared to BanglaT5 and T5-Small models. Our Panini model excels in rectifying a myriad of grammatical errors, including subject–verb agreement, tense inconsistencies, articles, prepositions, punctuation, and verb agreement, to name a few. This extraordinary proficiency showcases its superior aptitude in capturing intricate language patterns across diverse error typologies contrasted to the other two baseline models. Moreover, Panini exhibits adeptness in resolving errors in long sentences, in which the other two baselines fall short. Furthermore, the errors made by our method also manifest a discernible degree of meaningfulness, duly considering the semantic meaning of the predicted sentence. For instance, for the erroneous input "ভোর শেষ করে নিচের ডেকে চলে গেলাম।", it generated the correction as "ভোর শেষ করে আমরা নিচের ডেকে চলে গেলাম।", which is not semantically wrong at all.

To make the grammatical correction, our Panini takes the erroneous sentence as input and processes it using the encoder module. However, the decoding process begins by generating the first token of the target sentence using embeddings and positional encodings, similar to the encoder. Then, the decoder's masked multi-head

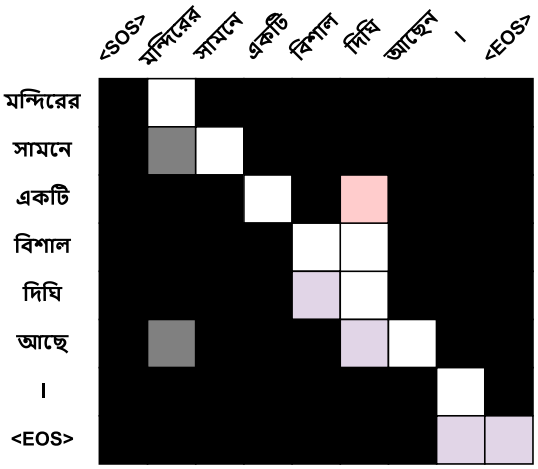

Fig. 3 Illustration of the attention dynamics across the source erroneous sentence for each decoding step

Table 4 Comparing the empirical outcomes of various transformer-based methods on the Bangla paraphrase task

Method	SacreBLEU	BERT score	Param. (M)
BanglaT5	29.6	89.29	247.53
T5-Small	20.8	86.86	60.51
Panini	27.5	91.13	74.36

self-attention mechanism focuses on the generated tokens, aiding the understanding of token relationships. Next, the encoder–decoder attention allows the decoder to reference the source sentence, aiding alignment between the source and target. Finally, the decoder predicts subsequent tokens using self-attention and encoder–decoder attention, iteratively generating tokens until the end-of-sentence ($< EOS >$) token is reached. Figure 3 illustrates how each word in the source erroneous sentence attends to and influences the words in the target corrected sentence during the translation process.

5.6 Transformers on Bangla paraphrase task

The BanglaT5, T5-Small, and Panini demonstrate competitive performance in the Bangla paraphrase task. While BanglaT5 and Panini achieve the highest SacreBLEU score and BERT score, respectively, the T5-Small model has the fewest trainable parameters among them, which refers to better asymptotic complexity. The BanglaT5 scores the highest SacreBLEU score of 29.6, which is 8.8 and 2.1 points higher than the performance of the T5-Small and Panini models. On the other hand, Panini enhances the BERT score of BanglaT5 and T5-Small by 1.85 and 4.27 points, respectively. Despite achieving a higher SacreBLEU score than Panini, BanglaT5's size is 3.33 times larger than that of Panini, which results in a significant increase in asymptotic complexity. This leads to atypical training and inference time. On the contrary, T5-Small, with the smallest parameter size of 60.51M, fails to achieve a high SacreBLEU or BERT score. Here, our Panini achieves the highest BERT score of 91.13 and demonstrates competitive performance in terms of the SacreBLEU score, considering its relatively small parameter size (Table 4).

5.7 Ablation study

To assess the influence of training corpus size on the efficacy of our proposed Panini, we undertook a series of experiments employing three different versions of the training set. We evaluated their performance on a shared test set, which comprised 50K instances. The three training sets varied in size, with the first and second sets scaled down by factors of 100 and 50, respectively, compared to the actual set. These smaller sets contained approximately 57.3K and 114.6K instances, respectively. In contrast, the third set was a comprehensive large-scale training set consisting of around 5.73M instances. The empirical performance of Panini on these three versions of the corpus can be found in Table 5.

The empirical findings (Table 5) emphasize the significant impact of utilizing a large-scale corpus in Panini to achieve optimal performance. During the model training on 57.3K instances, it achieved an accuracy of 52.28%, an f1 score of 0.523, a BERT score of 98.64, and a SacreBLEU score of 89.1, respectively. Then, training the model on another variant of the corpus, consisting of 114.6K instances, led to slight improvements in accuracy, f1 score, BERT score, and SacreBLEU, with an increase of 6.44%, 6.4×10^{-2}, 0.18, and 1.5, respectively. Although these improvements were negligible, training the model on the actual training set, which comprised

Table 5 The effectiveness of our proposed Panini in the context of the BGEC task across varying sizes of the training corpus

Method	Corpus size		Inference			
	Train	Test (K)	Accuracy (%)	F1 score	BERT score	SacreBLEU
Panini	57.3K	50	52.28	0.523	98.64	89.1
Panini	114.6K	50	58.72	0.587	98.82	90.6
Panini	5.73M	50	87.52	0.875	99.61	96.3

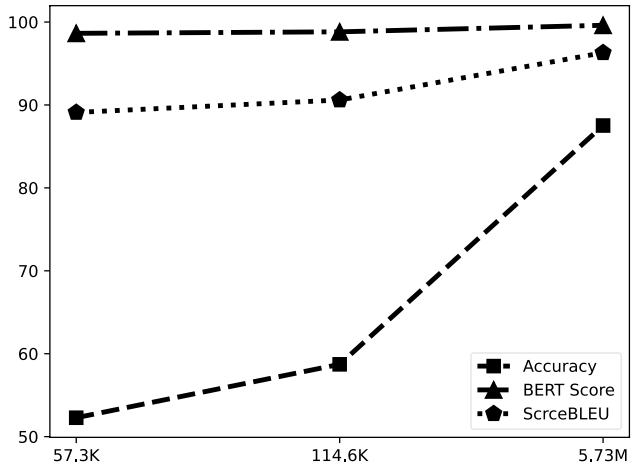

Fig. 4 The empirical outcomes of Panini on three different-sized training sets in terms of accuracy, BERT Score, and SacreBLEU

5.73M instances, resulted in more significant enhancements, with an increase of 35.22% in accuracy, 0.352 in f1 score, 0.97 in BERT score, and 7.2 in SacreBLEU, respectively (Fig. 4).

The ablation study serves as a vivid depiction of how corpus size profoundly impacts the model's performance. The findings of this study provide a lucid and all-encompassing comprehension of the intricate interplay between the size of the training corpus and the efficacy of the model. As the corpus size increases, our model consistently exhibits a discernible elevation in its performance trajectory. This empirical insight emphatically underscores the pivotal role data volume plays in increasing the model's performance, thereby enhancing its proficiency and effectiveness.

6 Conclusion

The Bangla grammatical error correction task holds paramount importance in ensuring the utmost perspicuity and scrupulousness of written Bangla text, thus bestowing a pinnacle of clarity and precision. In response to the task, a monolingual transformer-based baseline for Bangla grammatical error correction has been introduced in this study, aiming to fulfill the need for an effective BGEC method in the Bangla language by utilizing transformer models. In pursuit of this objective, a large-scale parallel corpus for the task has been developed and made publicly accessible, which in turn has made Bangla no longer a low-resource language for the task. Moreover, we introduced Panini, which has emerged as a new state-of-the-art method, outperforming the BanglaT5 and T5-Small baselines by a significant margin. It excelled not only in the BGEC task but also in the Bangla paraphrase task, surpassing the performance of the previous state-of-the-art method. Furthermore, the efficacy of transfer learning from the Bangla paraphrase task in the context of the BGEC task has been thoroughly examined and analyzed. However, the scrutiny of the influence of training corpus size on the effectiveness of our proposed Panini has unveiled the significant data dependency of the method, highlighting its profound reliance on extensive data resources. Nevertheless, while Panini exhibited commendable outcomes on BGEC task, there remains ample scope for improving the performance by targeting specific error categories and refining its efficacy. Overall, we introduced a robust foundation for the BGEC task, serving as a baseline for forthcoming advancements in the task. In future, we will alleviate the model's reliance on copious data through the utilization of zero-shot learning. Also, we will empirically investigate the efficacy of combining a pre-trained model from other languages with our monolingual pre-trained model through the utilization of knowledge distillation.

Acknowledgements This research is funded by Institute of Advanced Research (Grant No. UIU/IAR/02/2021/SE/22), United International University, Bangladesh.

Data availability Datasets will be made available on request.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Rozovskaya A, Roth D (2019) Grammar error correction in morphologically rich languages: the case of Russian. Trans Assoc Comput Linguist 7:1–17
2. Hu L, Tang Y, Wu X, Zeng J (2022) Considering optimization of English grammar error correction based on neural network. Neural Comput Appl 66:1–13
3. Grundkiewicz R, Junczys-Dowmunt M, Heafield K (2019) Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In: Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications, pp 252–263
4. Wang Y, Wang Y, Dang K, Liu J, Liu Z (2021) A comprehensive survey of grammatical error correction. ACM Trans Intell Syst Technol 12(5):1–51
5. Hasan KA, Mondal A, Saha A (2010) A context free grammar and its predictive parser for Bangla grammar recognition. In: 2010 13th International conference on computer and information technology (ICCIT). IEEE, pp 87–91
6. Hasan K, Mondal A, Saha A et al (2012) Recognizing Bangla grammar using predictive parser. arXiv preprint arXiv:1201.2010
7. Islam MA, Hasan KA, Rahman MM (2012) Basic hpsg structure for Bangla grammar. In: 2012 15th International conference on computer and information technology (ICCIT). IEEE, pp 185–189
8. Purohit PP, Hoque MM, Hassan MK (2014) An empirical framework for semantic analysis of Bangla sentences. In: 2014 9th International forum on strategic technology (IFOST). IEEE, pp 34–39

9. Purohit PP, Hoque MM, Hassan MK (2014) Feature based semantic analyzer for parsing Bangla complex and compound sentences. In: The 8th International conference on software, knowledge, information management and applications (SKIMA 2014). IEEE, pp 1–7
10. Karim MS, Robi FRH, Hossain MM, Rahman MT et al (2018) Implementation and performance evaluation of semantic features analysis system for Bangla assertive, imperative and interrogative sentences. In: 2018 International conference on bangla speech and language processing (ICBSLP). IEEE, pp 1–5
11. Hasan KA, Hozaifa M, Dutta S (2014) Detection of semantic errors from simple Bangla sentences. In: 2014 17th International conference on computer and information technology (ICCIT). IEEE, pp 296–299
12. Rabbi RZ, Shuvo MIR, Hasan KA (2016) Bangla grammar pattern recognition using shift reduce parser. In: 2016 5th International conference on informatics, electronics and vision (ICIEV). IEEE, pp 229–234
13. Al Hadi A, Khan MYA, Sayed MA (2016) Extracting semantic relatedness for Bangla words. In: 2016 5th International conference on informatics, electronics and vision (ICIEV). IEEE, pp 10–14
14. Alamgir T, Arefin MS (2017) An empirical framework for parsing Bangla imperative, optative and exclamatory sentences. In: 2017 International conference on electrical, computer and communication engineering (ECCE). IEEE, pp 164–169
15. Khatun S, Hoque MM (2018) Semantic analysis of Bengali sentences. In: 2018 International conference on bangla speech and language processing (ICBSLP). IEEE, pp 1–6
16. Saha Prapty A, Rifat Anwar M, Azharul Hasan K (2021) A rule-based parsing for Bangla grammar pattern detection. In: Proceedings of international joint conference on advances in computational intelligence: IJCACI 2020. Springer, pp 319–331
17. Afroz S, Susmoy M, Anjum F, Nowshin N (2021) Examining lexical and grammatical difficulties in Bengali language using nlp with machine learning. PhD thesis, Brac University
18. Faisal AMF, Rahman MA, Farah T (2021) A rule-based Bengali grammar checker. In: 2021 Fifth world conference on smart trends in systems security and sustainability (WorldS4). IEEE, pp 113–117
19. Alam M, UzZaman N, Khan M et al (2007) N-gram based statistical grammar checker for Bangla and English
20. Kundu B, Chakraborti S, Choudhury SK (2011) Nlg approach for Bangla grammatical error correction. In: 9th International conference on natural language processing, ICON, pp 225–230
21. Kundu B, Chakraborti S, Choudhury SK (2012) Combining confidence score and mal-rule filters for automatic creation of Bangla error corpus: grammar checker perspective. In: Computational linguistics and intelligent text processing: 13th international conference, CICLing 2012, New Delhi, India, March 11–17, 2012, Proceedings, Part II 13. Springer, pp 462–477
22. Sinha M, Dasgupta T, Jana A, Basu A (2014) Design and development of a Bangla semantic lexicon and semantic similarity measure. Int J Comput Appl 975:8887
23. Khan NH (2014) Verification of Bangla sentence structure using n-gram. Glob J Comput Sci Technol 14:1–5
24. Rahman MR, Habib MT, Rahman MS, Shuvo SB, Uddin MS (2016) An investigative design based statistical approach for determining Bangla sentence validity. Int J Comput Sci Netw Secur 16(11):30–37
25. Nipu AS, Pal U (2017) A machine learning approach on latent semantic analysis for ambiguity checking on Bengali literature. In: 2017 20th International conference of computer and information technology (ICCIT). IEEE, pp 1–4
26. Husna A, Mostofa M, Khatun A, Islam J, Mahin M (2018) A framework for word clustering of Bangla sentences using higher order n-gram language model. In: 2018 International conference on innovation in engineering and technology (ICIET). IEEE, pp 1–6
27. Rana MM, Sultan MT, Mridha M, Khan MEA, Ahmed MM, Hamid MA (2018) Detection and correction of real-word errors in Bangla language. In: 2018 International conference on bangla speech and language processing (ICBSLP). IEEE, pp 1–4
28. Mridha M, Rana MM, Hamid MA, Khan MEA, Ahmed MM, Sultan MT (2019) An approach for detection and correction of missing word in Bengali sentence. In: 2019 International conference on electrical, computer and communication engineering (ECCE). IEEE, pp 1–4
29. Rahman MR, Habib MT, Rahman MS, Islam GZ, Khan MAA (2020) An exploratory research on grammar checking of Bangla sentences using statistical language models. Int J Electr Comput Eng 10(3):3244–3252
30. Hossain N, Islam S, Huda MN (2021) Development of Bangla spell and grammar checkers: resource creation and evaluation. IEEE Access 9:141079–141097
31. Kundu SB, Chakraborti S, Choudhury SK (2013) Complexity guided active learning for Bangla grammar correction. In: 10th International conference on natural language processing, ICON, vol 1, p 4
32. Mridha M, Hamid MA, Rana MM, Khan MEA, Ahmed MM, Sultan MT (2019) Semantic error detection and correction in Bangla sentence. In: 2019 Joint 8th international conference on informatics, electronics & vision (ICIEV) and 2019 3rd international conference on imaging, vision & pattern recognition (icIVPR). IEEE, pp 184–189
33. Islam S, Sarkar MF, Hussain T, Hasan MM, Farid DM, Shatabda S (2018) Bangla sentence correction using deep neural network based sequence to sequence learning. In: 2018 21st International conference of computer and information technology (ICCIT). IEEE, pp 1–6
34. Shajalal M, Aono M (2018) Semantic textual similarity in Bengali text. In: 2018 International conference on bangla speech and language processing (ICBSLP). IEEE, pp 1–5
35. Abujar S, Masum AKM, Chowdhury SMH, Hasan M, Hossain SA (2019) Bengali text generation using bi-directional rnn. In: 2019 10th International conference on computing, communication and networking technologies (ICCCNT). IEEE, pp 1–5
36. Rakib OF, Akter S, Khan MA, Das AK, Habibullah KM (2019) Bangla word prediction and sentence completion using gru: an extended version of rnn on n-gram language model. In: 2019 International conference on sustainable technologies for Industry 4.0 (STI). IEEE, pp 1–6
37. Islam MS, Mousumi SSS, Abujar S, Hossain SA (2019) Sequence-to-sequence Bangla sentence generation with lstm recurrent neural networks. Procedia Comput Sci 152:51–58
38. Pandit R, Sengupta S, Naskar SK, Dash NS, Sardar MM (2019) Improving semantic similarity with cross-lingual resources: a study in Bangla—a low resourced language. In: Informatics, vol 6. MDPI, p 19
39. Noshin Jahan M, Sarker A, Tanchangya S, Abu Yousuf M (2020) Bangla real-word error detection and correction using bidirectional lstm and bigram hybrid model. In: Proceedings of international conference on trends in computational and cognitive engineering: proceedings of TCCE 2020. Springer, pp 3–13
40. Chowdhury MAH, Mumenin N, Taus M, Yousuf MA (2021) Detection of compatibility, proximity and expectancy of Bengali sentences using long short term memory. In: 2021 2nd International conference on robotics, electrical and signal processing techniques (ICREST). IEEE, pp 233–237
41. Iqbal MA, Sharif O, Hoque MM, Sarker IH (2021) Word embedding based textual semantic similarity measure in Bengali. Procedia Comput Sci 193:92–101

42. Anbukkarasi S, Varadhaganapathy S (2022) Neural network-based error handler in natural language processing. Neural Comput Appl 66:1–10
43. Dhar AC, Roy A, Habib MA, Akhand M, Siddique N (2022) Transformer deep learning model for Bangla–English machine translation. In: Proceedings of 2nd international conference on artificial intelligence: advances and applications: ICAIAA 2021. Springer, pp 255–265
44. Aurpa TT, Sadik R, Ahmed MS (2022) Abusive Bangla comments detection on Facebook using transformer-based deep learning models. Soc Netw Anal Min 12(1):24
45. Bijoy MH, Hossain N, Islam S, Shatabda S (2022) Dpcspell: a transformer-based detector–purificator–corrector framework for spelling error correction of Bangla and resource scarce Indic languages. arXiv preprint arXiv:2211.03730
46. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30:66
47. Akil A, Sultana N, Bhattacharjee A, Shahriyar R (2022) Banglaparaphrase: a high-quality Bangla paraphrase dataset. arXiv preprint arXiv:2210.05109
48. Shahgir H, Sayeed KS (2023) Bangla grammatical error detection using t5 transformer model. arXiv preprint arXiv:2303.10612
49. Junczys-Dowmunt M, Grundkiewicz R, Dwojak T, Hoang H, Heafield K, Neckermann T, Seide F, Germann U, Aji AF, Bogoychev N et al (2018) Marian: fast neural machine translation in c++. arXiv preprint arXiv:1804.00344
50. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res 21(1):5485–5551