VAIYAKARANA: A Dataset for Automatic Grammar Error Detection and Correction in Bangla

Anonymous ACL submission

Abstract

Bangla (Bengali) is the fifth most spoken language globally, yet the problem of automatic grammar correction in Bangla is still in its nascent stage. This is primarily due to the need for a large corpus of grammatically incorrect sentences with corresponding correct counterparts. The present state-of-the-art techniques to curate a corpus for grammatically wrong sentences involve random swapping, insertion and deletion of words. However, these steps may only sometimes generate grammatically wrong sentences in Bangla. This work proposes a pragmatic noise injection approach to generate grammatically wrong sentences in Bangla. We first categorize the different kinds of errors in Bangla into 6 broad classes and 13 finer classes. We also collected 1,513 human-generated sentences from essays written by native Bangla speakers to understand error patterns in real-time. We used this data to generate grammatically wrong sentences systematically and statistically from a correct sentence. We provide a dataset, Vaiyākaraṇa, consisting of 22,648,708 grammatically incorrect sentences and 56,621,770 sentences. We evaluated our corpus against transformer models and LLMs and benchmarked it against human evaluators who are native speakers of Bangla. Our analysis shows that neural models match human evaluators with more and more data in detecting whether the sentence is grammatically correct. We have also evaluated LLMs' ability to generate grammatically correct sentences given a wrong sentence and paraphrase for Bangla in zero-shot mode and after instruction-tuning with Vaiyākaraṇa. This shows the efficacy of our Vaiyākaraṇa corpus. Our experiment shows that the performance of LLMs improved by 3% to 7% in $F_{0.5}$, Sacre-BLEU, and BERT-score after instruction-tuning with Vaiyākaraṇa, but they fall short of humans in correcting grammatically wrong sentences.

1 Introduction

Grammatical Error Correction (GEC) aims to automatically detect and correct grammatical (and other related) errors in a text. Thus, given the following sentence in English, "A ten year oldest boy go to school.", it will detect that there are errors in the use of superlative degree as well as in verb number and will correct it to "A ten-year-old boy goes to school.". GEC systems are very effective in writing assistants (Napoles et al., 2017), speech recognition systems (Wang and Zheng, 2020), etc. CoNLL-2013 (Ng et al., 2013) and CoNLL-2014 (Ng et al., 2014) put an impetus on the advancement of GEC works in English.

Bangla (Bengali, বাংলা Bāṁlā[1]) is the fifth most spoken language in the world and yet, to the best of our knowledge, there are only a few works in GEC for Bangla. Alam et al. (2007) proposed a rule-based statistical grammar checker for Bangla. However, the grammatical rules are basic, and consequently, they fail to perform even for mildly tricky sentences. One of the prerequisites of GEC for training-based methods is to have pairs of correct and corresponding wrong sentences. Islam et al. (2018) generated wrong sentences from a corpus of 250K sentences of Bangla by randomly inserting, deleting and swapping words from random positions. The prevalent methods, such as random word swapping, deletion, and insertion, may have worked well for English and other European languages, but since Bangla and Indian languages are mostly free word-order languages, these methods fail to always generate grammatically wrong sentences for Bangla. Moreover, words in Indian languages are equipped with kāraka (case) and vibhakti (inflection). Thus, in many sentences, all the six permutations of subject-verb-object are grammatically correct for

[1]We use ISO15919 transliteration scheme in this paper: https://en.wikipedia.org/wiki/ISO_15919

Indian languages, including Bangla. For example, consider the sentence অমর গীতাকে ভালোবাসে। (amara gītākē bhālōvāsē।, Amara loves Geeta.)[2] in Bangla as shown in Table A1 of Appendix A. The first five variants are the same words in different word order. The next three are word substitution, word deletion and word insertion variants. All these 8 variants are grammatically correct. Hence, simply using word operations is not enough, and a deeper look into grammar rules is necessary to generate wrong sentences. LLMs, including the instruction fine-tuning paradigm, have been highly successful in generating synthetic data for various downstream NLP tasks for English (Li et al., 2024; Jin et al., 2024). However, their performance with Bangla is not up to the mark, thereby making them unreliable and unusable. For example, on prompting "বাংলায় ভুল বাক্য গঠন করো" (bāṁlāẏa bhula vākya gaṭhana karō, "generate a wrong sentence in Bangla") to the GPT-3.5 Turbo model, it generates sentences such as শহরের গলির পাশে নদী পাওয়া যায়নি। (śaharēra galira pāśē nadī pāōẏā yāẏani।, No river was found by the side of the alleys of the city.) This sentence is grammatically correct.

Hence, in this paper, we have proposed an alternate but pragmatic approach to generate grammatically incorrect sentences for Bangla. We first categorize the types of grammar errors that are possible in Bangla. We then systematically inject these errors in a given correct sentence to generate corresponding wrong sentences. We have curated a corpus Vaiyākaraṇa of 56,621,770 with 22,648,708 number of erroneous sentences using this method and following the error statistics obtained 1,513 sentences described in Sec 4.1. We have also generated a smaller corpus Nirdeśa from Vaiyākaraṇa, with 226,507 sentences across 13 error categories and have evaluated their quality against both neural models and human evaluators. The performance of neural models matches that of human evaluators with more and more data.

We also evaluated the efficacy of Vaiyākaraṇa in building better GEC models for Bangla by probing the efficiency of LLMs to generate grammatically correct sentences for Bangla and on paraphrasing in Bangla. On average, Vaiyākaraṇa improves the performance of LLMs by 3%-7% on BERT-score and Sacre-BLEU from the zero-shot paradigm on both the tasks. However, humans signficantly outperform them in correcting grammatically wrong sentences.

In sum, our contributions in this paper are:

1. We categorized grammatical errors into 13 categories (Sec. 3). To the best of our knowledge, this is the first attempt at such an extensive categorization of grammatical errors for Bangla.
2. We also collected 1,513 sentences through human generation and identified the errors made by humans in real-time. We also evaluated the percentage of each category of error that humans make in real-time (Sec 4.1).
3. We provide a pragmatic error generation approach (Sec. 4) to generate any number of grammatically erroneous sentences for Bangla using the statistics obtained from Sec 4.1.
4. Using the above approach, we generated Vaiyākaraṇa, a dataset of 22,648,708 grammatically inaccurate sentences for Bangla across the 13 error categories. We have also created a small dataset from Vaiyākaraṇa, Nirdeśa and converted it to alpaca format which will be essential for instruction-tuning LLMs for GEC in Bangla.
5. We benchmarked Vaiyākaraṇa against neural models and native human speakers of Bangla (Sec. 5) in the Grammar Error Detection (GED) task.
6. We evaluated the efficacy of Vaiyākaraṇa in building better GEC models for Bangla (Sec: 6).

2 Related Work

The GEC work for Indian languages is in its nascent stages. Sonawane et al. (2020) focused on categorizing and generating various kinds of inflectional errors for GEC for Hindi, which can also be employed for other Indian languages. S. et al. (2023) proposed Vyakranly, a Hindi translation and Grammar error detection toolkit for Hindi. The authors employed statistical and handwritten rules to detect and correct grammatical errors in Hindi. The spelling error detection task was done at the word level on a corpus of 19,500 words. Alam et al. (2007) proposed a rule-based statistical model for Bangla that detects whether a sentence is correct or not. Islam et al. (2018) generated wrong sentences from a corpus of 250K sen-

[2]For every Bangla sentence, we show its transliteration in ISO15919 and translation in English.

tences of Bangla by randomly inserting, deleting and swapping words at random positions. Their main idea was also to detect whether a sentence is grammatically correct. Rahman et al. (2023) proposed a CNN-based spelling error detection and correction model for Bangla on the sentence level. Oshin et al. (2023) provided a novel dataset for Bangla text error classification. The dataset consists of 10,000 sentences from YouTube videos across four error categories of which only 3,140 sentences were erroneous.

Hossain et al. (2023) proposed Panini, a Vaswani-style monolingual transformer-based method for GEC correction in Bangla. In addition, they also curated a 7.7M+ sentences baseline corpus for Bangla GEC synthetically. The sentences are classified into 10 broader classes. In contrast, we classified grammatical errors into 13 classes, including tense errors, Gurucaṇḍālī Dōṣa and Spelling Errors Non-Dictionary and Missing Word Errors (including verb missing errors). We conducted a user survey and collected 1,513 sentences to gather information about errors made by native and non-native Bangla speakers in real-time. The survey revealed that tense errors, Gurucaṇḍālī Dōṣa, and Spelling Errors in Non-dictionary are common errors in real-time. Hence, they need to be appropriately handled for Bangla GEC. Based on this manual data, we curated Vaiyākaraṇa a benchmark dataset 56,621,770 sentences covering sentences from the 18th to the 21st century.

Generating Methodologies: Sennrich et al. (2016) proposed back-translation methodology for data augmentation for machine translation which (Rei and Yannakoudakis, 2017) used for Grammar Error Detection whereas Zhou et al. (2020) used round-trip translation, a variant of back-translation, to synthesise noisy sentences using bridge language, e.g., English-Chinese-English, where Chinese is the bridge language. Mita et al. (2020) and Rothe et al. (2021) focussed on augmenting official GEC datasets to generate new datasets. Bangla lacks such an extensive GEC dataset covering all categories of errors or machine translation datasets with bridge languages. Moreover, all these procedures do not focus on generating wrong sentences for specific error categories, which is our primary objective. Sec 4.1 shows that not all categories are equally probable to occur in real-time, whereas Sec 7 shows that not all categories of errors are equally difficult to detect and correct. Hence, we chose the noise injection and error pattern induction methodology discussed by (Bryant et al., 2023) to generate noisy sentences.

GEC works for other languages are presented in Appendix B.

3 Grammar Error Categories

Automatic Grammar Error Correction (GEC) is a relatively unexplored territory for Bangla. The prevalent works did not concentrate on categorising most of the grammatical error types, such as word errors, Gurucaṇḍālī Dōṣa[3] and others. In this section, we try to formally categorise grammatical errors in Bangla based on grammar. We mostly follow the categories described in one of the most well-known grammar books of Bangla (Chakroborty, 2018).

The grammatical errors for Bangla can be classified into 6 broader classes. These classes as well as the 13 finer distinctions within them are described next. A sentence may contain multiple errors of one class or different classes as well. Table 1 lists example sentences[4] of the error classes. Appendix C explains the category of errors with examples.

4 Corpus

We aim to generate a corpus for automatic grammar correction in Bangla. We first curate a dataset by collecting hand-written sentences from Bangla speakers. However, since number of hand-written sentences cannot be scaled beyond a point, we also follow a rule-based approach for systematically generating erroneous sentences by injecting errors in correct sentences. We next discuss both these approaches.

4.1 Manual Generation

Our aim for manual generation is to get hand-written data on grammatical errors made by native Bangla speakers during writing. We organized a survey asking participants to write an essay on a particular topic. Each participant was expected to write an essay within 30 minutes comprising at least 15 sentences and 150 words on a topic chosen by her from a set of choices. The survey took place in a proctored environment to generate an

[3]This is a special kind of error, found in Bangla, as explained later.

[4]The text in violet shows the erroneous portion of a sentence corresponding to the correct text in blue.

Error Class	Sub-class	Example of Wrong Sentence (in violet) followed by Correct Sentence (in blue)
Spelling	Non-Dictionary	আমি কারখানায় কাড করি। (āmi kārakhānāẏa kāḍa kari।, I <non-word> in factory.) আমি কারখানায় কাজ করি। (āmi kārakhānāẏa kāja kari।, I work in factory.)
	Dictionary	আমি কাল বারি যাব। (āmi kāla vāri yāva।, I will go water tomorrow.) আমি কাল বাড়ি যাব। (āmi kāla vāṛi yāva।, I will go home tomorrow.)
		আমি কাল শাড়ি যাব। (āmi kāla śāṛi yāva।, I will go saree tomorrow.) আমি কাল বাড়ি যাব। (āmi kāla vāṛi yāva।, I will go home tomorrow.)
Word	Tense	আমি গতকাল পড়াশোনা করব। (āmi gatakāla paṛāśōnā karava।, I will study yesterday.) আমি গতকাল পড়াশোনা করেছিলাম। (āmi gatakāla paṛāśōnā karēchilāma।, I studied yesterday.)
		যখন শীত আসে তখন ফুল ফুটেছিল। (yakhana śīta āsē takhana phula phuṭēchila।, When winter comes, flowers bloomed.) যখন শীত আসে তখন ফুল ফোটে। (yakhana śīta āsē takhana phula phōṭē।, When winter comes, flowers bloom.)
	Person	আমি কারখানায় কাজ করে। (āmi kārakhānāẏa kāja karē।, I works in factory.) আমি কারখানায় কাজ করি। (āmi kārakhānāẏa kāja kari।, I work in factory.)
	Number	আমি এখানে চারজন থাকি। (āmi ēkhānē cārajana thāki।, I four stay here.) আমরা এখানে চারজন থাকি। (āmarā ēkhānē cārajana thāki।, We four stay here.)
	Gender	উত্তম একজন অসাধারণ অভিনেত্রী। (uttama ēkajana asādhāraṇa ābhinētrī।, Uttam is an outstanding actress.) উত্তম একজন অসাধারণ অভিনেতা। (uttama ēkajana asādhāraṇa abhinētā।, Uttam is an outstanding actor.)
	Case	আমি রান্নাঘরকে ভাত খাই। (āmi rānnāgharakē bhāta khāi।, I eat rice to kitchen.) আমি রান্নাঘরে ভাত খাই। (āmi rānnāgharē bhāta khāi।, I eat rice in kitchen.)
	Parts-of-Speech	হিমালয়ের সুন্দর অবিস্মরণীয়। (himālaẏēra sundara avismaraṇīẏa।, The beautiful of Himalaya is unforgettable.) হিমালয়ের সৌন্দর্য অবিস্মরণীয়। (himālaẏēra saundarya avismaraṇīẏa।, The beauty of Himalaya is unforgettable.)
	Missing	আমি কাল বাড়ি •। (āmi kāla vāṛi •।, I • home tomorrow.) আমি কাল বাড়ি যাব। (āmi kāla vāṛi yāva।, I will go home tomorrow.)
		উত্তম একজন অসাধারণ •। (uttama ēkajana asādhāraṇa •।, Uttam is an outstanding •.) উত্তম একজন অসাধারণ অভিনেতা। (uttama ēkajana asādhāraṇa abhinētā।, Uttam is an outstanding actor.)
Gurucaṇḍālī Dōṣa		নন্দবাবু ইহা লক্ষ্য করেছেন। (nandavāvu ihā lakṣya karēchēna।, Nanda has noticed this.) নন্দবাবু ইহা লক্ষ্য করিয়াছেন। (nandavāvu ihā lakṣya kariya□āchēna।, Nanda has noticed this.)
		নন্দবাবু ইহা লক্ষ্য করেছেন। (nandavāvu ihā lakṣya karēchēna।, Nanda has noticed this.) নন্দবাবু এটা লক্ষ্য করেছেন। (nandavāvu ēṭā lakṣya karēchēna।, Nanda has noticed this.)
Punctuation		আমি গতকাল পড়াশোনা করেছিলাম? (āmi gatakāla paṛāśōnā karēchilāma?, I studied yesterday?) আমি গতকাল পড়াশোনা করেছিলাম। (āmi gatakāla paṛāśōnā karēchilāma।, I studied yesterday.)
Semantic		মানস আকাশ খেতে ভালোবাসে। (mānasa ākāśa khētē bhālōvāsē।, Manas loves to eat the sky.) মানস আকাশ দেখতে ভালোবাসে। (mānasa ākāśa dēkhatē bhālōvāsē।, Manas loves to see the sky.)
		মানস আকাশ খেতে ভালোবাসে। (mānasa ākāśa khētē bhālōvāsē।, Manas loves to eat the sky.) মানস মাছ খেতে ভালোবাসে। (mānasa mācha khētē bhālōvāsē।, Manas loves to eat fish.)

Table 1: Grammatical Error Types in Bangla

exam-like situation; this enabled us to gather live data (with errors) on Bangla. We gathered 1,513 sentences and 17,056 words from 76 essays written by 36 participants (32 participant wrote 2 essays each, whereas 4 participants wrote 3 essays) . The maximum length of a sentence is 69 words whereas the minimum length is 1. The details of the topic and participants are provided in Appendix F. Of the 1,513 sentences written, 523 sentences (∼35%) were grammatically incorrect. Out of these, 401 (∼77%) had single errors, while 122 (∼23%) sentences had multiple errors. Of these, 92 contained 2 errors (∼18%) and 21 contained 3 errors (∼4%). The remaining 9 sentences contained up to 7 errors. A total of 678 (∼4%) words were erroneous in these 523 sentences.

Table 2 shows the number of errors of each category described in Table 1 along with their percentage of occurrence with respect to the total number of erroneous words. Spelling mistakes are the most common type of error, occurring more than 64% of the time. Further investigation reveals that more than 45% of spelling errors are due to the mixing of the characters ‘ন’(n)/‘ণ’(ṇ); ‘র’(r)/‘ড়’(ṛ)/‘ঢ়’(ṛh); and ‘স’(s)/‘শ’(ś)/‘ষ’(ṣ). The mixing of “কি” (ki, whether) versus “কী” (kī, what) is the most prominent type of dictionary-based word spelling error. Tense, person, gender, Number, Gender and Semantic errors are not that frequent. Case and Punctuation errors are quite common, though. We also noticed that multiple-error sentences mostly contain multiple spelling errors. Occurrence of (spelling, punctuation) and (spelling, case, punctuation) are also common. As many as 6 different types of error occurred together (spelling (both types), case, missing word, Gurucaṇḍālī Dōṣa, punctuation).

4.2 Vaiyākaraṇa

We collected 56,621,570 sentences from the Vācaspati (Bhattacharyya et al., 2023) corpus and 200 sentences from a well-known grammar book (Chakroborty, 2018). The corpus was curated to cover the temporal and spatial variations of Bangla. The 200 sentences from the grammar book make the dataset more grammatically and

Error Class	#Occurences	Percentage
Non-Dictionary	338	49.85%
Dictionary	87	12.83%
Spelling Errors	425	62.68%
Tense Errors	15	2.21%
Person Errors	13	1.92%
Number Errors	2	0.29%
Gender Errors	0	0.00%
Case Errors	81	11.95%
POS Errors	14	2.06%
Missing Words	32	4.72%
Word Errors	157	20.94%
Punctuation Errors	78	11.50%
Semantic Errors	1	0.15%
Gurucaṇḍālī Dōṣa	17	2.51%
Total	678	100.00%

Table 2: Grammatical errors in manual annotation

Error Class	#Sentences	Total_Percentage	Error_Cat Percentage
Non-Dictionary	11,324,354	20.00	50.00
Dictionary Spelling	2,604,601	4.60	11.50
Spelling Error	13,928,955	24.60	61.50
Tense Error	498,271	0.88	2.20
Person Error	452,975	0.80	2.00
Number Error	2,000	0.004	0.009
Gender Error	100	0.0002	0.0004
Case Error	2,604,602	4.60	11.50
POS Error	462172	0.82	2.04
Missing Word	1,069,020	1.88	4.72
Word Error	50,89,140	8.98	22.47
Punctuation Errors	2,604,602	4.60	11.50
Semantic Errors	100	0.0002	0.0004
Gurucaṇḍālī Dōṣa	640,182	1.13	2.82
Multiple-Errors Different Categories	385,729	0.68	1.70
Incorrect	*22648708*	40.00	100.00
Correct	*33,973,062*	60.00	0.00
Total	56,621,770	100.00	100.00

Table 3: Grammatical Error in Vaiyākaraṇa

linguistically enriched. It also serves as a gold standard sentence for generating errors. We manually typed these sentences to avoid any error. We, thus, curated a dataset with 56,621,770 sentences. We followed the data cleaning and pre-processing steps as described in Appendix D to make the curated dataset suitable enough for error generation.

We injected noise in these sentences following error patterns in Sec 4.1 and curated Vaiyākaraṇa with 22,648,708 erroneous sentences and 56,621,770 total sentences. For homonym, POS, tense, Person, and Case errors, we mostly replaced the correct word with its corresponding wrong word (we collected all such pair of words present in the book (Chakroborty, 2018)). While replacing, we monitored that for generating tense and Gurucaṇḍālī Dōṣa errors, there are at least 2 verbs and 2 verbs or pronouns present in the sentence. We do not replace at least one verb in both cases to generate erroneous sentences. The detailed steps taken to inject noise for different kinds of errors are described in Appendix E.

Table 3 shows the number of sentences for each category of errors (inclusive of the manual data). We have conducted a chi-square test for both the broad and finer error categories in Vaiyākaraṇa, and the resulting p-value is < 0.01. Hence, we can conclude that the distribution of error categories in names is in accordance with real-world errors. A total of 21,430,458 contains single error in the sentences whereas 1,218,250 sentences contains multiple errors. Table A3 shows the distribution of multiple errors in Vaiyākaraṇa. The maximum number of errors in a sentence in Vaiyākaraṇa is 10.

Although we focused on generating Vaiyākaraṇa for Bangla, the aforementioned procedures of injecting noise to generate grammatically wrong sentences can also be applied to other Indian languages with little or no modification.

5 Evaluation

5.1 Machine Evaluation

We benchmarked Vaiyākaraṇa against the transformer-based models and LLMs in Grammatical Error Detection in bangla (GED). The models include *monolingual* models such as BanglaBERT (Bhattacharjee et al., 2022), VĀC-BERT (Bhattacharyya et al., 2023) and sahajBERT as well as *multilingual models like* IndicBERT (Doddapaneni et al., 2023) and MuRIL (Khanuja et al., 2021). We have also evaluated Vaiyākaraṇa against multilingual LLMs such as GPT-4 (OpenAI et al., 2023), GPT-Neo (Black et al., 2022), Bloom 1.1B (Workshop et al., 2023). Each model is run 5 times for 20 iterations with different seed values, and mean and standard deviations are reported in Table 4. Since standard deviations are low, the highest was not very different from the mean. We have also reported the macro-F1 score achieved by Random Forest classifier, a non-neural model on the categorization task in Table 4 and BanglaT5 (Hossain et al., 2023). Hyperparameter details of all the models are

Model	Parameters	Binary	Broad	Finer
		Mean	Mean	Mean
IndicBERTv2.0	117M	85.05±0.01	83.20±0.05	80.90±0.03
MuRIL	236M	80.13±0.02	78.22±0.02	75.78±0.02
BanglaBERT	110M	85.66±0.02	83.54±0.02	81.24±0.02
VAC-BERT	17M	84.87±0.03	82.84±0.03	80.72±0.02
sahajBERT	17M	80.60±0.07	78.44±0.05	76.30±0.
RandomForest	0.42M	47.00±0.55	38.50±0.74	32.30±1.25
BanglaT5	290M	88.90±0.10	84.50±0.68	82.48±0.14
Human	12	**88.30**±3.46	**84.20**±3.75	**82.30**±3.90

Table 4: Macro-F1 of transformer models on Vaiyākaraṇa.

provided in Appendix O. Unless otherwise mentioned, the *macro-F1 score* is used as the evaluation metric for all the classification tasks.

5.1.1 Error Classification Tasks

We evaluated the performance of neural models to detect different error categories in three classification tasks. The first is a *binary* class, simply indicating if a given sentence is grammatically correct or wrong. The first *multi-class* task is for the 6+1 broad classes (ignoring the sub-classes), while the last one attempts to classify into the finer classes as well (13+1 classes)..

5.1.2 Results

Table A4 of Appendix H shows the performance of all neural models and Random Forest classifier on the 56,621,770 sentences in Vaiyākaraṇa.

BanglaT5 achieves the max mean and highest macro-F1 score for both the multi-class classification tasks and binary classification tasks.

We have conducted an ablation study to evaluate the performance of BanglaT5 in detecting various types of errors in Bangla with a change in the number of erroneous sentences for each class in Sec 7. The result indicates that the transformer models are as good as humans in detecting error categories in Bangla (maximum macro-F1 of human is more than that of BanglaT5).

5.2 Prompts in LLMs

In this section, we discuss the ability of LLMs to detect and correct Bangla grammatical errors. We sub-sampled Vaiyākaraṇa and created a corpus Nirdeśa of 566,238 in accordance with the occurrence of errors obtained in Sec 4.1. We created Nirdeśa to avoid memory shortage to instruction-tune LLMs. We have performed chi-square testing on this dataset, Nirdeśa, and the p-value is 0.043. We converted Nirdeśa into Alpaca-JSON format for instruction tuning LLMs like GPT-Neo, GPT-2, GPT-4 and BLOOM. The details of Nirdeśa is provided in Appendix L. We instruction tuned the LLMs using the following prompts "বাক্যটি ঠিক না ভুল তা নির্ধারণ কর।" (vākyaṭi ṭhika nā bhula tā nirdhāraṇa kara।, is this sentence grammatically correct?) and "ব্যাকরণ সংশোধন কর।" (vyākaraṇa saṁśōdhana kara।, grammatically correct this sentence) and evaluated their responses against the ground truth on the human evaluated 600-sentence set mentioned in Sec. 5.3. Table 5 shows the results of prompts with and without instruction-tuning the LLMs. Example sentences and responses to the prompts are shown in Appendix J. An in-depth analysis of the responses shows that the LLMs perform decently in identifying tense, person and gender errors (∼45%). However, their performance drops to below 20% for dictionary-based spelling errors (homonyms), POS errors and Gurucaṇḍālī Dōṣa which increases to nearly 30% after instruction-tuning GPT-4. This result highlights the limitations of directly using LLMs for GEC in Bangla.

5.3 Human Evaluation

To evaluate our benchmark dataset more rigorously, we next did a human evaluation of the same. We built an interface where given a sentence, a participant marks the error class (including the correct one). We took help of 12 Bangla speakers, each of whom was provided with a set of 50 sentences. This was selected randomly from a set of 2500 sentences from Vaiyākaraṇa, comprising 650 correct sentences and 1850 wrong sentences. There was no overlap between the set of sentences that each participant received. The mean macro-F1 scores achieved by 12 humans for binary, broad classes, and finer classes are 81.00%, 68.50% and 57.33% respectively, while the highest macro-F1 scores by a human are 90.50%, 87.50% and 83.33% respectively. We tested the transformer models against these 12×50=600 sentences under the setting described in Sec 5.1. Table A4 shows the results of various transformer models on the 600 manually evaluated sentences.

Additionally, we have conducted another survey with 12 people (5 persons are overlapping with the previous set of 12 annotators) and 600 sentences (the same sentences used for LLM evaluation and human categorization). Each annotator was given a set of 50 random sentences and was asked to mark them as right or wrong. In addition, if they felt a sentence was wrong, they were asked to provide a correct variant of the sentence with

Model	Without Instruction Tuning		With Instruction Tuning	
	1st Prompt	2nd Prompt	1st Prompt	2nd Prompt
GPT-3.5	28.30	29.70	40.60	41.80
GPT-4.0	30.30	32.60	43.30	47.50
OPT	26.50	27.20	35.30	36.60
GPT-Neo	26.38	27.58	27.90	30.60
BLOOM	25.00	25.50	29.00	29.50

Table 5: Macro-F1 of different LLMs for prompts with and without instruction tuning for 600 sentences.

as few changes to the original sentence as possible. The maximum score achieved is 47 out of 50, with an average of 39.3 with a standard deviation of 5.66. Most corrected variants (∼92%) matched the gold standard sentence. However, some correct answers deviated more, and the maximum deviation was 4 words (average sentence length is around 10 words).

5.4 Erroneous Word Detection

We have also tried to probe into the ability of neural models for GED (Grammatical Error Detection) for Bangla. In the earlier section, we tried to assess the ability to categorise different types of errors in the sentence. In this section, we tried to evaluate the ability to detect erroneous words in the sentences. Detecting erroneous words correctly will pave the way for making better GEC models. While injecting noise to generate Vaiyākaraṇa, we also marked the words that have been changed and recorded their positions, which will help us in conducting this experiment. We treated this problem as a *sequence pair classification* task as done by (Bhattacharyya et al., 2023) where a sentence is passed as one sequence, and every word in it is passed as the second sequence to determine whether it is contextually correct on the Nirdeśa dataset. Table 6 shows the performance of transformer models on the spelling error detection task. BanglaT5 (79.26%) has the highest macro-F1 score. We conducted this experiment with GPT-4 also both at zero-shot and after instruction tuning with Nirdeśa. We used the same 20,000 sentences used in Sec:4.1, which is not part of the Nirdeśa dataset. The macro-F1 score for GPT-4 increases from 32.36% at zero-shot level to 51.34% after instruction tuning with Nirdeśa and prompt "বাক্যে ভুল শব্দগুলো চিহ্নিত কর।" (vākyē bhula śavdagulō cihnita kara.). The model achieves 25.36% macro-F1 score in detecting all the errors in sentences with multiple erroneous words (they are able to detect one or two but not all.

Model	Macro-F1
IndicBERT-v2.0	77.90
MuRIL	52.68
sahjBERT	70.45
BanglaBERT	78.60
VAC-BERT	77.50
BanglaT5	79.26

Table 6: Macro-F1 different transformer models for GED task in Bangla.

6 Corpus Efficacy

In the previous section, we evaluated capabilities of different models to detect and categorize grammatically wrong sentences. In this section, we demonstrate the efficacy of Vaiyākaraṇa in building better GEC models for Bangla.

We created a test corpus of 52,100 sentences (details of the sentences in Appendix K) which were not present in Vaiyākaraṇa and tested the ability of the models to generate correct grammatical sentences in two conditions, one without instruction tuning the models with Nirdeśa (zero-shot), and the other after instruction tuning the models with Nirdeśa. Table 7 shows the performance of models in generating grammatically correct sentences with and without Nirdeśa. There is an average increase of more than 5% on $F_{0.5}$ score as well as bert-score (details about evaluation metric is in Appendix M). Since BERT architectures are not able to generate sentences we have conducted this experiment with LLMs (GPT-4, GPT-Neo) and BanglaT5. In all the architectures the average $F_{0.5}$ and Bert-score are increasing from the zero-shot paradigm.

To ascertain that Vaiyākaraṇa indeed is the chief contributing factor for this improvement in performance, we tested the performance of LLMs models for paraphrasing before and after fine-tuning with Nirdeśa as conducted by (Hossain et al., 2023). The paraphrasing dataset is developed by (Akil et al., 2022) and contains 5,763 sentences. Table 8 shows the performance of models in paraphrasing with and without Nirdeśa. The performance of models is enhanced by 3-7% after fine-tuning with Nirdeśa. These experiments prove the mettle of Vaiyākaraṇa as a corpus for building better GEC models for Bangla.

7 Ablation Study

We conducted an experiment to understand the ability of neural models to detect different category of errors in Bangla.The experiments are con-

Model	Zero-Shot			With Fine-tuning		
	$F_{0.5}$	BERT-score	Sacre-BLEU	$F_{0.5}$	BERT-score	Sacre-BLEU
GPT-4	69.25	71.85	36.40	72.57	76.86	41.45
GPT-2	67.03	67.86	32.90	69.27	73.6	40.10
BLOOM	63.43	65.86	30.86	65.77	69.24	36.60
BanglaT5	65.17	66.82	31.67	67.7	71.5	38.90

Table 7: Performance of models with and without fine-tuning with Nirdeśa on grammatically correct sentence generation

Model	Zero-Shot		With Fine-tuning	
	Sacre-BLEU	BERT-score	Sacre-BLEU	BERT-score
GPT-4	49.00	62.70	57.60	66.80
GPT-2	46.38	57.58	50.74	62.41
BLOOM	45.50	55.86	49.00	59.50
BanglaT5	48.66	57.97	55.38	62.53

Table 8: Performance of models with and without fine-tuning with Nirdeśa on paraphrasing

ducted at various levels: with fewer than 100, 1,000, 5,000, 10,000, 50,000, 100,000 and greater than 100,000 erroneous sentences of a particular category. In addition to the erroneous sentences each test set contains an additional 60% correct sentences as observed from the human survey in Sec: 4.1. We have conducted chi-square testing for all the test sets where the probabilities of correct sentence and wrong sentence are in the ratio of 6:4 and achieved p-value < 0.05. Figure 1 shows the macro-F1 score of each class of errors at different erroneous sentence levels. The X-axis denotes the different categories of errors, and the Y-axis denotes the macro-F1 score of that category of error at different sub-samples of error sentences for BanglaT5 (best performing model). The figure shows that with an increase in the number of samples, the macro-F1 score increases. This phenomenon is observed even for sentences with multiple categories of errors. From the figure, we can conclude that it is relatively easy to detect tense, missing word and person errors, whereas spelling error-dictionary, Gurucaṇḍālī Dōṣa and multiple errors are challenging to detect. It indicates the effectiveness of the corpus in detecting different types of errors, which is essential for building explainable GEC models in Bangla.

Figure 7 in Appendix N shows the macro-F1 score of the broad and finer category of errors at different erroneous sentence levels. They are also sub-sampled based on statistics from Sec: 4.1 are chi-square tested with p-value< 0.05.

Fig 2 shows the performance of GPT-4 in detecting different category of errors before and after instruction tuning with Nirdeśa at different sub-samples of erroneous sentences.

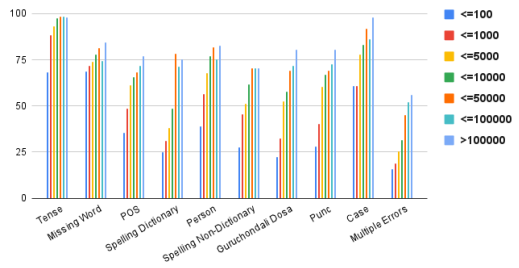

Figure 1: Figure showing performance of BanglaT5 on different error categories in Bangla.

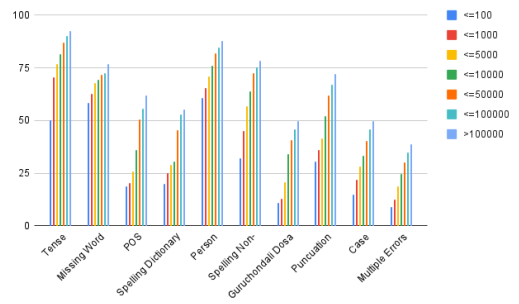

Figure 2: Figure showing performance of GPT-4 on different error categories in Bangla.

8 Discussions and Future Work

In this paper, we proposed a rule-based noise injection methodology for generating grammatically wrong sentences in Bangla. We have generated erroneous sentences across 13 categories in Bangla, which is the most extensive categorization of grammatical errors for Bangla. We curated Vaiyākaraṇa consisting of 22,648,708 wrong and 56,621,770 sentences. We also collected a set of 1,513 sentences (523 being wrong) from manually written essays. The results show that the neural models perform similar to the human evaluators in detecting error categories and words whereas human outperforms neural models in correcting grammatically wrong sentences. We compared the performance of LLMs in generating grammatically correct sentence and paraphrasing in Bangla in zero-shot and after instruction-tuning with Vaiyākaraṇa. The performance of LLMs improve by 3%-7% on both the tasks.In future, we would like to use this dataset to develop better GEC models with explainability. Further, we believe that this methodology can be applied to generate benchmarks for other Indian languages.

9 Limitations

Curating a large quality benchmark for GEC requires a good quality lemmatizer and POS tagger. Bangla suffers from a lack of quality lemmatizers and POS taggers. Hence, we had to resort to manually adding words available from a grammar book (Chakroborty, 2018).

Also, hand-written Bangla data is not readily available. We conducted a survey and collected 1,513 hand-written sentences.

Finally, while the 12 human evaluators are all native speakers of Bangla, evaluating against Bangla grammarians could have given us more insights into the process. We are planning to do that in the future.

10 Ethics Statement

The Vaiyākaraṇa benchmark is curated by merging sentences from Vacaspati corpus (Bhattacharyya et al., 2023) and (Chakroborty, 2018). The authors of Vacaspati provided us with the corpus, and (Chakroborty, 2018) is publicly available. Hence, there is no copyright infringement in curating Vaiyākaraṇa. We have made efforts to ensure that Vaiyākaraṇa is also devoid of any objectionable statements. We have also conducted a manual essay writing survey for gathering real word errors. The participants have kindly allowed us to use their essays for research purpose. We will release Vaiyākaraṇa, the Alpaca format of Vaiyākaraṇa, manual hand-witten data and code for the rule-based noise injection methodology upon acceptance of the paper under a non-commercial license.

References

Ajwad Akil, Najrin Sultana, Abhik Bhattacharjee, and Rifat Shahriyar. 2022. BanglaParaphrase: A high-quality Bangla paraphrase dataset. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 261–272, Online only. Association for Computational Linguistics.

Md. Jahangir Alam, Naushad UzZaman, and Mumit Khan. 2007. N-gram based statistical grammar checker for bangla and english.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad Uddin, Kazi Mubasshir, Md. Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. Banglabert: Lagnuage Model Pretraining and Benchmarks for Low-Resource Language Understanding Evaluation in Bangla. In *Findings of the North American Chapter of the Association for Computational Linguistics: NAACL 2022*.

Pramit Bhattacharyya, Joydeep Mondal, Subhadip Maji, and Arnab Bhattacharya. 2023. VACASPATI: A diverse corpus of Bangla literature. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1130, Nusa Dua, Bali. Association for Computational Linguistics.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. Gpt-neox-20b: An open-source autoregressive language model. *Preprint*, arXiv:2204.06745.

Adriane Boyd. 2018. Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, page 159.

Bamandev Chakroborty. 2018. *Uchchatara Bangla Byakaran*, volume 2nd. Akshay Malancha.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton,

Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Sam Davidson, Aaron Yamada, Paloma Fernandez Mira, Agustina Carando, Claudia H. Sanchez Gutierrez, and Kenji Sagae. 2020. Developing NLP tools with a new corpus of learner Spanish. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7238–7243, Marseille, France. European Language Resources Association.

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

Nahid Hossain, Mehedi Hasan Bijoy, Salekul Islam, and Swakkhar Shatabda. 2023. Panini: a transformer-based grammatical error correction method for bangla. *Neural Comput. Appl.*, 36(7):34633477.

Budi Irmawati, Hiroyuki Shindo, and Yuji Matsumoto. 2017. Generating artificial error data for indonesian preposition error corrections. *International Journal of Technology*, 8(3):549–558.

Sadidul Islam, Mst. Farhana Sarkar, Towhid Hussain, Md. Mehedi Hasan, Dewan Md Farid, and Swakkhar Shatabda. 2018. Bangla sentence correction using deep neural network based sequence to sequence learning. In *2018 21st International Conference of Computer and Information Technology (ICCIT)*, pages 1–6.

Yiqiao Jin, Mohit Chandra, Gaurav Verma, Yibo Hu, Munmun De Choudhury, and Srijan Kumar. 2024. Better to ask in english: Cross-lingual evaluation of large language models for healthcare queries. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 26272638, New York, NY, USA. Association for Computing Machinery.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual Representations for Indian Languages. *arXiv preprint*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Katerina Korre and John Pavlopoulos. 2022. Enriching grammatical error correction resources for Modern Greek. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4984–4991, Marseille, France. European Language Resources Association.

Myunghoon Lee, Hyeonho Shin, Dabin Lee, and Sung-Pil Choi. 2021. Korean grammatical error correction based on transformer with copying mechanisms and grammatical noise implantation methods. *Sensors*, 21(8).

Zihao Li, Yucheng Shi, Zirui Liu, Fan Yang, Ali Payani, Ninghao Liu, and Mengnan Du. 2024. Quantifying multilingual performance of large language models across languages. *Preprint*, arXiv:2404.11553.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.

Masato Mita, Shun Kiyono, Masahiro Kaneko, Jun Suzuki, and Kentaro Inui. 2020. A self-refinement strategy for noise reduction in grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 267–280, Online. Association for Computational Linguistics.

Jakub Náplava and Milan Straka. 2019. CUNI system for the building educational applications 2019 shared task: Grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 183–190, Florence, Italy. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European*

Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Mitodru Niyogi and Arnab Bhattacharya. 2024. Paramanu: A family of novel efficient indic generative foundation language models. *Preprint*, arXiv:2401.18034.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, and Paul Baltescu et al. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Nabilah Oshin, Syed Hoque, Md Fahim, Amin Ahsan Ali, M Ashraful Amin, and Akmmahbubur Rahman. 2023. BaTEClaCor: A novel dataset for Bangla text error classification and correction. In *Proceedings of the First Workshop on Bangla Language Processing (BLP-2023)*, pages 124–135, Singapore. Association for Computational Linguistics.

Chowdhury Rahman, MD.Hasibur Rahman, Samiha Zakir, Mohammad Rafsan, and Mohammed Eunus Ali. 2023. BSpell: A CNN-blended BERT based Bangla spell checker. In *Proceedings of the First Workshop on Bangla Language Processing (BLP-2023)*, pages 7–17, Singapore. Association for Computational Linguistics.

Gaoqi Rao, Qi Gong, Baolin Zhang, and Endong Xun. 2018. Overview of NLPTEA-2018 share task Chinese grammatical error diagnosis. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–51, Melbourne, Australia. Association for Computational Linguistics.

Marek Rei and Helen Yannakoudakis. 2017. Auxiliary objectives for neural error detection models. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 33–43, Copenhagen, Denmark. Association for Computational Linguistics.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Rachel S., Vasudha S., Shriya T., Rhutuja K., and Lakshmi Gadhikar. 2023. Vyakranly: Hindi grammar & spelling errors detection and correction system. In *2023 5th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, pages 1–6.

Sagor Sarker. 2021. BNLP: Natural language processing toolkit for Bengali language. *arXiv preprint*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Aiman Solyman, Zhenyu Wang, Qian Tao, Arafat Abdulgader Mohammed Elhag, Rui Zhang, and Zeinab Mahmoud. 2022. Automatic arabic grammatical error correction based on expectation-maximization routing and target-bidirectional agreement. *Knowledge-Based Systems*, 241:108180.

Ankur Sonawane, Sujeet Kumar Vishwakarma, Bhavana Srivastava, and Anil Kumar Singh. 2020. Generating inflectional errors for grammatical error correction in Hindi. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 165–171, Suzhou, China. Association for Computational Linguistics.

Oleksiy Syvokon, Olena Nahorna, Pavlo Kuchmiichuk, and Nastasiia Osidach. 2023. UA-GEC: Grammatical error correction and fluency corpus for the Ukrainian language. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 96–102, Dubrovnik, Croatia. Association for Computational Linguistics.

Lihao Wang and Xiaoqing Zheng. 2020. Improving grammatical error correction models with purpose-built adversarial examples. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2858–2869, Online. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. *Preprint*, arXiv:2109.01652.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ili, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, and Pawan Sasanka Ammanamanchi et al. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. 2020. Improving grammatical error correction with machine translation pairs. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 318–328, Online. Association for Computational Linguistics.

Original sentence	অমর গীতাকে ভালোবাসে। amara gītākē bhālōvāsē।
Word order 1	গীতাকে অমর ভালোবাসে। gītākē amara bhālōvāsē।
Word order 2	গীতাকে ভালোবাসে অমর। gītākē amara bhālōvāsē।
Word order 3	অমর ভালোবাসে গীতাকে। gītākē amara bhālōvāsē।
Word order 4	ভালোবাসে অমর গীতাকে। gītākē amara bhālōvāsē।
Word order 5	ভালোবাসে গীতাকে অমর। gītākē amara bhālōvāsē।
Word substitution	শ্যামল গীতাকে ভালোবাসে। śyāmala gītākē bhālōvāsē।
Word deletion	অমর ভালোবাসে। amara bhālōvāsē।
Word insertion	অমর গীতাকে খুব ভালোবাসে। amara gītākē khuva bhālōvāsē।

Table A1: Word order shuffling, substitution, deletion and insertion may not necessarily generate wrong sentences in Bangla.

A Word Order

Table A1 shows that all possible word order of sentence অমর গীতাকে ভালোবাসে। (amara gītākē bhālōvāsē।) is correct.

B Related Work

In this section we discuss about GEC in English and other non-Indian languages.

English: CoNLL-shared task 2013 (Ng et al., 2013) and CoNLL-shared task 2014 (Ng et al., 2014) played a pivotal role in advancing GEC works in English. Other than providing 55,000+ grammatically incorrect sentences in English, they also categorized grammatical errors in English into 5 broad classes and 27 finer classes. Napoles et al. (2017) presented a parallel corpus of 1,511 sentences for English representing an extended range of language proficiency and uses holistic edits that make the original text more native sounding. Yannakoudakis et al. (2011) curated a corpus of 1,238 scripts from 1,238 distinct learners. The BEA-2019 shared task (Bryant et al., 2019) contributed a new benchmark dataset of 43,169 sentences curated from the Write&Improve+LOCNESS corpus that represents a broader range of native learners of English.

Other Languages: Unlike English, low-resource Asian languages suffer from the unavailability of large corpora for neural models. Attempts

have been made to enrich resources for GEC in many languages: Spanish (Davidson et al., 2020), German (Boyd, 2018), Russian (Rozovskaya and Roth, 2019), Czech (Náplava and Straka, 2019), Greek (Korre and Pavlopoulos, 2022), and Chinese (Rao et al., 2018). Syvokon et al. (2023) presented a corpus annotated for GEC and fluency edits for Ukrainian. Lee et al. (2021) gave four different noising methods, such as grapheme-to-phoneme noising rules and heuristic-based noising rules and others, to generate incorrect sentences for Korean. Lichtarge et al. (2019) proposed a rule-based system for deliberately injecting noises for low-resource languages like Indonesian (Irmawati et al., 2017). Solyman et al. (2022) proposed semi-supervised noising methods to generate 13,333,929 synthetic parallel examples from a monolingual corpus for Arabic.

C Grammar Error Categories

This section augments Sec. 3 by explaining the different error categories through examples. The grammatical errors for Bangla can be classified into 5 broader classes. These classes as well as the 12 finer distinctions within them are described next. A sentence may contain multiple errors of one class or different classes as well. Table 1 lists example sentences of the error classes.

C.1 Spelling Errors

Spelling errors are amongst the most frequent types of errors. In Bangla and major Indian languages, there are almost similar sounding consonants and, thus, mistakes between ন / ণ (n / ṇ), শ / ষ / স (ś / ṣ / s), র / ড় / ঢ় (r / ṛ / ṛh), etc. are prominent among even the native speakers. Spelling errors are further classified into 2 types.

1. **Non-Dictionary Words**: Spelling errors of this type result in words that are not in a dictionary. We have considered Vācaspati (Bhattacharyya et al., 2023) as the vocabulary of Bangla words since it covers literary works of almost 8 centuries and works from both India and Bangladesh. In the example shown in Table 1, কাজ (kāja) gets changed to কাব (kāva) which is not a word.
2. **Dictionary Words**: A spelling error of this type produces another word which is in the dictionary. However, in the context of the sentence, it is an error. For example, in Table 1, changing ড় (ṛ) of বাড়ি (vāṛi) to র (r) produces a perfect word বারি (vāri). The sentence, however, ceases to have any valid meaning. Mostly these errors are of *homonym* types, i.e., similar sounding words. Simple non-homonym typos may, however, also result in a dictionary word শাড়ি (śāṛi) that does not make sense in the sentence, as shown in the second example.

C.2 Word Errors

A prominent class of grammatical errors in almost any language including Bangla is word errors. We have categorized word errors further into different sub-classes as explained next.

1. **Tense Error**: In Bangla, like most other languages, there are specific verb forms for the three tenses. Hence, not using the correct form results in an error, as shown in the example in Table 1. However, tense errors are more common when multiple verbs are used in a sentence, and a mismatch among the tenses occur. In the second example in the table, while the first verb আসবে (āsavē) is in future tense, the second verb ফুটেছিল (phuṭēchila) is in past tense.
2. **Person Error**: Similar to tenses, there are different verb forms and pronouns for different persons in Bangla. It is, thus, an error to use the wrong person of a verb. The sentence in Table 1 shows an example where instead of the first person form করি (kari), the third person form করে (karē) is used with the pronoun আমি (āmi, I). These errors are common in Indian languages.
3. **Number Error**: In Bangla, the verb forms for both singular and plural numbers are the same. However, there are distinct forms for pronouns. The example in Table 1 shows such a wrong usage where the singular form আমি (āmi) is used instead of the plural form আমরা (āmarā). Number errors are more common in other Indian languages compared to Bangla.
4. **Gender Error**: In Bangla, the verb forms and pronouns for different genders are the same. However, there are distinct forms for adjectives as well as nouns. Moreover, the gender and number of an adjective should match that of the noun it qualifies. Hence, in the example in Table 1, since the proper noun উত্তম (uttama) is masculine, the correct adjective used should be the masculine form অভিনেতা (abhinētā) and not the feminine form অভিনেত্রী (abhinētrī). While strictly speaking, masculine forms of adjectives should not be used for feminine nouns, it is a common practice to accept them. In

such sentences, the masculine form takes the role of a gender-neutral form. Hence, the sentence সুচিত্রা একজন অসাধারণ অভিনেতা। (sucitrā ēkajana asādhāraṇa abhinētā।, Suchitra is an outstanding actor.) where সুচিত্রা (sucitrā) is a feminine proper noun but the adjective অভিনেতা (abhinētā) is masculine is not considered as incorrect. Many Indian languages such as Hindi has different forms of verbs for different genders and, thus, this kind of error is more common in those languages as compared to Bangla.

5. **Case Error**: Bangla and other Indian languages use a lot of inflected words. For different cases, different word forms are used that modify the original word. Case endings loosely correspond to prepositions in English. In the example in Table 1, the wrong case accusative is used instead of the correct case locative.
6. **Parts-of-Speech Error**: Sometimes, a word is used in a wrong parts-of-speech (POS). Since Indian languages including Bangla use a lot of nouns and their corresponding adjectives, these errors are common. Instead of a noun form, the adjective form is sometimes erroneously used, as shown in the example in Table 1.
7. **Missing Word Error**: These sentences are incomplete because of a missing word. Missing a verb in Bangla will always generate this kind of error, as shown in the example in Table 1, while missing a random word may or may not be grammatically wrong. Missing a noun corresponding to its adjective will also generate an erroneous sentence. The second example in Table 1 shows such a sentence.

C.3 Mixing of Language Variants: Gurucaṇḍālī Dōṣa

Bangla has a unique temporal language feature. All written works in Bangla till the 19th century were exclusively in সাধু ভাষা (sādhu bhāṣā, "refined language"). Authors started switching to (calita bhāṣā, "colloquial language") during the the 20th century and, currently, almost all the works are in this variant of the language. The two differ mostly in verb forms and pronouns, and use exclusive sets of these. This is similar to the old English usage of "thou shalt" versus the modern "you shall", etc., but is more elaborate. A sentence should be written in either of the variants. It is, thus, an error to mix, for example, pronouns of one variant with verbs of another variant. The example in Table 1 shows two cases. The sentence নন্দবাবু ইহা লক্ষ্য করেছেন। (nandavāvu ihā lakṣya karēchēna।) mixes the sādhu bhāṣā pronoun form ইহা (ihā) with the calita bhāṣā verb form করিয়াছেন (kariẏāchēna). Either the verb form or the pronoun can be corrected, as shown in the examples. This mixing error is known as "গুরুচণ্ডালী দোষ" (Gurucaṇḍālī Dōṣa) in Bangla.

C.4 Punctuation Errors

Punctuation errors occur due to usage of wrong punctuation marks, or absence of punctuation marks where needed, or spurious usage of punctuation marks. Thus, while a simple imperative sentence ends with a । (fullstop mark), putting ? (interrogative mark) results in an error, as shown in Table 1.

C.5 Semantic Errors

Semantic error is a special class of error where the sentence's semantic meaning becomes inconsistent or fictitious in the real world. For example, consider the sentence মানস আকাশ খেতে ভালোবাসে। (mānasa ākāśa khētē bhālōvāsē।) which literally means "Manas loves to eat the sky." Although this sentence is grammatically correct as far as usage of words, spellings, etc. are concerned, it is still considered a wrong sentence due to its semantics. Note that this is for ordinary usage in a language, and such sentences may be correct in science fiction or other fantasy novels. Table 1 shows two correct sentences corresponding to the above wrong one. While in the first example, the verb is modified, in the second, the noun is modified to produce a semantically meaningful sentence.

C.6 Multiple Errors

These sentences suffer from multiple errors of the same category or a combination of different categories of errors. For example, the sentence আমরা বান্ধুরা গতকাল কাশী যাবে। āmarā vāndhurā gatakāla kāśī yāvē। (We non-word will go to Kashi tomorrow.) consists of spelling error (non-dictionary), person error and tense error. The correct sentence can be আমরা বন্ধুরা আগামীকাল কাশী যাব। āmarā vandhurā āgāmīkāla kāśī yāva।

D Data Cleaning

- *Cleaning of Unicode characters*: Unicode characters "0020" (space), "00a0" (no-break space), "200c" (zero width non-joiner), "1680" (ogham

space mark), “180e” (mongolian vowel separator), “202f” (narrow no-break space), “205f” (medium mathematical space), “3000” (ideographic space), “2000” (en quad), “200a” (hair space) are separated from the texts.

- *Cleaning of different punctuation marks*: In Bangla, usage of punctuation marks has also evolved alongside words. In particular, we have treated the following as punctuation marks: “...”, “।...”, “।।”, “!–”, “–”.

E Generation of different types of errors

- **Spelling Errors:** Spelling errors are those for which the original intention was to write the correct word, but some characters are wrongly written. Typically, the misspelled word should be within one or at most two edit distance from the original word. They can be, thus, generated by substituting, inserting, or deleting one or two characters of a randomly chosen word in a sentence. These generated spelling errors may be of non-dictionary or dictionary types. We further collected 300 homonym word pairs from (Chakroborty, 2018). These homonyms are very common in Bangla. We replaced the original word in sentences with its corresponding homonyms to generate dictionary-based spelling errors.
- **Word Errors:** We have followed different procedures to generate different types of word errors in Bangla.
  1. **Tense Error:** We collected 24 most commonly used verbs and their forms across three tenses and three persons, resulting in 470 verb forms from (Chakroborty, 2018). These verb forms are replaced against the original word to generate erroneous sentences. These errors are difficult to generate since if the sentence contains only one verb and it is in its present form, then replacing it with past or future tense will not generate an error. For example in the sentence আমি বাড়ি যাব। (āmi vāṛi yāva।) (I will go home) if we replace "যাব" (yāva) with "গিয়েছিলাম" (giẏēchilāma) the resulting sentence আমি বাড়ি গিয়েছিলাম। (āmi vāṛi giẏēchilāma।) (I went to home) is not grammatically incorrect. So, to generate Tense errors, we need at least two verbs in the sentence, and we need to change only one, which we did randomly to generate Vaiyākaraṇa. For example, রাম যখন পড়তে বসবে, তখন লিখবে। rāma yakhana paṛatē vasavē, takhana likhavē। (Ram will write when he sits down to read.) if we change one amongst the verbs "বসবে" (vasavē) or "লিখবে" (likhavē) the resulting sentences রাম যখন পড়তে বসবে, তখন লিখেছিল। rāma yakhana paṛatē vasavē, takhana likhēchila। (Ram wrote when he sits down to read.) and রাম যখন পড়তে বসেছিল, তখন লিখবে। rāma yakhana paṛatē vasēchila, takhana likhavē। (Ram will write when he sat down to read) are gramatically wrong. If we change both the verbs it will again result in a grammatically correct sentence রাম যখন পড়তে বসেছিল, তখন লিখেছিল। (rāma yakhana paṛatē vasēchila, takhana likhēchila।) (When Ram sat down to read, he wrote.) These issues compelled us to adopt the noise injection methodology to generate erroneous sentences, which would not have been possible if we adopted other methodologies discussed in Sec: 2.There is one more type of tense error which occurs with respect to time like for sentence গতকাল আমি বাড়ি গিয়েছিলাম। (gatakāla āmi vāṛi giẏēchilāma) (I went to home yesterday) if changed to গতকাল আমি বাড়ি যাব। gatakāla āmi vāṛi yāva। (I will go to home yesterday) or আগামীকাল আমি বাড়ি গিয়েছিলাম। āgāmīkāla āmi vāṛi giẏēchilāma। (I went to home tomorrow) will generate grammatically incorrect sentences. We have crafted 15 such sentences manually and added them to Vaiyākaraṇa.
  2. **Person Error:** To generate these types of errors, we replaced the original verb form with its corresponding verb form from the other two types of persons.
  3. **Number Error:** To generate this kind of error, we collected 23 pronouns with both of their singular-plural forms from (Chakroborty, 2018). We injected this error by deliberately replacing the original singular (respectively, plural) pronoun with its corresponding plural (respectively, singular) form. For pronoun detection, we used the POS tagger by Sarker (2021) since pronouns are typically a frozen list and taggers do well on detecting them.
  4. **Gender Error:** We handcrafted 100 sentences for this kind of error. In each sentence, we chose a random word and changed

its case. We employed three native speakers to validate the error category, and based on majority voting, we added the sentences in Vaiyākaraṇa

5. **Case Error:** We collected a list of 20 cases and inflections and randomly interchanged them in the sentences to generate wrong sentences.
6. **POS Error:** We collected 350 noun-adjective word pairs from (Chakroborty, 2018). We replaced a noun (respectively, adjective) with its corresponding adjective (respectively, noun) to generate errors.
7. **Missing Word Error**: We ran the POS tagger (Sarker, 2021) and deleted verbs from the sentence to generate erroneous sentences. We applied the same technique to delete the noun corresponding to its adjective to generate errors. For other cases, we randomly deleted some words from the sentences. We asked three native speakers to validate whether the generated sentence was an error, and based on majority voting, we marked the sentences. If it is an error, we add the sentence to Vaiyākaraṇa. Else, we discard it.

- **Semantic Error:** We handcrafted 100 sentences for this kind of error. We employed three native speakers to validate the error category of the sentences, and based on majority voting, we added the sentences in Vaiyākaraṇa.
- **Gurucaṇḍālī Dōṣa:** We collected 140+ verbs and pronouns with their corresponding sādhu and calita forms from (Chakroborty, 2018). We then replaced the original word with its counterpart to generate this kind of error. In order to generate these sentences, we make sure that at least one verb or pronoun retains its original form so that the resulting sentence is actually an error that mixes the two variants. For example আমি খেতে খেতে হাঁটছি। (āmi khētē khētē hāṁṭachi।) we randomly changed "খেতে" (khētē) to "খাইতে" (khāitē) generating a wrong sentence আমি খাইতে খেতে হাঁটছি। (āmi khāitē khētē hāṁṭachi।). Changing all three "খেতে" (khētē), "খেতে" (khētē) and "হাঁটছি" (hāṁṭachi) will lead to a grammatically correct sentence আমি খাইতে খাইতে হাঁটিতেছিলাম। (āmi khāitē khāitē hāṁṭitēchilāma।).

By following these steps, we generated 226,507 grammatically incorrect sentences, as outlined in Table 3. Following the procedure outlined above, we can generate any number of grammatically incorrect sentences for Bangla.

F Manual Generation

Table A2 provides the details of the essays given for the manual annotation survey. All the 9 essays given in this survey are very commonly asked in 10th standard board exams. Each participant were asked to write an essay on a randomly picked topic. 36 participants undertook the survey 3 of whom are non-native speakers.

G Multiple Error Categories

We have observed from the manual survey that there are single errors in a sentence 77% of the time. The remaining 23% of the sentences contain multiple errors. Amongst them, the most common is multiple spelling errors in a sentence. We have categorised multiple errors of the same type in that category of error only, not inducted them in the multiple error category. Vaiyākaraṇa has 832,521 such sentences. Very few sentences have more than one type of error. Here, we show the distribution of sentences having different types of errors in a sentence. We have only considered that category of errors prevalent in the human survey done in Sec 4.1. Table A3 show the distribution of multiple errors in Vaiyākaraṇa.

H Results of Transformer based models

Table A4 shows the performance of all neural models and Random Forest classifier on the 56,621,770 sentences in Vaiyākaraṇa.

I LLM Evaluation

Large language models (LLMs), such as FLAN (Wei et al., 2022), OPT (Zhang et al., 2022), and PaLM (Chowdhery et al., 2022), have demonstrated remarkable performance in natural language understanding (NLU) tasks for English. GPT-3.5 and GPT-4 (OpenAI et al., 2023) with ChatGPT interface based on GPT (Brown et al., 2020) have garnered significant interest due to their outstanding performance in unifying all NLU tasks into generative tasks. Niyogi and Bhattacharya (2024) proposed a monolingual LLM in Bangla. We evaluated the performance of these LLMs in detecting and correcting erroneous sentences in Bangla. We have used the pre-trained version of these models with and without instruction fine-tuning.

We have prompted each of the models with two prompts "বাক্যটি সঠিক অথবা ভুল কিনা তা নির্ধারণ কর।" (vākyaṭi saṭhika athavā bhula kinā tā nirdhāraṇa kara।, is this sentence grammatically correct?) and "'সঠিক ব্যাকরণ সংশোধন কর।" (saṭhika vyākaraṇa saṁśōdhana kara।, correct this sentence) and evaluated their responses against the ground truth on the human evaluated 600-sentence set. The first prompt focuses on whether a grammatically incorrect sentence can be identified. Table 5 shows that the highest accuracy is only 32.00% for the 600 sentences that are manually evaluated. The second prompt that aims to correct a sentence was passed for the 520 wrong sentences present in the set of 600. GPT-3.5 achieved the highest but still corrected only 35.00% of the sentences. An in-depth analysis of the responses shows that the LLMs perform decently in identifying tense, person and gender errors (45.00%). However, their performance drops to below 20% for dictionary-based spelling errors (homonyms), POS errors and Gurucandali Dosa. Even after instruction-tuning the LLMs with Vaiyākaraṇa, GPT-4 manages to correct 47.50% of 520 sentences. This result highlights the limitations of LLMs for GEC in Bangla.

J LLM Prompts

In this section we show the detailed prompts and their responses. The responses shown here are generated by GPT3.5 using ChatGPT without instruction tuning.

J.1 Prompts for Detecting Erroneous Sentences

Fig 3 and Fig 4 are responses generated by GPT-3.5 over ChatGPT without instruction tuning for prompt বাক্যটি সঠিক না তা নির্ধারণ কর। (vākyaṭi saṭhika nā tā nirdhāraṇa kara।). Fig 3 shows that GPT-3.5 identified গান শুনেই অন্যরস ছেড়ে ঠাকুর এখন অন্যরসে মজেছেন। (gāna śunēi anyarasa chēṛē ṭhākura ēkhana anyarasē majēchēna।) as correct even though there is *spelling error (dictionary)* in the sentence and the target sentence is গান শুনেই অন্নরস ছেড়ে ঠাকুর এখন অন্যরসে মজেছেন। (gāna śunēi annarasa chēṛē ṭhākura ēkhana anyarasē majēchēna।). Fig 4 also shows that GPT-3.5 denoted ইন্দ্রজলিকের ম্যাজিকে সবাই মুগ্ধ। as correct even though the sentence suffers from *POS error* and the target sentence is ঐন্দ্রজালিকের ম্যাজিকে সবাই মুগ্ধ। (aindrajālikēra myājikē savāi mugdha।)

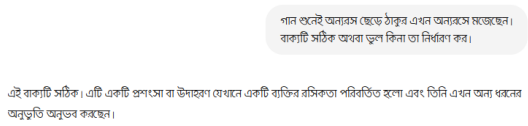

Figure 3: Grammar error detection prompt 1

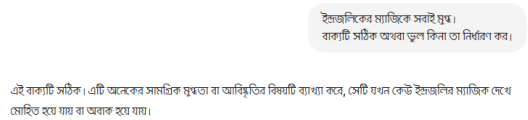

Figure 4: Grammar error detection prompt 2

J.2 Prompts for Correcting Erroneous Sentences

Fig 5 and Fig 6 show the output of GPT3.5 when prompted with এই বাক্যটিকে সঠিক করো। (ēi vākyaṭikē saṭhika karō।). Fig 5 shows that the LLM generates another sentence ইন্দ্রজলিকের ম্যাজিক সবার মনে আশ্চর্য এবং উৎসাহিত করে। with same *POS Error* present in the input sentence instead of the target sentence ঐন্দ্রজালিকের ম্যাজিকে সবাই মুগ্ধ। (aindrajālikēra myājikē savāi mugdha।). Fig 6 captures the inability of GPT-3.5 to identify Homonym errors (*Spelling Error Dictionary class*) by modifying the sentence to কোনো ধুলো জমেছে না। (kōnō dhulō jamēchē nā।) in place of the target sentence কোণে ধূলো জমেছে। (kōṇē dhūlō jamēchē।).

K Efficacy

We created a test set of 52,100 sentences to evaluate the efficacy of Vaiyākaraṇa. The distribution of error categories on the test set is shown in Table A5.

L Nirdeśa

We created a dataset, Nirdeśa, of 566,238 sentences from Vaiyākaraṇa to instruction-tune the LLMs and converted it to Alpaca-JSON format.

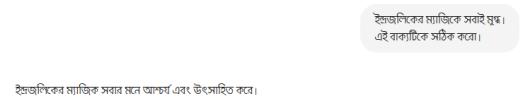

Figure 5: Grammar error correction prompt 1

Figure 6: Grammar error correction prompt 2

The distribution of error categories in Nirdeśa is shown in Table A6.

M Evaluation Metrics

In this section, we describe the evaluation metric used in Sec: 6 for the paraphrasing task.

BERT score: It measures the semantic similarity between two pieces of text.

$BERTScore = \frac{1}{N} \times \Sigma_i^N F_1 \text{score}(BERT_{similarity}, BERT_{Precision}, BERT_{recall})$

SacreBLEU: It measures the similarity between predicted and target sentences (mostly human-annotated). It focuses not only on the accuracy but also on the overall fluency and coherence of the predicted sentence. The mathematical equation for calculating SacreBLEU score is as follows:

$\text{SacreBLEU} = 100 \times e^{(1-BP)} \times \frac{sum_ofNgram_matches}{total_Ngrams}$

$\text{BP} = \min(1, e^{(1-\frac{reference_length}{candidate_length})})$ where $reference_length$ is the length of the targeted sentence and $candidate_length$ is the length of the predicted sentence by the model. BP or Brevity Penalty is a penalty term to adjusts the score for the length of the generated sentences by the model corresponding to the targeted sentence.

$F_{0.5}$ **score**: It is a standard metric used for GEC in other languages with more emphasis on the precision of the prediction than recall.

$F_{0.5} = \frac{1.25 \times Precision \times Recall}{.25 \times Precision + Recall}$

N Finer and Broad Categories

Fig 7 shows the performance (macro-F1) of BanglaT5 on broad and finer category of errors with different numbers of erroneous sentences in the test set.

O Model Hyper Parameters

We fine-tuned the transformer based models with Adam optimizer (Kingma and Ba, 2015) and learning rate of 2e-5 for 20 epochs. The batch size of each transformer-based models have been 16 with maximum length set at 512. Same parameter values have been used for instruction-tuning

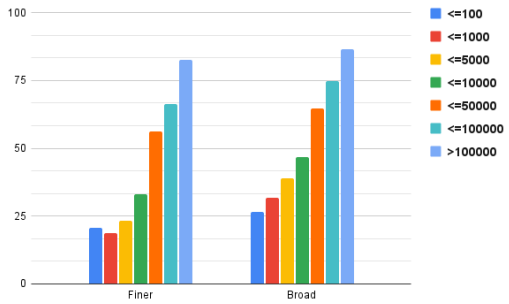

Figure 7: Figure showing performance of BanglaT5 on broad and finer error categories in Bangla.

LLMs. Random Forest classifier is used with number of trees, `n_estimators`, set to 100 and maximum depth of each tree is set to 6. We have set `min_samples_leaf` to 11 and `n_jobs` to 4 for Random Forest with "gini" as the measure. All other parameters are set to their default values. The total number of trainable parameters for Random Forest is 441,875.

Essay topic	# Essays
বিজ্ঞান আশীর্বাদ না অভিশাপ vijñāna āśīrvāda nā abhiśāpa Is science boon or bane?	8
একটি বৃষ্টির দিন ēkaṭi vr̥ṣṭira dina A rainy day	8
একটি নদীর আত্মকথা ēkaṭi nadīra ātmakathā Autobiography of a river	6
একটি স্মরণীয় দিন ēkaṭi smaraṇīẏa dina A memorable day	7
খেলা শুধু খেলা নয় khēlā śudhu khēlā naẏa Sports is not just sports	6
হঠাৎ আলাদিনের আশ্চর্য প্রদীপ কুড়িয়ে পেলে কী করবে haṭhāt ālādinēra āścarya pradīpa kuṛiẏē pēlē kī karavē What will you do if you suddenly get Aladdin's lamp?	5
সামাজিক মাধ্যম আশীর্বাদ না অভিশাপ sāmājika mādhyama āśīrvāda nā abhiśāpa Is social media boon or bane?	12
ভীন গ্রহের প্রাণী ও তোমার কথোপকথন bhīna grahēra prāṇī ō tōmāra kathōpakathana Dialogue between an extraterrestrial being and you	7
পনেরো বছর আগের তুমি আর আজকের তুমির মধ্যে কথোপকথন panērō vachara āgēra tumi āra ājakēra tumira madhyē kathōpakathana Dialogue between 15-years older you and present you	7
একটি বটগাছের আত্মকথা	10
ēkaṭi vaṭagāchēra ātmakathā Autobiography of a Banyan Tree	10
Total	**76**

Table A2: Grammar essays for manual survey

Error Class	#Occurences
Non_Dictionary-Dictionary	185,729
Case-Non_Dictionary-Dictionary	50,000
Gurucaṇḍālī Dōṣa-Non_Dictionary-Dictionary	25,000
Non_Dictionary-Dictionary-Punctuation	25,000
Punctutation-Punctuation	25,000
Case-Non_Dictionary-Dictionary-Gurucaṇḍālī Dōṣa	25,000
Non_Dictionary-Dictionary-Gurucaṇḍālī Dōṣa-Punctuation	25,000
Non_Dictionary-Dictionary-Gurucaṇḍālī Dōṣa-Punctuation-Case-Missing Word	25,000
Total	385,729

Table A3: Distribution of Multiple Errors in Vaiyākaraṇa

Model	Parameters	Binary	Broad	Finer
		Mean	Mean	Mean
IndicBERTv2.0	117M	84.65±0.14	81.80±0.45	80.10±0.53
MuRIL	236M	78.93±0.22	76.96±0.45	73.48±0.66
BanglaBERT	110M	84.86±0.18	81.94±0.32	80.54±0.12
VAC-BERT	17M	82.65±0.25	81.44±0.44	79.82±0.32
sahajBERT	17M	79.30±0.20	76.24±0.18	74.60±0.20
RandomForest	0.42M	45.20±0.55	36.40±0.74	30.80±1.55
BanglaT5	290M	87.60±0.18	83.06±0.88	81.85±0.25

Table A4: Macro-F1 of transformer models on Vaiyākaraṇa.

Error Class	#Occurences
Non-Dictionary	8,000
Dictionary	2,000
Spelling Errors	10,000
Tense Errors	1,000
Person Errors	1,000
Number Errors	40
Gender Errors	30
Case Errors	2,000
POS Errors	750
Missing Words	750
Word Errors	5,570
Punctuation Errors	2,000
Semantic Errors	30
Gurucaṇḍālī Dōṣa	2,000
Multiple Errors Different Categories	500
InCorrect	20,100
Correct	32,000
Total	52,100

Table A5: Distribution of different error categories in the test set of 52,100 sentences.

Error Class	#Occurences	Total_Percentage	Error_Cat Percentage
Non-Dictionary	113,244	20.00	50.00
Dictionary	26,046	4.60	11.50
Spelling Errors	139,290	24.60	61.49
Tense Errors	4,983	0.88	2.20
Person Errors	4530	0.80	2.00
Number Errors	20	0.004	0.009
Gender Errors	10	0.002	0.004
Case Errors	26,046	4.60	11.50
POS Errors	4622	0.82	2.04
Missing Words	10,690	1.89	4.72
Word Errors	50,901	8.99	22.47
Punctuation Errors	26,046	4.60	11.50
Semantic Errors	10	0.002	0.004
Gurucaṇḍālī Dōṣa	6,402	1.13	2.83
Multiple Errors Different Categories	3,858	0.68	1.70
InCorrect	226,507	40.00	100.00
Correct	339,731	60.00	0.00
Total	566,238	100.00	100.00

Table A6: Distribution of different error categories in the test set of 20,000 sentences.