Advancing Bangla Punctuation Restoration by a Monolingual Transformer-Based Method and a Large-Scale Corpus

Mehedi Hasan Bijoy[1,†], Mir Fatema Afroz Faria[2,†], Mahbub E Sobhani[3]
Tanzid Ferdoush[3] and Swakkhar Shatabda[3,*]
[1]Aalto University, [2]North South University, [3]United International University
{mehedi.bijoy@aalto.fi, afroz.fariaa@gmail.com, msobhani171134@bscse.uiu.ac.bd,
ferdoushtanzid@gmail.com, and swakkhar@cse.uiu.ac.bd}
† denotes equal contributions
∗ denotes corresponding author

Abstract

Punctuation restoration is the endeavor of reinstating and rectifying missing or improper punctuation marks within a text, thereby eradicating ambiguity in written discourse. The Bangla punctuation restoration task has received little attention and exploration, despite the rising popularity of textual communication in the language. The primary hindrances in the advancement of the task revolve around the utilization of transformer-based methods and an openly accessible extensive corpus, challenges that we discovered remained unresolved in earlier efforts. In this study, we propose a baseline by introducing a monolingual transformer-based method named Jatikarok[1] , where the effectiveness of transfer learning has been meticulously scrutinized, and a large-scale corpus containing 1.48M source-target pairs to resolve the previous issues. The Jatikarok attains accuracy rates of 95.2%, 85.13%, and 91.36% on the BanglaPRCorpus, Prothom-Alo Balanced, and BanglaOPUS corpora, thereby establishing itself as the state-of-the-art method through its superior performance compared to BanglaT5 and T5-Small. Jatikarok and BanglaPRCorpus are publicly available at https://github.com/mehedihasanbijoy/Jatikarok-and-BanglaPRCorpus.

1 Introduction

The continuous effort to bridge the linguistic gap between human natural language and digital devices has propelled natural language processing (NLP) to its current level of advancement. Despite these advances in NLP, Bangla language processing continues to present significant challenges including multimodal complexities stemming from intricate language rules. Proper punctuation placement, particularly in the Bangla language, plays a pivotal role in further reducing this barrier and facilitating downstream Bangla natural language processing (BNLP) tasks.

Previous studies have highlighted the dominance of transformer-based models such as BERT (Fu et al., 2021), RoBERTa (Nagy et al., 2021), and ALBERT (Shi et al., 2021) and have showcased their efficacy in leveraging contextual information for punctuation restoration in high-resource languages. Additionally, architectural enhancements such as attention mechanisms has also shown good performance (Yi and Tao, 2019). Following the trend of domain-specific fine-tuning of pretrained models, alongside post-processing techniques, has also achieved close to adequate performance (Chordia, 2021). Cross-lingual augmentation strategies enhance transformer models for languages with diverse resources, which is nonexistent in (Alam et al., 2020). The study conducted by (Rahman et al., 2023) only restored four types of punctuation marks in the Bangla language. However, there are at least nine more punctuation marks that need to be addressed to exhaustively capture the meaning. Moreover, it should be pointed out that deep learning models may not be capable of covering a significant proportion of punctuation in cases where the corpus is comparably small (Monsur et al., 2022). However, transformer-centric approaches have started demonstrating impressive performance in various BNLP tasks, including grammar and spelling error correction (Bijoy et al., 2022). Surprisingly, transformer-based methods have yet to be applied in any studies for the Bangla punctuation restoration task. Consequently, in this study, we leverage the impressive capabilities of transformers and initiate an investigation into their unexplored potential for the task.

In this study, we propose a transformer-based method named Jatikarok for the task with a uniquely tailored architecture of six encoder and decoder layers, optimizing the balance between

[1] যতিকারক

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 18–25
December 7, 2023 ©2023 Association for Computational Linguistics

model complexity and computational efficiency for Bangla punctuation restoration, while enhancing its performance through transfer learning, which consequently renders it a monolingual method. Furthermore, we introduce BanglaPRCorpus, a large-scale parallel corpus for the task consisting of 1.48 million source-target pairs. The contributions of this paper are summarized below:

- A monolingual transformer-based method called Jatikarok has paved the way for the first-ever monolingual transformer-based baseline in the Bangla punctuation restoration task.
- We benchmarked our proposed method on various corpora, and it has emerged as the state-of-the-art approach on two additional corpora, namely Prothom-Alo Balanced and BanglaOPUS, in addition to ours.
- The effectiveness of transfer learning from the Bangla grammatical error correction task has been scrutinized for its ability to capture intricate linguistic patterns within this specific task.
- A large-scale parallel corpus comprising 1.48M source-target pairs has been developed by incorporating 16 Bangla punctuation marks and made publicly available, making Bangla no longer a resource-scarce language for the task.

The subsequent sections of the paper are organized as follows: Section 2 presents an in-depth analysis of the background of Bangla punctuation restoration; the process of constructing our corpus is expounded upon in Section 3; Section 4 elucidates the architecture of our proposed method; Section 5 presents the tangible results derived from our empirical study; Section 6 culminates our investigation by offering concluding remarks and outlining potential avenues for future research.

2 Literature Review

The task of punctuation restoration has garnered widespread attention, leading to the emergence of novel insights within methods and datasets. We delve into an examination of the recent studies conducted for punctuation restoration. Our extensive studies identified several contemporary transformer-based and deep learning methods in the realm of Bangla punctuation restoration tasks and in various high and low-resource languages such as Transformer (Lai et al., 2023; Nguyen et al., 2019; Wu et al., 2022), RNN (Rahman et al., 2023; Kim, 2019) and Hybrid (Yi et al., 2020; Bakare et al., 2023).

Among RNN-based methods (Rahman et al., 2023) proposed a novel approach comprised of a bidirectional recurrent neural network (BRNN) model with an attention mechanism. The authors trained a large Bangla dataset focusing specifically on predicting the exclamation mark and achieved 96.8% accuracy with various post-processing techniques. Likewise, (Kim, 2019) took a similar approach to solve the task.

The advent of NLP has seen the employment of transformer-based methods where M-BERT, BERT, RoBERTa, BioBERT, and ELECTRA have been utilized (Sunkara et al., 2020; Huang et al., 2021). (Alam et al., 2020) explored transformer-based language models to restore punctuation and improved Bangla training and evaluation data whereas (Monsur et al., 2022) utilized inadequate supervision and proposed a unique method for acquiring dialogue data in languages with few resources and evaluated the dataset by finetuning BanglaBERT (Bhattacharjee et al., 2022). Predicting punctuation for sequences instead of individual tokens by utilizing RoBERTa-base, (Courtland et al., 2020) proposed an innovative approach to solving the task. (Guerreiro et al., 2021) followed a homogeneous approach and proposed a contextual embedding-based punctuation prediction model. RoBERTa outperformed other transformer-based models in the comparison.

Our study has also revealed that the implementation of hybrid models has yielded exceptional results in addressing complex natural language processing tasks. By taking advantage of the evaluation of different BERT transformer models using LSTM and GRU with a linear neural network layer (Bakare et al., 2023) proposed a robust punctuation restoration algorithm. Besides, (Makhija et al., 2019) proposed a LSTM-CRF(Conditional Random Field) model that uses pre-trained BERT embeddings to make tagging decisions that take step interdependence into account to solve the punctuation restoration problem.

A thorough analysis found that transformer-based methods outperform RNN-based ones.

While RNNs may struggle with feature coverage and handling large datasets, transformers do not face these challenges. However, a downside of transformer-based approaches is that they require huge datasets to perform effectively.

3 Corpus Creation

We consider a total of 16 distinct punctuation marks, including period(’ ।’), comma(’,’), exclamation mark (’!’), question mark (’?’), semicolon (’;’), Bangla colon ('ঃ'), colon (’:’), double quotation mark (”), single quotation mark (’), hyphen (’-’), opening parenthesis (’(’), closing parenthesis (’)’), opening curly brace (’{’), closing curly brace (’}’), opening square bracket (’[’), and closing square bracket(’]’), to curate the corpus. The details of these punctuation marks are delineated as follows:

Period(।**)**: A definitive halt, denoting the terminus of a sentence in Bangla.

Comma(,): An eloquent separator, orchestrating rhythm within lists, crafting succinct pauses, and clarifying sentence structure.

Exclamation Mark(!): A linguistic exclamation point, amplifying emphasis, evoking astonishment or fervor, typically crowning the culmination of sentences.

Question Mark(?): An inquisitive note, framing direct queries. Its presence, positioned at sentence conclusions, signifies an inquest for insight.

Semicolon(;): A poised pause, surpassing a comma’s subtlety yet shying from a full stop’s grandeur. It adroitly links kindred concepts.

Bangla Colon(ঃ**)**: A signal which indicates that what comes next is elaborating on, explaining, or providing examples related to the preceding clause or phrase.

Colon(:): An introducer of elucidation, explanations, and verbatim passages within sentences, colonizes text with structured context.

Double Quotation Mark(”): A textual embrace for direct discourse or citations in Bangla script, encapsulating borrowed expressions.

Single Quotation Mark(’): An enigmatic gesture, encircling quotes within quotes or indicating nuanced semantics, an annotation of depth.

Hyphen(-). A linguistic bridge, tethering word parts, fusing compound lexemes, and demarcating ranges with subtle precision.

Opening Parenthesis(( ): A grammatical cradle, ensconcing auxiliary or clarifying content, nurturing intricate sentence ecosystems.

Closing Parenthesis( )): A tender closure, rounding out preceding parenthetical, nurturing textual harmony and enclosure.

Opening Curly Brace({): A technical flourish, sometimes corralling supplementary information or code within contexts of expertise.

Closing Curly Brace(}): A counterpart to the opening brace, it brings closure, marking the ambit of enclosed insights or code.

Opening Square Bracket([): A gateway to lists, references, and augmented text in Bangla, welcoming expanded textual horizons.

Closing Square Bracket(]): The ultimate gatekeeper, sealing the opening bracket’s portal, concluding augmented textual exploration.

3.1 Data Sourcing

We source our data from a publicly available Bangla paraphrase corpus (Akil et al., 2022). This dataset comprises approximately 466,000 carefully produced pairs of artificially created rephrased sentences in the Bangla language. These rephrased sentences have been meticulously crafted to uphold both the meaning’s coherence and the diversity of sentence structure, guaranteeing their outstanding quality.

3.2 Data Preprocessing

We consider 72 distinct characters that frequently occur in Bangla text denoted as $DC = \{DC_1, DC_2, ..., DC_{72}\}$, in addition to 16 Bangla punctuation marks represented by $PM = \{PM_1, PM_2, ..., PM_{16}\}$ and a space SP, resulting in a set of 89 Bangla characters represented by $C = \{DC + PM + SP\} = \{C_1, C_2, ..., C_{89}\}$. Next, we take into account each of the sentences indicated as $S = \{S_1, S_2, ..., S_N\}$, where N represents the number of characters in the sentence. We iterate through each of the characters $S_i \in S$ and remove any character that is not present in the unique character set C.

3.3 Punctuation Removal Procedure

We randomly remove N punctuation marks from a sentence S, based on their availability, where $N >= 1$ & $N <= 10$. To achieve this, we follow these steps: (Step 1) Initially, we count the number of punctuation marks, P_{count}, present in the sentence. If P_{count} is less than the number of punctuation marks we intend to remove from the sentence, we simply skip the sentence. (Step 2) Otherwise, to remove a punctuation mark $PM_i \in PM$, we begin by shuffling the list of punctuations, PM. (Step 3) Proceeding to the next step, we iterate through the list of punctuations, PM, and determine whether the sentence contains the specific punctuation mark, PM_i. (Step 4) If the punctuation mark is present such that $PM_i \in S$, we remove it from the sentence and continue with the process. (Step 5) Finally, we repeat these steps from 1 to 4 for N times to achieve the removal of the desired N punctuation marks from a sentence.

3.4 Corpus Statistic

Our proposed Bangla punctuation restoration corpus (BanglaPRcorpus) consists of 1.48 million source-target pairs. In these pairs, the source sentences lack punctuation, while the target sentences are the corrected versions where missing punctuation is restored. To do so, we systematically eliminated punctuation marks in varying quantities, ranging from 1 to 10, within each sentence. Moreover, the minimum, maximum, and average number of words in a sentence of our corpus is 2, 127, and 12.9, respectively.

4 Methodology

4.1 Problem Formulation & Overview

Consider two sequences of tokens, $X_I = \{x_1, x_2, ..., x_n\}$, and $Y_I = \{y_1, y_2, ..., y_k\}$, where X_I represents an erroneous input sequence with missing punctuation marks, and Y_I represents the corresponding corrected sequence with punctuation marks restored. The encoder ($E(\cdot)$) of our method takes an erroneous input sentence X_I, which is first tokenized using a pre-trained tokenizer ($T(\cdot)$), and generates a representative vector of the sentence, denoted as $V = [V_1, V_2, ..., V_{512}]$. Subsequently, the decoder ($D(\cdot)$) utilizes the representative vector V, along with the previously generated tokens, to autoregressively generate the corresponding correct sentence. The entire procedure can mathematically be abbreviated as follows:

$$\hat{Y} = D((E(T([X_I]), W^E), D_{out}^{t-1}), W^D) \quad (1)$$

4.2 Motivations

Bangla is the fifth (Bhattacharyya et al., 2023) most spoken language, considering the number of speakers. Beyond mere documentation, Bangla serves multifarious communicative purposes, highlighting its diverse utility. A method aimed at enhancing typing proficiency by rectifying misused punctuation could offer substantial advantages. The endeavor of punctuation restoration holds significant importance to its enhancement of text lucidity and interpretability, thus circumventing potential ambiguities. Consequently, it contributes to the amelioration of downstream NLP tasks.

4.3 Jatikarok

In this section, we provide the details on Jatikarok.

4.3.1 Encoder

Given an input sequence of tokens $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$, where n is the sequence length, we assigned unique discrete values to each word. We ensured uniform input dimension by expanding each input sequence $\mathbf{X}_i$ by incorporating padding. Subsequently, each token, x_i, undergoes an embedding layer $\mathbf{E}$ to convert discrete inputs into continuous vector representations using a trainable matrix in a latent space, such that $\mathbf{E}_i = \mathbf{Embed}(\mathbf{x_i})$. Notably, these matrices are fine-tuned via backpropagation during training to minimize the loss. These embeddings are subsequently combined with positional encodings $\mathbf{PE}$ to account for token order where $\mathbf{PE}_i$ represents the positional encoding for x_i. The combined embeddings, denoted as $\mathbf{Z}_i = \mathbf{E}_i + \mathbf{PE}_i$, are then fed into a stack of K identical layers, each composed of two main components: a multi-head self-attention mechanism and position-wise feedforward networks. The self-attention mechanism computes weighted representations for each token by attending to all tokens in the sequence $\mathbf{X}$ using learnable query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) vectors. This self-attention mechanism is defined as follows (Vaswani et al., 2017):

$$\mathbf{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{softmax}(\frac{\mathbf{QK}^T}{\sqrt{\mathbf{d_k}}})\mathbf{V} \quad (2)$$

The self-attention mechanism calculates weighted representations of each token by considering interactions with all other tokens in the sequence, enabling the capture of contextual dependencies. The position-wise feedforward networks introduce non-linearity through two linear transformations followed by a non-linear activation function, ReLU, enhancing the acquired representations. The outputs of each layer are sequentially

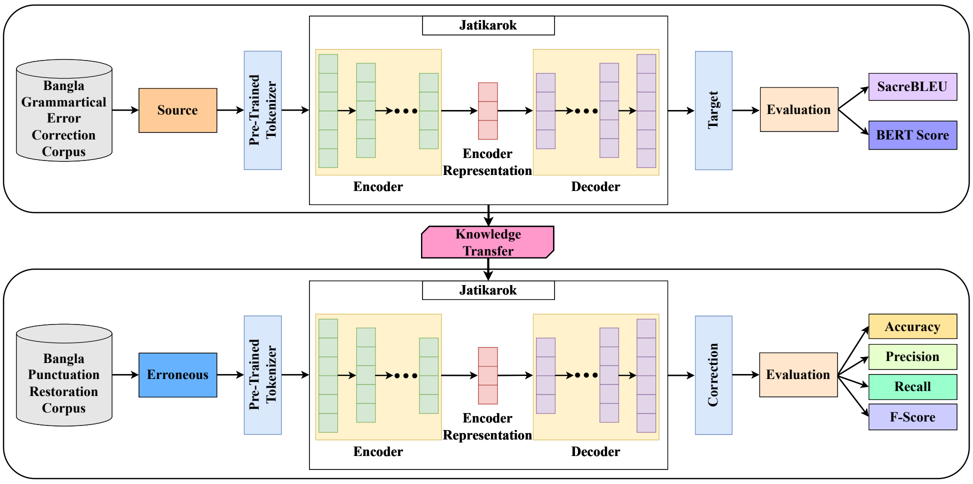

Figure 1: **(Top)** Jatikarok is initially trained on the Bangla Grammatical Error Correction (BGEC) task. **(Middle)** The insights acquired during the BGEC training are preserved for subsequent knowledge transfer to the Bangla Punctuation Restoration (BPR) task. **(Bottom)** Jatikarok is then fine-tuned on BPR corpora, leveraging the knowledge gleaned from the BGEC task.

propagated through the stack of K identical layers, yielding refined representations that encode both local and global dependencies, incorporating rich contextualized portrayals of the input sequence $\mathbf{X}$.

4.3.2 Decoder

Firstly, the target sequence, which is denoted as $\mathbf{Y} = \{y_1, y_2, \ldots, y_m\}$, where m is the sequence length, is embedded into a latent space using learned embeddings: $\mathbf{E}y_i = \mathbf{Embed}(y_i)$. To convey information about token order, positional encodings $\mathbf{PE}y_i$ are added to these embeddings ($\mathbf{E}y_i$). The resulting embeddings $\mathbf{Z}y_i = \mathbf{E}y_i + \mathbf{PE}y_i$ are then passed through a stack of L similar decoder layers, each composed of two primary components: a masked multi-head self-attention mechanism and position-wise feedforward networks. The computation of masked multi-head self-attention follows the same equation as regular multi-head self-attention (as given in Equation 2), with the crucial distinction that it enforces a restriction preventing the model from attending to tokens that occur in the future within the sequence. In contrast, the position-wise feedforward networks introduce non-linearity through linear transformations followed by a non-linear activation function (ReLU), enhancing the learned representations similar to the encoder's feedforward networks. The obtained representations from each layer are sequentially propagated through the stack of L similar decoder layers, resulting in refined target sequence representations denoted as $\mathbf{Y}$.

4.3.3 Hyperparameters

To maintain consistency, a hidden size dimension of 512 is employed across all layers within the encoder and decoder. Moreover, the feedforward neural network layers, which consist of 2048 neurons, contribute significantly to the model's depth and capacity. In order to mitigate the risks of overfitting, a dropout ratio of 0.1 is applied, thereby promoting robust and effective learning. The incorporation of the ReLU activation function introduces essential non-linearity to the network's computations. Throughout the training process, a learning rate of 5×10^{-5} is applied, and the model undergoes 100 epochs of training using the AdamW optimizer. This optimization process is carefully guided by the categorical cross-entropy loss function, which effectively steers the model towards achieving the desired translation outcomes.

5 Experimental Analysis

5.1 Datasets

- **BanglaPRCorpus (Ours).** It consists of 1,481,149 (1.48M) source-target pairs. We split the corpus into training and test sets, keeping 85% of the data in the training set and 15% in the test set, with each type of erroneous sentence, based on the number of punctuation removed, to maintain a balanced distribution. As a result, our training and test sets comprise 1,258,977 (1.26M) and 222,172 (222.1K) source-target instances, re-

Method	**#Params.**	BanglaPRCorpus					Prothom-Alo Balanced					BanglaOPUS				
		ACC	**PR**	**RE**	**F1**	**F0.5**	**ACC**	**PR**	**RE**	**F1**	**F0.5**	**ACC**	**PR**	**RE**	**F1**	**F0.5**
BiLSTM (Rahman et al., 2023)	**11.54M**	–	–	–	–	–	–	0.594	0.44	0.506	–	–	0.546	0.394	0.458	–
BanglaT5	247.53M	83.94%	0.839	0.839	0.839	0.841	76.53%	0.77	0.77	0.77	0.783	80.66%	0.806	0.806	0.806	0.813
T5-Small	60.51M	72.67%	0.728	0.727	0.727	0.728	74.95%	0.74	0.75	0.75	0.761	74.81%	0.748	0.748	0.748	0.754
Jatikarok	74.36M	**95.2%**	**0.953**	**0.952**	**0.952**	**0.955**	**85.13%**	**0.85**	**0.851**	**0.845**	**0.852**	**91.36%**	**0.914**	**0.914**	**0.914**	**0.92**

Table 1: The juxtaposition of the quantitative performance of different existing methods across various corpora.

spectively.

- **Prothom-Alo Balanced (Rahman et al., 2023).** It encompasses a total of 80150 source-target pairs after our meticulous preprocessing. The corpus was partitioned into training and test sets by maintaining an 85% and 15% split. Consequently, the resultant training set and test set comprise 68128 and 12022 source-target pairs, respectively.

- **Bangla OPUS (Tiedemann, 2012).** Following a comprehensive text preprocessing phase, we identified a total of 877,299 source-target pairs within the corpus. Subsequently, we divided the corpus in an 85:15 ratio to establish distinct training and test sets. This division resulted in 745,705 pairs within the training set, while the test set comprised 131,594 pairs.

5.2 Baselines

- **BanglaT5(Akil et al., 2022).** It is a pre-trained language model developed by fine-tuning the T5(Raffel et al., 2020) architecture specifically for the purpose of Bangla paraphrase task.

- **T5-Small(Raffel et al., 2020).** It is a variant of the Text-To-Text Transfer Transformer (T5) architecture featuring a smaller number of parameters (≈70M) compared to larger versions of T5 (220M).

5.3 Performance Evaluation

We evaluate the effectiveness of our model in restoring punctuations with accuracy, precision, recall, and the F-beta score. Mathematically, accuracy(Eq. 3), precision(Eq. 4), recall(Eq. 5), and the Fβ score(Eq. 6) can be defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

$$Precision(PR) = \frac{TP}{TP + FP} \tag{4}$$

$$Recall(RE) = \frac{TP + TN}{TP + FN} \tag{5}$$

$$f\beta\ score = (1 + \beta^2) \times \frac{PR \times RE}{\beta^2 \times PR + RE} \tag{6}$$

Where TP, TN, FP, and FN mean True Positive, True Negative, False Positive, and False Negative.

5.4 Main Results

5.4.1 Quantitative Results

The quantitative performance of different transformer-based methods on various corpora has been presented in Table 1. Our proposed model, Jatikarok, demonstrates significant performance superiority over both BanglaT5 and T5-Small across all three corpora, establishing itself as the new state-of-the-art method. It surpasses BanglaT5 and T5-Small in all evaluation measures, including accuracy, precision, recall, F1 score, and F0.5 score. Our method outperforms BanglaT5, which is the second-best model in comparison, despite having a parameter size three times smaller. It achieves 11.26%, 8.6%, and 10.7% higher accuracy scores on the BanglaPRCorpus, Prothom-Alo Balanced, and BanglaOPUS corpora, respectively. However, for multiple punctuation marks removed in a sentence, we did not consider them in the metrics individually, rather we calculated the overall accuracy, precision, recall, and F-scores considering the whole sentence.

5.4.2 Qualitative Results

The qualitative performance of BanglaT5, T5-Small, and Jatikarok has been juxtaposed in Table 2, effectively highlighting the superiority of our Jatikarok over BanglaT5 and T5-Small. The examples in the table explicitly illustrate that as the number of missing punctuation marks increases in a sentence, the performance of other methods decreases, while our Jatikarok maintains better accuracy. For instance, all methods performed well when only one punctuation mark was missing in a sentence. As the number increases to two, only our Jatikarok correctly corrects the sentence. However, when punctuation marks increase rapidly, all

(Input) স্বাভাবিকভাবেই এটা তিনি মেনে নিতে পারেন নি
(BanglaT5) স্বাভাবিকভাবেই এটা তিনি মেনে নিতে পারেন নি। (✓)
(T5-Small) স্বাভাবিকভাবেই এটা তিনি মেনে নিতে পারেন নি। (✓)
(Jatikarok) স্বাভাবিকভাবেই এটা তিনি মেনে নিতে পারেন নি। (✓)
(Input) তিনি অনুর্ধ্ব১৯ বিশ্বকাপ খেলেন ২০০০ সালে
(BanglaT5) তিনি অনুর্ধ্ব১৯ বিশ্বকাপ খেলেন ২০০০ সালে। (×)
(T5-Small) তিনি অনুর্ধ্ব১৯ বিশ্বকাপ খেলেন ২০০০ সালে। (×)
(Jatikarok) তিনি অনুর্ধ্ব-১৯ বিশ্বকাপ খেলেন ২০০০ সালে। (✓)
(Input) কিছু বিবরণ অনুসারে এই সংখ্যা ১২০০০০১৩০০০০
(BanglaT5) কিছু বিবরণ অনুসারে, এই সংখ্যা ১২০০০০১৩০০০০। (×)
(T5-Small) কিছু বিবরণ অনুসারে এই সংখ্যা ১২০০০০১৩০০০০। (×)
(Jatikarok) কিছু বিবরণ অনুসারে, এই সংখ্যা ১২০,০০০,১৩০,০০০। (×)

Table 2: The qualitative performance of different transformer-based methods.

methods fail, as demonstrated in the last example. For an erroneous input ”কিছু বিবরণ অনুসারে এই সংখ্যা ১২০০০০১৩০০০০”, Jatikarok generated output ”কিছু বিবরণ অনুসারে, এই সংখ্যা ১২০,০০০,১৩০,০০০।”, where the actual correction is ”কিছু বিবরণ অনুসারে, এই সংখ্যা ১২০,০০০-১৩০,০০০!”, which is superior to the corrections made by the other two methods. It accurately reinstated a comma between two words in the middle of the sentence (...অনুসারে, এই...), a task where the other two methods failed. Moreover, it also added commas in the number (১২০,০০০,১৩০,০০০) to enhance readability, a feat the other two methods did not accomplish.

5.5 Ablation Study

Table 3 illustrates how model performance improves with larger corpus sizes. The corpus consisting of 1.5M instances displayed the most substantial performance, while the corpus containing 148.1K instances showed the least significant performance. The corpus consisting of 740.5K instances demonstrated intermediate performance, surpassing the smaller corpus size but falling short of the 1.5M corpus. A clear pattern emerges: larger corpus sizes correspond to improved performance. The 1.5M corpus achieved an impressive accuracy of 95.2%, surpassing the 740.5K corpus by 5.48%, and the 148.1K corpus by 11.89%.

Method	Corpus Size	Inference				
		Acc	PR	RE	F1	F0.5
Jatikarok	148.1K	83.31%	0.833	0.833	0.833	0.834
Jatikarok	740.5k	89.72%	0.897	0.897	0.896	0.897
Jatikarok	1.48M	95.2%	0.953	0.952	0.952	0.953

Table 3: The impact of the corpus size on our proposed method.

6 Conclusion

This study addressed the primary obstacle hindering the progress of the task by introducing a comprehensive baseline. Specifically, we introduced the groundbreaking Jatikarok, a monolingual transformer-based method meticulously designed to harness the power of transfer learning by adapting knowledge from Bangla grammatical error correction to effectively tackle intricate linguistic patterns inherent to this specific task. Furthermore, the efficacy of our proposed Jatikarok is validated across various corpora, solidifying its status as a state-of-the-art method for this task by outperforming BanglaT5 and T5-Small. In conjunction with the model, a substantial parallel corpus containing 1.48M source-target pairs has been made publicly accessible, which has been carefully curated by incorporating 16 Bangla punctuation marks. Consequently, this resource eliminates the scarcity of materials for Bangla, effectively transforming it into a language well-equipped for punctuation tasks. In our future study, we will empirically investigate the effectiveness of knowledge distillation through the transfer of knowledge from the multilingual model to our monolingual model.

References

Ajwad Akil, Najrin Sultana, Abhik Bhattacharjee, and Rifat Shahriyar. 2022. Banglaparaphrase: A high-quality bangla paraphrase dataset. *arXiv preprint arXiv:2210.05109*.

Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Punctuation restoration using transformer models for high-and low-resource languages. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 132–142.

Adebayo Mustapha Bakare, Kalaiarasi Sonai Muthu Anbananthen, Saravanan Muthaiyah, Jayakumar Krishnan, and Subarmaniam Kannan. 2023. Punctuation restoration with transformer model on social media data. *Applied Sciences*, 13(3):1685.

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

Pramit Bhattacharyya, Joydeep Mondal, Subhadip Maji, and Arnab Bhattacharya. 2023. Vacaspati: A diverse corpus of bangla literature. *arXiv preprint arXiv:2307.05083*.

Mehedi Hasan Bijoy, Nahid Hossain, Salekul Islam, and Swakkhar Shatabda. 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*.

Varnith Chordia. 2021. Punktuator: A multilingual punctuation restoration system for spoken and written text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 312–320.

Maury Courtland, Adam Faulkner, and Gayle McElvain. 2020. Efficient automatic punctuation restoration using bidirectional transformers with robust inference. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 272–279.

Xue-Yong Fu, Cheng Chen, Md Tahmid Rahman Laskar, Shashi Bhushan TN, and Simon Corston-Oliver. 2021. Improving punctuation restoration for speech transcripts via external data. *arXiv preprint arXiv:2110.00560*.

Nuno Miguel Guerreiro, Ricardo Rei, and Fernando Batista. 2021. Towards better subtitles: A multilingual approach for punctuation restoration of speech transcripts. *Expert Systems with Applications*, 186:115740.

Qiushi Huang, Tom Ko, H Lilian Tang, Xubo Liu, and Bo Wu. 2021. Token-level supervised contrastive learning for punctuation restoration. *arXiv preprint arXiv:2107.09099*.

Seokhwan Kim. 2019. Deep recurrent neural networks with layer-wise multi-head attentions for punctuation restoration. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7280–7284. IEEE.

Viet Dac Lai, Abel Salinas, Hao Tan, Trung Bui, Quan Tran, Seunghyun Yoon, Hanieh Deilamsalehy, Franck Dernoncourt, and Thien Huu Nguyen. 2023. Boosting punctuation restoration with data generation and reinforcement learning. *arXiv preprint arXiv:2307.12949*.

Karan Makhija, Thi-Nga Ho, and Eng-Siong Chng. 2019. Transfer learning for punctuation prediction. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 268–273. IEEE.

Syed Mostofa Monsur, Sakib Chowdhury, Md Shahrar Fatemi, and Shafayat Ahmed. 2022. Shonglap: A large bengali open-domain dialogue corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5797–5804.

Attila Nagy, Bence Bial, and Judit Ács. 2021. Automatic punctuation restoration with bert models. *arXiv preprint arXiv:2101.07343*.

Binh Nguyen, Vu Bao Hung Nguyen, Hien Nguyen, Pham Ngoc Phuong, The-Loc Nguyen, Quoc Truong Do, and Luong Chi Mai. 2019. Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging. In *2019 22nd conference of the oriental COCOSDA international committee for the co-ordination and standardisation of speech databases and assessment techniques (O-COCOSDA)*, pages 1–5. IEEE.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Habibur Rahman, Md Rezwan Shahrior Rahin, Araf Mohammad Mahbub, Md Adnanul Islam, Md Saddam Hossain Mukta, and Md Mahbubur Rahman. 2023. Punctuation prediction in bangla text. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(3):1–20.

Ning Shi, Wei Wang, Boxin Wang, Jinfeng Li, Xiangyu Liu, and Zhouhan Lin. 2021. Incorporating external pos tagger for punctuation restoration. *arXiv preprint arXiv:2106.06731*.

Monica Sunkara, Srikanth Ronanki, Kalpit Dixit, Sravan Bodapati, and Katrin Kirchhoff. 2020. Robust prediction of punctuation and truecasing for medical asr. *arXiv preprint arXiv:2007.02025*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer.

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

Yangjun Wu, Kebin Fang, and Yao Zhao. 2022. A context-aware feature fusion framework for punctuation restoration. *arXiv preprint arXiv:2203.12487*.

Jiangyan Yi and Jianhua Tao. 2019. Self-attention based model for punctuation prediction using word and speech embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7270–7274. IEEE.

Jiangyan Yi, Jianhua Tao, Ye Bai, Zhengkun Tian, and Cunhang Fan. 2020. Adversarial transfer learning for punctuation restoration. *arXiv preprint arXiv:2004.00248*.