Punctuation Prediction in Bangla Text

HABIBUR RAHMAN*, MD. REZWAN SHAHRIOR RAHIN*, and
ARAF MOHAMMAD MAHBUB, United International University, Bangladesh
MD. ADNANUL ISLAM, Monash University, Australia
MD. SADDAM HOSSAIN MUKTA, United International University, Bangladesh
MD. MAHBUBUR RAHMAN, Military Institute of Science and Technology, Bangladesh

Punctuation prediction is critical as it can enhance the readability of machine-transcribed speeches or texts significantly by adding appropriate punctuation. Furthermore, systems like **Automatic Speech Recognizer (ASR)** produce texts that are unpunctuated, making the readability difficult for humans and also hampers the performance of various **natural language processing (NLP)** tasks. Such NLP related tasks have been investigated thoroughly for English; however, very limited work is done for punctuation prediction in the Bangla language. In this study, we train a **bidirectional recurrent neural network (BRNN)** along with Attention model with a plausibly large Bangla dataset. Afterwards, we apply extensive postprocessing techniques for predicting punctuation more accurately with the employed model. Initially, we perform experimentation with a relatively imbalanced dataset, and our model shows promising results ($F1 = 56.9$ for Period) in punctuation prediction. Later, we also investigate the model's performance using a balanced Bangla dataset to achieve higher performance scores ($F1 = 62.2$ for Question). Thus, the goal of this study is to propose an efficient approach that can predict punctuation in Bangla texts effectively. Our study also includes investigation on how our postprocessing techniques affect the prediction performance. Being an early attempt for the punctuation prediction in Bangla text, our work is expected to significantly contribute in the NLP field for the Bangla language, and will pave the way for future work with the Bangla language in this direction.

CCS Concepts: • **Information systems** → *Document representation*; • **Computing methodologies** → **Natural language processing;**

Additional Key Words and Phrases: Neural networks, punctuation prediction, natural language processing, BRNN

*Joint First Authors.

Funding. The authors did not receive support from any organization for the submitted work. No funding was received to assist with the preparation of this manuscript. No funding was received for conducting this study. No funds, grants, or other support was received.

Conflicts of interest. The authors have no relevant financial or non-financial interests to disclose. The authors have no conflicts of interest to declare that are relevant to the content of this article. The authors have no financial or proprietary interests in any material discussed in this article.

Authors' addresses: H. Rahman, Md. R. S. Rahin, A. M. Mahbub, and Md. S. H. Mukta, United International University, United City, Dhaka-1200, Bangladesh; emails: habiburrahmanshamimm@gmail.com, rezwanshahriorrahin@gmail.com, araf@gtaf.org, saddam@cse.uiu.ac.bd; Md. A. Islam, Monash University, Clayton VIC 3800, Australia; email: Adnan.Islam@monash.edu; Md. M. Rahman, Military Institute of Science and Technology, Mirpur Cantonment, Dhaka-1216, Bangladesh; email: mahbub@cse.mist.ac.bd.

2375-4699/2023/03-ART81 $15.00
https://doi.org/10.1145/3575804

ACM Reference format:
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 Trans. Asian Low-Resour. Lang. Inf. Process.* 22, 3, Article 81 (March 2023), 20 pages.
https://doi.org/10.1145/3575804

1 INTRODUCTION

Natural Language Processing (NLP) is one of the major challenges of artificial intelligence that bridges the interaction between natural human languages and computing devices. The applications of NLP are widespread in modern times and are expected to see continuous developments in the future. NLP has paved the way for real-world applications such as automatic text summarization, sentiment analysis, **parts-of-speech (PoS)** tagging, machine translation, and so on [11, 13, 25]. The processing of natural languages is quite complex for a machine to master due to the complicated nature of human-understandable words linking together to make certain meanings [14, 15]. However, through rigorous research in this field, more and more systems are integrating NLP-based applications that have become more cohesive in our daily interaction with machines.

To perceive the meaning of a sentence, punctuation marks play a vital role to direct where one should stop, pause, and show emotions. Systems like **Automatic Speech Recognition (ASR)** generate unpunctuated texts, and restoring punctuation in those texts not only improves the readability but also can assist in the consequent processing tasks such as machine translation [29], sentiment analysis [28], and various fields of NLP. Furthermore, punctuation prediction has a wide range of pragmatic applications. For example, proofreading of books can be done with such punctuation predicting systems; thus, reducing cost and time for publishing books with high accuracy [23]. Moreover, the systems that require translations may have a monumental effect in terms of producing meaningful machine translations [29].

There are numerous studies focusing on punctuation prediction and restoration (e.g., [5, 10, 19, 35, 38]); however, not much work is done previously for Bangla language; in other words, there is yet a lot to explore and work to be done in Bangla language [16, 25]. Furthermore, there are a very limited number of publicly available effective Bangla punctuation datasets. To this end, we create two Bangla datasets and make them publicly available. These datasets will not only facilitate works related to punctuation prediction or restoration, but also will benefit Bangla NLP community as a whole through advancing various NLP related tasks. Moreover, in the existing works, the Exclamation mark is often ignored, even though it has its own significance in the representation of human expressions to various extents in various situations. More specifically, no work can be found for punctuation prediction or restoration in Bangla text considering Exclamation as a punctuation mark. Hence, this is one of the key contributions of this work, as we focus on Exclamation mark prediction as well.

Bangla is one of the most widely spoken languages with its enriched vocabulary and strict grammar essence. With over 228 million native and more than 37 million second-language speakers, Bangla is the fifth most spoken native language and the seventh most spoken language in the world [13]. To curtail this significant research gap in a top language like Bangla, we make the following major contributions in this study:

- We propose a recurrent neural network approach for punctuation prediction in Bangla text. Particularly, we propose the first-ever attempt for predicting the Exclamation mark in Bangla text.
- We propose and experiment extensively with several postprocessing techniques to improve the prediction performance in both Bangla and English text.

- We experiment with both Bangla and English languages, and present benchmark performance scores for punctuation prediction particularly in Bangla text, to drive future research in this direction.
- We prepare datasets (particularly, a balanced dataset) for punctuation prediction or restoration in Bangla text, and make them publicly available to facilitate future research with the Bangla language.

2 RELATED WORK

There are numerous works related to punctuation prediction and restoration based on various approaches. Even though these studies have been done for many languages previously, Bangla is not one of them. For the punctuation restoration in Bangla, only one work can be found so far (Alam et al. [3]), where they primarily explored different transformer models and proposed an augmentation technique improving performance on noisy ASR texts significantly. However, they only considered Comma, Period, and Question marks as the target punctuation. In general, amongst the many strategies, the ones that come up the most for predicting punctuation, are primarily based on language modeling task, sequence labeling task, and monolingual machine translation problem. Major features for this task include lexical, acoustic, and a combination of both lexical and acoustic features. Although hybrid feature based models deliver better performance than models based on only lexical or acoustic features; training data for the hybrid model is scarce. Contrarily, the lexical models can take any textual material as training data.

For language modeling tasks, Beeferman et al. [6] proposed a Hidden Markov model, relying exclusively upon lexical features, to insert Commas in texts, consisting of a trigram language model and Viterbi decoder. When it comes to sequencing modeling tasks, the approaches are mostly based on **conditional random fields (CRFs)** and deep learning. Using CRFs, Pham et al. [30] proposed the first Vietnamese punctuation prediction system, where they showed various combinations of features, i.e., they illustrated the effects of different combinations of features - starting with unigram word feature followed by bigram word feature along with label transition feature. Liu et al. [20] proposed a novel three-stage **long short-term memory (LSTM)** based model to predict punctuation in the Chinese language, where they trained the model with lexical features along with two linguistic features (PoS tags and chunking) followed by pause duration features. Afterwards, pitch information was fused with the lexical and pause features, which yielded improved results. Tilk et al. [37] introduced a **bidirectional recurrent neural network (BRNN)** model with an attention mechanism to restore punctuation in the unsegmented text, taking lexical and acoustic features combined, enabling their model to make use of variable length contexts before and after the current position in the text, and showing significant performance in predicting punctuation. Besides, Oktem et al. [27] proposed a BRNN model using both lexical and acoustic features simultaneously to predict punctuation, where they processed both lexical and prosodic information in parallel to predict punctuation. To analyze the influence of acoustic features on punctuation prediction, their model extracted three main acoustic elements: pause, intensity, and fundamental frequency.

Additionally, there have also been works related to punctuation in medical reports; For instance, Salloum et al. [32] proposed a method for punctuation restoration in medical reports using a BRNN architecture with an attention mechanism and late fusion. Due to the nature of such dataset, consisting of a large amount of domain-specific vocabulary, a novel vocabulary reduction model was used. Moreover, Kim [18] proposed a new neural network architecture for punctuation restoration based on stacked BRNN with layer-wise multi-head attention (DRNN-LWMA), which aimed for better contextual learning from various aspects. In recent times, bidirectional LSTMs

have exhibited impressive results. For example, Juin et al. [17] proposed a BRNN model with attention and PoS tags to add punctuation to unpunctuated sentences automatically. Moro et al. [24] proposed to predict punctuation marks using a prosody based model that is resource-efficient (low computational requirements) and more robust to ASR errors with low latency, which is suitable for real-time operations. They performed an automatic **phonological phrase (PP)** alignment to extract the type and duration of the phonological phrase and also the pause duration from ASR. Zelasko et al. [45] trained two variants of **deep neural network (DNN)** sequence labeling models, bidirectional LSTM and **convolutional neural network (CNN)**, to predict the punctuation in the domain of conversational speech. They proposed to utilize the information from both sides of the conversation, relative timing, and duration of each word to predict punctuation.

Apart from these, Tundik et al. [38] proposed a joint word-level and character-level model for punctuation recovery by using both CNN and RNN. They came up with a hybrid model composed of word embedding and character embedding, showing positive results. Nanchen et al. [26] proposed a technique for combining punctuation prediction models, and showed results of individual and combined models based on an empirical evaluation. Szaszák et al. [35] proposed a model to consider character, word, and prosody based features simultaneously, to implement a highly language-independent robust platform for punctuation restoration, which can also deal well with highly agglutinating languages with less constrained word order. Their hybrid model is a triple of character, word, and prosody based model with a combined BiLSTM architecture.

Makhija et al. [21] presented a transfer learning approach, where they proposed a sequence labeling architecture consisting of two layers composed of pre-trained BERT model and BiLSTM along with CRF classifier, coming up with one of the most promising performances in this task.

For monolingual machine translation, the models are based on self-attention and transformer models. Varavs et al. [39] proposed a system using a novel transformer model for punctuation and capitalization restoration along with the use of self-attention, achieving significant results for the Latvian language. Furthermore, Wang et al. [40] proposed a novel self-attention based lightweight neural network. They utilized lexical features only and followed the transformer model that relies solely on the self-attention layers. Yi et al. [43] proposed a self-attention based model getting motivated by the excellent results of self-attention models. Hence, although numerous studies achieved better performance using BRNN or BiLSTM models for different languages, no related study can be found for the Bangla language using these models.

3 METHODOLOGY AND DESIGN

As there is relatively less number of works related to Bangla punctuation prediction, we take the initiative on implementing a compatible approach at this stage. Based on our reviews of various literatures and articles, we find that the sequence-to-sequence model shows convincing results. We investigate various sequence-to-sequence models and finally select the bidirectional recurrent neural network with attention approach, which we call BiAttention model.

3.1 System Architecture

The first step starts with Bangla dataset collection, followed by a preprocessing and embedding step. Then, we train our BiAttention model by feeding the preprocessed input words. In our model, the bidirectional method allows us to train our model with both forward and backward context of a sequence [33]. Convincing results from several previous works [17, 37, 38, 42, 45] with Bidirectional approach motivate us to choose this approach. The Attention mechanism [4] increases our model accuracy by providing greater focus to the most relevant part. We use **gated recurrent unit (GRU)** [8] as a recurrent cell. GRU generally performs well to capture long-range dependencies; at the

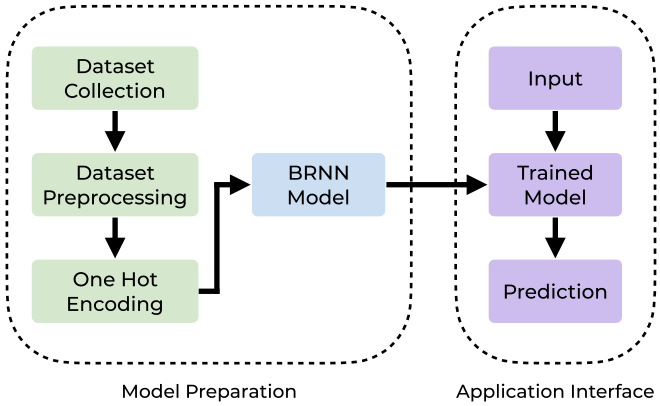

Fig. 1. Proposed system architecture for punctuation prediction in Bangla text. The system has two phases. Preparing the BiAttention model is the key phase, where the interface is used to test the model.

same time, it is simpler with a fewer number of parameters and provides benefits like LSTM [7, 9, 12, 34]. We use a one hot encoding approach [31] to represent word text to a machine-understandable form. Figure 1 depicts our overall system architecture.

3.2 Neural Network Architecture

As mentioned earlier, we are using the GRU as a recurrent cell. GRU provides a less complex and effective implementation with excellent results. Tanh is used as a nonlinear activation function to handle memory content.[1] Next, we discuss our neural network on how it proceeds with the input (Figure 2).

At time t, the pre-processed one hot encoded input word sequences $X = (x_1, x_2, x_3, \ldots, x_N)$ of length N is given as input in our model. The output of the model y_t represents the punctuation probability between the words x_t and x_{t-1}. A bidirectional GRU works as the recurrent layer here. Each bidirectional layer consists of two sublayers at the same time - a forward recurrent layer $\overrightarrow{h_t}$ and a backward recurrent layer $\overleftarrow{h_t}$. These two GRU layers are preceded by a shared embedding layer with weight W_e. Since we are using One Hot Encoding as the embedding layer, the size of the embedding vector is set the same as the vocabulary sizes. Equations (1) and (2) represent the hidden state of the GRU at time t without the bias.

$$\overrightarrow{h_t} = GRU(x_t\, W_e,\ \overrightarrow{h_t}) \tag{1}$$

$$\overleftarrow{h_t} = GRU(x_t\, W_e,\ \overleftarrow{h_t}) \tag{2}$$

Here, the backward recurrent layer is computed similar to the forward recurrent layer, except the input sequence X that is given in reverse order. Equation (3) shows that the output from the bidirectional layer is concatenated to produce a state h_t and is passed as input to a unidirectional GRU. The unidirectional GRU has been used to synchronize the bidirectional layer output with input sequences. Equation (4) shows the output state s_t from the unidirectional GRU unit.

$$h_t = [\,\overrightarrow{h_t},\ \overleftarrow{h_t}\,] \tag{3}$$

$$s_t = GRU(h_t,\ s_{t-1}) \tag{4}$$

[1]We choose Tanh over another popular activation function, ReLU; since ReLU can have very large outputs (i.e., exploding nature).

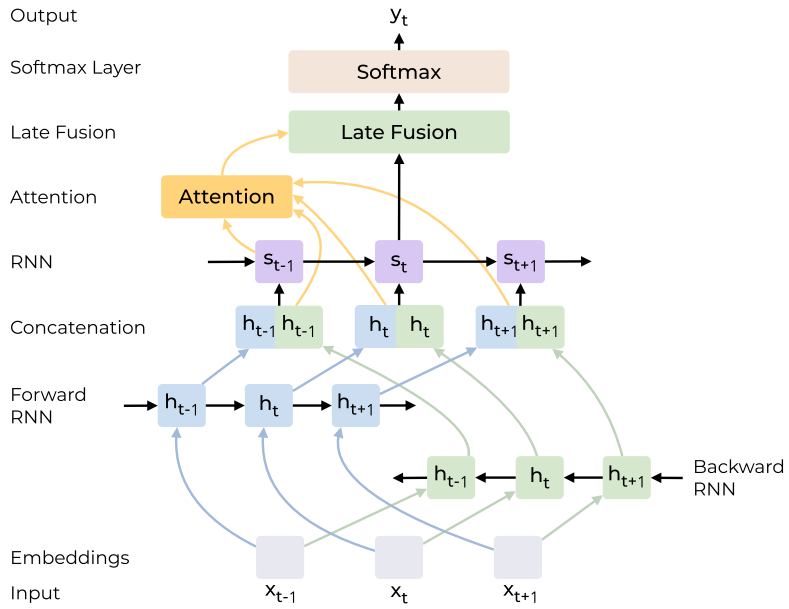

Fig. 2. BiAttention model architecture of the proposed system.

Equation (5) shows the Attention mechanism, which combines all the input states into a weighted vector a_t.

$$a_t = \sum_{i=1}^{N} \alpha_{(t,\ i)}\ h_t) \tag{5}$$

Here, $\alpha_{(t,i)}$ is the weight, which indicates the amount of influence of each input state as described in [4]. The output of the unidirectional layer and attention state is then late fused [41] to produce the output f_t using Equation (6), where a_t is the attention calculated from Equation (5) and s_t is the state of the GRU derived from Equation (4) at time step t.

$$f_t = a_t W_{fa}\ o\ \sigma(a_t W_{fa} W_{ff} + s_t W_{fs} + b_f) + s_t \tag{6}$$

Finally, the punctuation probabilities before each encoded word (except the first word from a sequence) are generated using the softmax function shown in Equation (7). We use softmax (instead of sigmoid) here since there are more than two punctuation signs (i.e., classes). A detailed graphical representation of our neural network architecture is reflected in Figure 2.

$$y_t = Softmax(f_t W_y + b_y) \tag{7}$$

Table 1 demonstrates an illustrative example on how our model predicts punctuation in a given sentence as text. Let us consider the unpunctuated input – হঠাৎ বৃষ্টি বাহিরে যাবো সবাই বললো যাবে না (*English transcription: Hothath brishti bahire jabo sobai bollo jabe na*) with a <END> token added at the end of the sequence. Recall, the output of the model y_t represents the punctuation probability between the words x_t and x_{t-1}, excluding the first word of the sequence. w_n is the weight of the punctuation probable to occur before the particular word, and the punctuation mark with the maximum weight, w_n (in square brackets), is selected by the model. Henceforth, the model would output – হঠাৎ বৃষ্টি! বাহিরে যাবো? সবাই বলল, যাবে না। *(English transcription: Hothath brishti! bahire jabo? sobai bollo, jabe na.)*. Notice that the output has several punctuation marks as predicted by our model as shown in Table 1.

Table 1. An Illustrative Example for Punctuation Prediction in a Given Text

Word	**Space(_)**	[Comma(,)]	**Period(|)**	**Question(?)**	**Exclamation(!)**	**Prediction**
Hothath	–	–	–	–	–	–
brishti	[w_1]	w_2	w_3	w_4	w_5	_SPACE
bahire	w_1	w_2	w_3	w_4	[w_5]	!EXCLAMATION
jabo	[w_1]	w_2	w_3	w_4	w_5	_SPACE
sobai	w_1	w_2	w_3	[w_4]	w_5	?QUESTION
bollo	[w_1]	w_2	w_3	w_4	w_5	_SPACE
jabe	w_1	[w_2]	w_3	w_4	w_5	,COMMA
na	[w_1]	w_2	w_3	w_4	w_5	_SPACE
<END>	w_1	w_2	[w_3]	w_4	w_5	|PERIOD

3.3 Postprocessing Method

We conduct several postprocessing analyses and study how it impacts the performance of the model. Our postprocessing approaches aim to improve our BiAttention model's prediction performance through putting or replacing punctuation marks after some previously analyzed words. More specifically, we gather separate lists of top words (for each of the four punctuation marks),[2] after which the punctuation marks occur most in our dataset. Then, we take five mini-batches from the test dataset each consisting of randomly taken but unique 10,000 sentences. Next, we undergo the following steps for postprocessing.

First, we input each of these mini-batches into our model to predict the punctuation marks. Then, the prediction of our model is further post-processed by putting and replacing punctuation marks after the enlisted top words for each of the punctuation marks both separately and combinedly. More specifically -

(1) We consider the punctuation postprocessing of the mini-batches using the top words for a particular punctuation, for example, Comma. Hence, each of those top words (from the Comma's list only), which appears in a test sentence, are followed by a Comma through either replacing other punctuation marks or appearing newly. Similar techniques are also applied considering the Period, Question, and Exclamation marks independently.
(2) Additionally, we perform this experimentation by applying postprocessing for all the punctuation marks sequentially, in a feed-forward manner, not independently. For example, after being postprocessed using Comma's list, the resulting text is next postprocessed by Period's list, Question's list, and Exclamation's list, respectively. This is how we perform postprocessing using the top words for all the punctuation marks combined.

Next, choosing the appropriate number of top words becomes a major issue here. Therefore, we experiment with different numbers of top words for each punctuation separately, after which the corresponding punctuation is put or replaced during postprocessing. Precisely, we consider (1) the topmost word (1 word), (2) top 3 words, (3) top 5 words, (4) top 7 words, and (5) top 9 words, respectively.

We rigorously analyze the results from each of the steps (i.e., top 1, 3, 5, 7, 9 words, respectively) for each punctuation to discover the best one(s). We find no motive behind choosing more than

[2]These four Bangla punctuation marks are - Comma (,), Period (|), Question (?), and Exclamation (!).

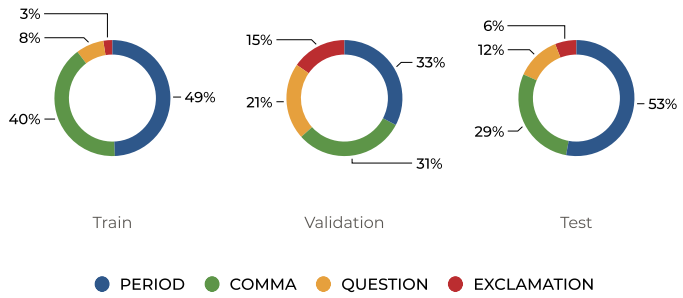

Fig. 3. Visualization of distribution, in percentage, for each of the four punctuation symbols (Period, Comma, Question, Exclamation) in Bangla OPUS dataset.

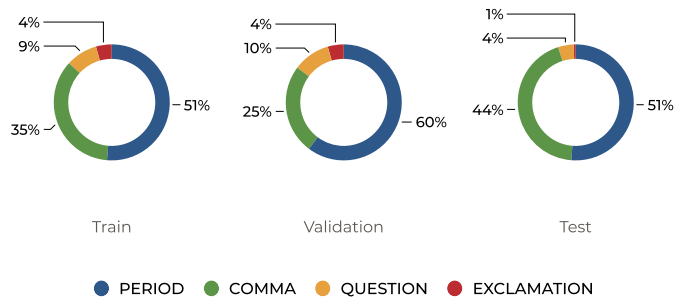

Fig. 4. Visualization of distribution, in percentage, for each of the four punctuation symbols (Period, Comma, Question, Exclamation) in the English OPUS dataset.

top 9 words due to witnessing no possible improvement in the model's performance. Besides, since we find no significant differences in performance scores considering top 2, 4, 6, 8 words, we only present the results for top 3, 5, 7 words (i.e., most relevant results). The results are shown in the later section.

4 EXPERIMENTAL EVALUATION

4.1 Dataset Collection

We train three models to conduct our analysis, two models for Bangla and one model for English. To feed our models with enough text samples, we collect two large Bangla and one English dataset. The English dataset and one of the Bangla datasets are collected from OPUS [36], and another Bangla dataset is primarily collected from the Prothom-Alo archive [44]. The Bangla and English datasets from OPUS contain more than 1 million sentences and the Prothom-Alo dataset contains more than 8 million sentences. The OPUS datasets contain text from various sources such as Bangla articles, conversations, translations, and so on, and the Prothom-Alo dataset comprises of news articles from various domains (e.g., politics, health, education, etc.). As the sources and domains vary a lot, the datasets required considerable prepossessing.

We divide the datasets into three separate sets: training, validation, and test sets. The training sets for the Bangla and English models contain more than 600K and 700K sentences, respectively, whereas the validation and test sets both contain 200K sentences each for Bangla, and for English, they contain 50K sentences each, with no data overlapping. Figures 3 and 4 show the distributions of the punctuation marks in training, validation, and test sets of the Bangla and English OPUS datasets.

Table 2. Distributions of Punctuation in Train, Validation, and Test Sets in All Three Datasets

Dataset	Punctuation	Training Set	Validation Set	Test Set
Bangla-OPUS	Period	397972	55546	114044
	Comma	325699	52167	62213
	Question	62834	36252	26636
	Exclamation	20448	26065	12894
English-OPUS	Period	674444	43348	99696
	Comma	466556	17979	85365
	Question	115654	7443	8450
	Exclamation	57869	3204	1133
Bangla-Balanced	Period	90003	29991	30007
	Comma	90570	29221	30438
	Question	90001	30000	30000
	Exclamation	90003	29998	30000

It is noted that the distributions of the OPUS datasets are not well-balanced for each punctuation, and the most occurred punctuation mark is Period. In order to investigate the results on a balanced dataset, we prepare a new dataset from the Prothom-Alo news archive. This new Bangla dataset contains nearly equal distribution of all punctuation marks. Afterwards, we perform preprocessing of all three datasets–we denote the balanced dataset as Bangla-Balanced and the other two datasets as Bangla-OPUS and English-OPUS.[3] Table 2 shows the punctuation frequencies in all these three datasets in terms of training, validation, and test sets.

4.2 Dataset Preprocessing

As part of the preprocessing step, we run the same procedure for each of our three datasets. Firstly, we remove special characters and then normalize the URLs, emails, hashtags, and so on. The majority of the sentences in our dataset comprise fewer than 50 words. Then, we tokenize the numeric characters present in the dataset using the <NUM> tag. Then, we separate all punctuation marks and represent them as a single character. We work with four punctuation marks and add punctuation names along with the punctuation characters, e.g., the Comma character ',' are replaced with ',COMMA', Question mark '?' with '?QUESTIONMARK', and so on. Figure 5 shows a snippet of the dataset, before and after performing the preprocessing task.

These cleaned and formatted datasets then produce vocabularies of 100K, 74K, and 97K word tokens from Bangla-OPUS, English-OPUS, and Bangla-Balanced datasets, respectively. We use the one hot encoding technique to represent our word vocabulary. One hot encoding is widely used for its re-scalability; thus determining the probability that is spontaneous for the model [37]. In particular, the work of Tilk et al. [37] motivates us to choose one hot encoding, as their work shows that one hot encoding produces significant results for punctuation prediction with BiAttention model. Here, the encoded value for each word is the same as the index of the word in the vocabulary list. In Figure 6, the first line represents a sequence of one hot encoded words, and the next line represents the corresponding punctuation mark after each word. The punctuation map used here

[3]We make all these datasets publicly available: https://github.com/hrahmansha/Punctuation-Prediction-Datasets.

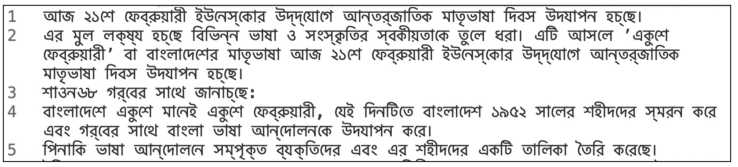

Fig. 5. A segment of the Bangla-OPUS dataset after preprocessing.

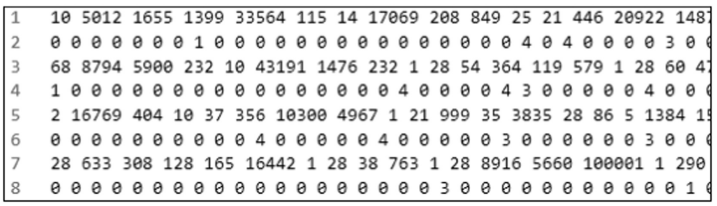

Fig. 6. One hot encoding representation of the dataset.

Table 3. Model Parameters

Parameter	Value
Hidden Layer Size	212
Learning Rate	0.02
Number of Iteration	5
Maximum Input Sequence	200
Minibatch Size	32

is - Comma mark to 1, Period mark to 2, Question mark to 3, and Exclamation mark to 4. '0' represents a space between words, instead of any punctuation.

4.3 Experimental Settings and Evaluation Metrics

Our BiAttention model is implemented using the Tensorflow toolkit [1]. We train our model inside the Google Colaboratory system. We choose our model parameters based on our literature review and after running several simulations of our model. Table 3 lists the best-suited parameters for our model.

Punctuation distribution in a dataset can be highly imbalanced. Due to such imbalanced nature of punctuation datasets, accuracy could be a misleading metric. Along with the accuracy, we evaluate the performance of our system using evaluation metrics - Precision, Recall, F1-score, and **Slot**

Error Rate (SER). The Slot Error Rate described in [22] can be a good measure when F1-score exhibits some undesirable properties. Equations (8)–(11) show how we calculate Accuracy, Precision, Recall, and F1-score to evaluate our results, where we denote True Positive as TP, True Negative as TN, False Positive as FP, and False Negative as FN.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Here,
TP = Number of correct predictions for the presence of target punctuation
TN = Number of correct predictions for the absence of target punctuation
FP = Number of wrong predictions for the punctuation that is not actually present (i.e., absent)
FN = Number of wrong predictions for the punctuation that is not actually absent (i.e., present).

5 RESULTS AND DISCUSSIONS

We train our model with three different datasets separately, and refer to the trained models as BiAttention-Bangla, BiAttention-English, and BiAttention-Balanced. Next, we present the results for all these datasets successively.

5.1 OPUS Datasets

First, we present the results obtained using Bangla-OPUS and English-OPUS datasets. The initial results obtained directly (before postprocessing) from our BiAttention-Bangla model is depicted in Figure 7. Note that, we show accuracy, precision, recall, and F1-score for prediction of each punctuation separately and combinedly (denoted as 'OVERALL' in the figure). We find the highest scores ($P = 62.5$, $R = 52.3$, $F1 = 56.9$) for predicting Period independently. Although our model performs plausibly well also for Comma ($P = 43.1$) and Question ($P = 51.1$), it performs poorly while predicting Exclamation ('!'). One of the reasons behind this is, the frequency of Exclamation marks in this dataset is relatively less than that of other punctuation marks. This data imbalance affects the slot error rate as well, which causes a higher SER value of 77.9% for the BiAttention-Bangla model.

Similarly, Figure 8 represents the results for the BiAttention-English model in terms of accuracy, precision, recall, and F1-score. This model achieves higher scores for Period ($P = 72.1$, $R = 67.1$, $F1 = 69.5$) and Question ($P = 75.3$, $R = 68.9$, $F1 = 72.0$) marks, but the performance for Comma is not in the region of Period and Question marks. Besides, the model significantly suffers while predicting the Exclamation marks. The BiAttention-English model achieves the SER of 55.6%.

From the above results, it is noticeable that the BiAttention-English model faces much difficulty while restoring the Exclamation mark. Although the Exclamation mark prediction result for BiAttention-Bangla model ($P = 24.1$, $R = 16.4$, $F1 = 19.5$) is higher than BiAttention-English model ($P = 13.0$, $R = 16.1$, $F1 = 14.4$), the performance of both models is still at a lower level. Moreover, the performance scores of BiAttention-Bangla model for the prediction of other punctuation marks are lower than that of BiAttention-English model. To further illustrate the

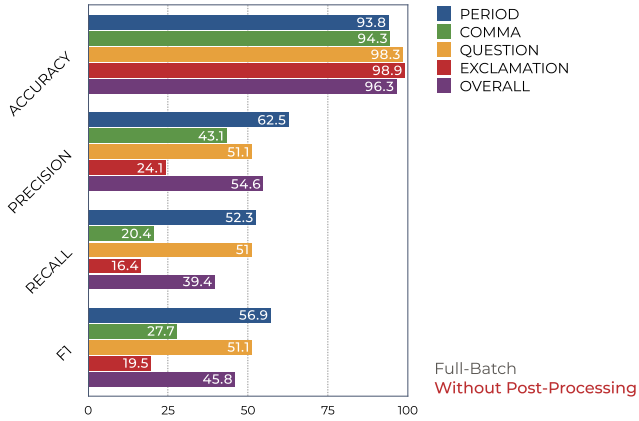

Fig. 7. A representation of individual and overall Accuracy, Precision, Recall, and F1-score (in %) achieved by the BiAttention-Bangla model. The model performs well particularly for Period prediction ($F1 = 56.9$) in Bangla text.

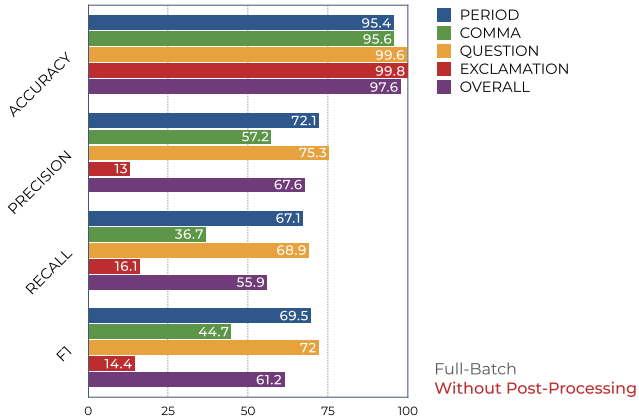

Fig. 8. A representation of individual and overall Accuracy, Precision, Recall, and F1-score (in %) achieved by the BiAttention-English model. The model performs well specifically for Period ($F1 = 69.5$) and Question ($F1 = 72.0$) marks prediction in English text.

results, we also generate the confusion matrix from the outputs of BiAttention-Bangla and BiAttention-English models. Figures 9 and 10 show the corresponding confusion matrix of BiAttention-Bangla and BiAttention-English models, respectively. Another important point here to be noted is the accuracy value; the accuracy achieved for both individual and overall evaluation (Figures 7 and 8) seems unaligned with the value of other metrics, as the values are incredibly high. To this end, from the confusion matrix, we can see the true negative value ('Others') for each punctuation becomes high for an NLP task like punctuation prediction. Thus, looking into the confusion matrix and the accuracy that we got from BiAttention-Bangla and BiAttention-English, it is notable that accuracy can be a misleading measure to evaluate punctuation prediction models.

However, the results displayed so far are without any postprocessing approaches. Next, we present the F1-scores after our postprocessing analyses over one mini-batch (of test data). In Figure 11, we show the postprocessing results of BiAttention-Bangla model on mini-batch 3 for each punctuation separately and combined (i.e., OVERALL), considering top 3, 5, 7 words from

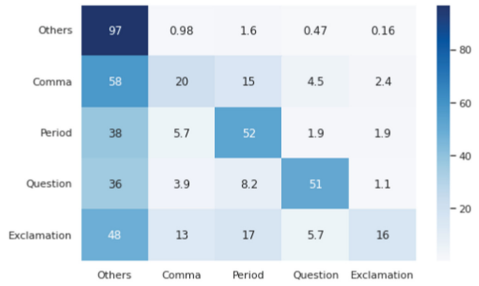

Fig. 9. Confusion matrix for the BiAttention-Bangla model.

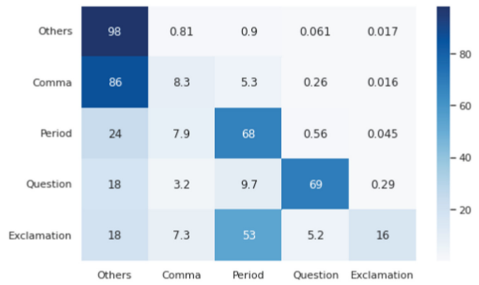

Fig. 10. Confusion matrix of the BiAttention-English model.

each punctuation's list of top words, along with the corresponding scores before postprocessing (i.e., ORIGINAL). For example, the graph at the top-left corner in Figure 11 shows results for each punctuation considering top 3, 5, 7 words respectively only from the Period's list of top words (i.e., top Period words), along with its original score (using the BiAttention-Bangla model). Carefully note that, during postprocessing, considering the top words of a particular punctuation (for example, Comma), all four punctuation marks can get affected, e.g., a Comma may be newly added, and a Comma may also replace a Period, a Question, and/or a Exclamation. That is why we show the results for all four punctuation marks separately and combined, when post-processed with a particular punctuation's list of top words. Similarly, we also show results considering the list of top Comma words, top Question words, top Exclamation words, and top All[4] words, respectively (in order of top to bottom in Figure 11).

We run similar postprocessing approaches on the BiAttention-English model's output. Figure 12 shows the F1-scores for mini-batch 4 after the postprocessing. Similar to Figure 11, from the top left corner, we show results for each punctuation considering top 3, 5, and 7 Period words. We also show the results considering all four punctuation marks in the bottom left figure.

Corresponding improvements (i.e., increase in F1-scores) achieved after applying each postprocessing approaches over the employed BiAttention-Bangla and BiAttention-English models (ORIGINAL) are shown at the right of each chart in Figures 11 and 12. For example, in Figure 11, although we find no improvement for BiAttention-Bangla model using top Period words (top-left), improvements are found using top Comma words (top-right) for predicting Comma and Question marks. Besides, we find significant improvements for predicting Exclamation mark using top Exclamation words and top All (considering all punctuation marks' lists altogether) words. Overall, with top

[4]This approach considers top Period words, followed by top Comma, Question, and Exclamation words respectively to generate the final postprocessing output.

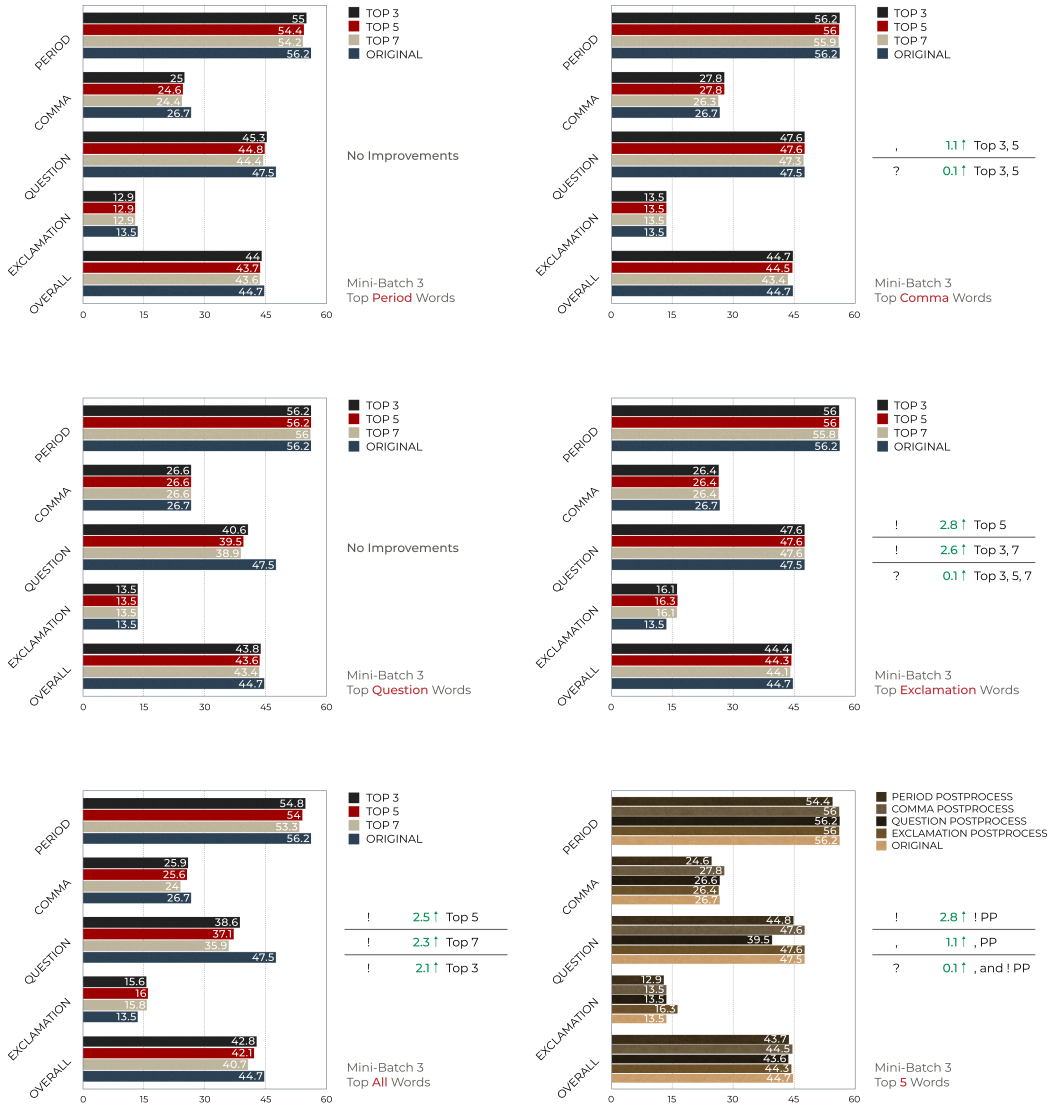

Fig. 11. F1-score for one mini-batch after applying postprocessing approach on BiAttention-Bangla model output. The top-left figure shows the results after postprocessing considering only top 3, 5, and 7 Period words. Similarly, top-right figure shows results for the Comma words. The bottom-left figure shows results considering top words from all four punctuation marks.

5 words for all cases, we achieve increments in F1-scores by 1.1 for Comma using top Comma words, 0.1 for Question using both top Comma and Exclamation words, and 2.8 for Exclamation using top Exclamation words. However, neither top Period nor top Question words could offer any improvement for the BiAttention-Bangla model. For the BiAttention-English model, we achieve a maximum increase of 0.4 on F1-score for the Comma mark with taking top 7 words from all punctuation and 0.1 on the Period mark with taking top 5 Question words. An increase of 0.5 is found for the Exclamation mark when taking top 7 Period words (top-left of Figure 12). Yet, we find no improvement on Question mark with any of the postprocessing approaches on this batch.

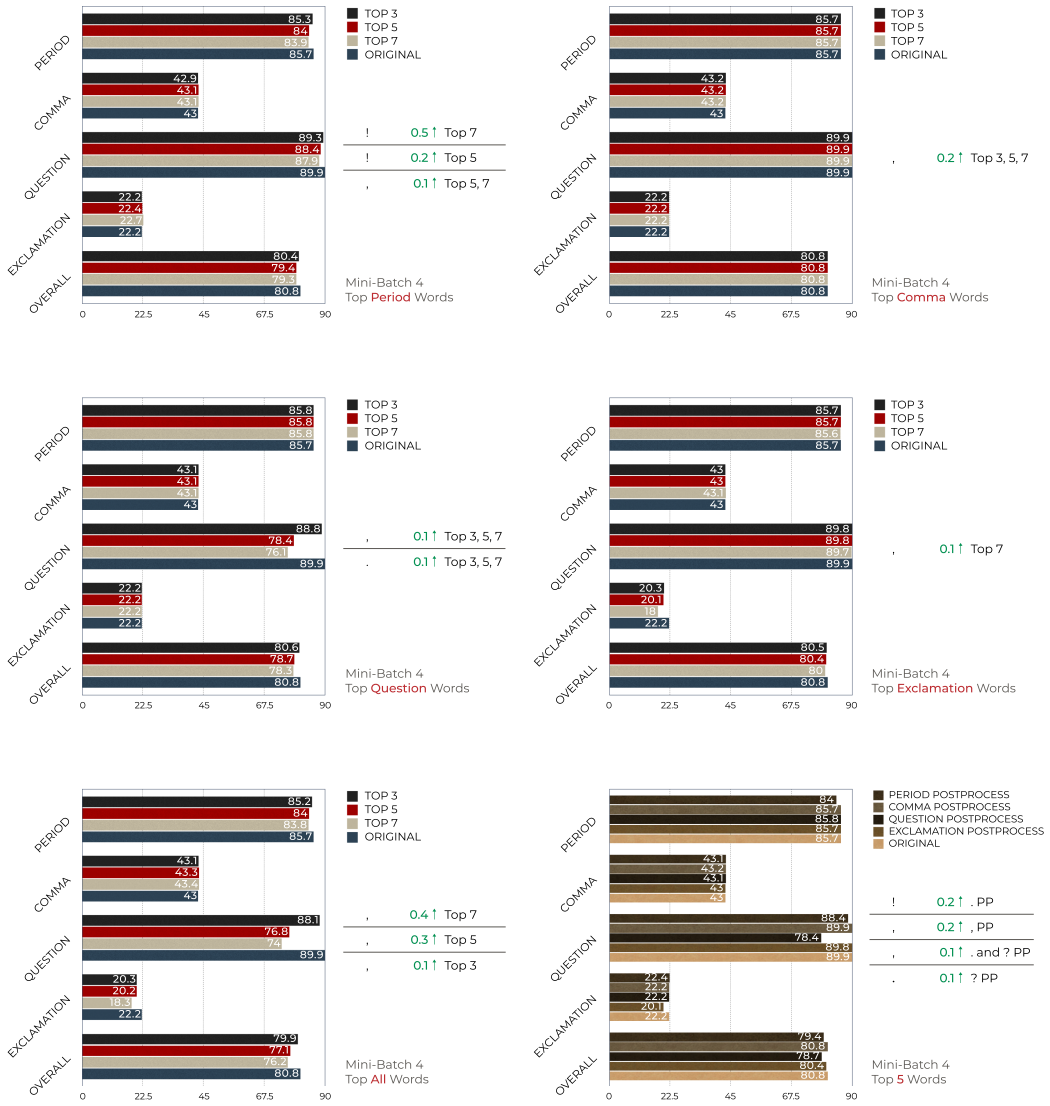

Fig. 12. F1-score for one mini-batch after applying postprocessing approach on BiAttention-English model output. The top-left figure shows the results after postprocessing with taking only top 3, 5, and 7 Period words. Similarly, the top-right figure shows results for the Comma words. The bottom-left figure shows results with taking top words from all four punctuation marks.

For all the mini-batches, the summary of the maximum improvements achieved after postprocessing is presented in Table 4 (for BiAttention-Bangla model) and Table 5 (for BiAttention-English model).

Hence, the insights from our result analysis help us to conclude that careful postprocessing using top words can improve the performance of a punctuation prediction model (particularly, BiAttention-Bangla). However, we need to be mindful about choosing the number of top punctuation words while postprocessing lest we increase the performance of a single punctuation mark and decrease performance for the rest. On the other hand, the postprocessing provides negligible

Table 4. Maximum Improvements in F1-score (in %) After Applying Postprocessing on BiAttention-Bangla Model Output

Mini-Batch	Period	Comma	Question	Exclamation
1	N/A	1.83%	N/A	N/A
2	0.18%	4.71%	N/A	14.95%
3	N/A	4.21%	0.21%	21.48%
4	N/A	1.44%	0.2%	7.89%
5	N/A	2.78%	N/A	7.14%

Table 5. Maximum Improvements in F1-score (in %) After Applying Postprocessing on BiAttention-English Model Output

Mini-Batch	Period	Comma	Question	Exclamation
1	N/A	0.60%	0.31%	N/A
2	N/A	0.23%	N/A	N/A
3	N/A	0.09%	0.10%	N/A
4	0.17%	0.93%	N/A	2.25%
5	N/A	0.08%	N/A	N/A

improvements for the BiAttention-English model, although there is a 2.25% of improvement for the Exclamation mark prediction, 0.93% for the Comma prediction, 0.17% for the Period prediction, and 0.31% for Question prediction (Table 5). In summary, for BiAttention-Bangla model, our postprocessing approaches improve the performance of Exclamation prediction significantly (up to 21.48%), while reasonable improvement (up to 4.71%) is achieved for Comma prediction along with slight improvements for Question and Period prediction. For the BiAttention-English model, the postprocessing approaches offer a maximum 2.25% improvements for the Exclamation, and small improvement for the Comma, Question, and Period prediction.

5.2 Balanced Dataset

Next, we present the results for Bangla-Balanced dataset. Importantly, there is no prior work on punctuation prediction in Bangla language, where an analysis is performed using a balanced dataset due to lack of such dataset in Bangla languauge. Here, we show the performance of our BiAttention-Balanced model (using Bangla-Balanced dataset). Figure 13 shows the results obtained directly from the model. The standout number that comes out is for the Question mark, as the model achieves $P = 66.7$, $R = 58.3$, $F1 = 62.2$ for predicting the Question marks, offering significant (30%) improvement over the BiAttention-Bangla model. The model also outperforms the BiAttention-Bangla model for Comma, with scores of $P = 64.9$, $R = 42.0$, $F1 = 51.0$. However, the BiAttention-Balanced model's scores for the Period mark ($P = 54.9$, $R = 42.4$, $F1 = 47.8$) are lower than that of the BiAttention-Bangla model. This is mainly due to the fact of having a relatively lower number of Period marks in the balanced dataset, when compared to that in the imbalanced OPUS datasets. Nevertheless, the most promising results are achieved for the Exclamation mark prediction by this model. Here, this balanced BiAttention model achieves scores almost twice the scores of the imbalanced models, with scores of $P = 49.5$, $R = 33.9$, $F1 = 40.2$. Since there is no

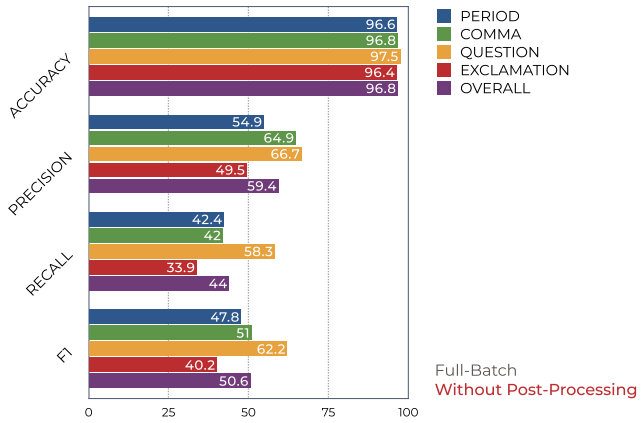

Fig. 13. A representation of individual and overall Accuracy, Precision, Recall, and F1-score (in %) achieved by our BiAttention-Balanced model. The model performs significantly well for Question ($F1 = 62.2$), Comma ($F1 = 51.0$), and Exclamation ($P = 49.5$ and $F1 = 40.2$).

Table 6. Results of BiAttention-Bangla and BiAttention-Balanced

Language Model	Evaluation Metrics	Comma	Period	Question	Exclamation	Overall
BiAttention-Bangla	Precision	43.1	65.5	51.2	24.1	54.6
	Recall	20.4	52.3	51.0	16.4	39.4
	F1-score	27.7	56.9	51.1	19.5	45.8
BiAttention-Balanced	Precision	64.9	54.9	66.7	49.5	59.4
	Recall	42	42.4	58.3	33.9	44.0
	F1-score	51.0	47.8	62.2	40.2	50.6

BiAttention-Balanced significantly outperforms BiAttention-Bangla for Comma, Question, Exclamation, and Overall (all punctuation marks) mainly due to its balanced nature.

study of the Exclamation mark prediction in Bangla text prior to this one, our work sets a new benchmark here, and will pave the way for further extensive study in this field of Bangla NLP. Furthermore, the BiAttention-Balanced model achieves overall scores of $P = 59.4$, $R = 44.0$, $F1 = 50.6$, i.e., plausibly higher scores than the BiAttention-Bangla model.

The bottom line of the performance of the model on the balanced dataset is that, due to higher and fairly distributed occurrences of the punctuation marks, the scores are considerably higher than that of the imbalanced dataset. Specifically, the balanced nature of the dataset makes the model perform better. Besides, the Prothom-Alo news transcript contains more (semantically) accurate sentences, which instigates better performance of the model [2]. To better understand the BiAttention-Balanced model's performance, we compile the results of both Bangla models in Table 6.

6 CONCLUSION AND FUTURE WORK

In the field of Bangla language processing, there is a lot to explore on punctuation prediction or restoration. In our work, we propose a system that efficiently predicts punctuation marks in Bangla text, and outputs reasonably correct punctuated texts. We provide a comparative analysis of the

employed model using three different datasets. In this course of work, we prepare two Bangla datasets and one English dataset for punctuation prediction, which are also made publicly available. Therefore, our work will facilitate future work in this research arena specifically for Bangla language, and ultimately unfold the doorway for further research and exploration on different NLP tasks of Bangla language. Moreover, we introduce different postprocessing techniques along with their results with two different languages. These techniques can also be applied for improving punctuation prediction in other languages' texts, which we leave as a potential future work of this study. It is worth mentioning, due to the lack of previous work of such kind with Bangla language, we face a lot of shortcomings in terms of necessary resources. Furthermore, the features of Bangla language are quite complex and unique in their own way, compared to other top languages such as English and German, which already have numerous notable related studies.

In the future, we plan to incorporate and explore more punctuation symbols such as Colon (:), Semicolon (;), and Hyphen (-). Moreover, a comparative analysis of various benchmark models (e.g., LSTM, Transformer, etc.) for prediction and restoration of punctuation in Bangla text is yet to be performed extensively. In addition, we plan to work with domain-specific datasets such as novels, medical reports, social media blogs on a particular domain, and so on, to ameliorate the quality of Bangla punctuation datasets.

REFERENCES

[1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th {USENIX} Conference on Operating Systems Design and Implementation (OSDI'16)*. USENIX Association, 265–283.

[2] Farzana Islam Adiba, Tahmina Islam, M. Shamim Kaiser, Mufti Mahmud, and Muhammad Arifur Rahman. 2020. Effect of corpora on classification of fake news using naive Bayes classifier. *International Journal of Automation, Artificial Intelligence and Machine Learning* 1, 1 (2020), 80–92.

[3] Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Punctuation restoration using transformer models for resource-rich and-poor languages. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT'20)*. 132–142.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[5] Miguel Ballesteros and Leo Wanner. 2016. A neural network architecture for multilingual punctuation generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing; 2016 Nov. 1–5; Austin (TX, USA).[place unknown]: ACL; 2016. p. 1048–53.* ACL (Association for Computational Linguistics).

[6] Doug Beeferman, Adam Berger, and John Lafferty. 1998. Cyberpunc: A lightweight punctuation annotation system for speech. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, Vol. 2. IEEE, 689–692.

[7] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[10] Mingfeng Fang, Haifeng Zhao, Xiao Song, Xin Wang, and Shilei Huang. 2019. Using bidirectional LSTM with BERT for Chinese punctuation prediction. In *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP'19)*. IEEE, 1–5.

[11] H. M. M. Hasan and M. A. Islam. 2020. Emotion recognition from Bengali speech using RNN modulation-based categorization. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT'20)*. IEEE, 1131–1136.

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[13] Md. Adnanul Islam, Md. Saidul Hoque Anik, and A. B. M. Alim Al Islam. 2021. Towards achieving a delicate blending between rule-based translator and neural machine translator. *Neural Computing and Applications* 33, 18 (2021), 12141–12167.

[14] Md. Adnanul Islam and A. B. M. Alim Al Islam. 2016. Polygot: Going beyond database driven and syntax-based translation. In *Proceedings of the 7th Annual Symposium on Computing for Development*. ACM, Article 28, 4 pages.
[15] Md. Adnanul Islam, A. B. M. Alim Al Islam, and Md. Saidul Hoque Anik. 2017. Polygot: An approach towards reliable translation by name identification and memory optimization using semantic analysis. In *4th International Conference on Networking, Systems and Security (NSysS'17)*. IEEE, 1–8.
[16] Md. Adnanul Islam, Md. Saddam Hossain Mukta, Patrick Olivier, and Md. Mahbubur Rahman. 2022. Comprehensive guidelines for emotion annotation. In *Proceedings of the 22nd ACM International Conference on Intelligent Virtual Agents* (Faro, Portugal) *(IVA'22)*. Association for Computing Machinery, Article 5, 8 pages. https://doi.org/10.1145/3514197.3549640
[17] Chin Char Juin, Richard Xiong Jun Wei, Luis Fernando D'Haro, and Rafael E. Banchs. 2017. Punctuation prediction using a bidirectional recurrent neural network with part-of-speech tagging. In *TENCON 2017-2017 IEEE Region 10 Conference*. IEEE, 1806–1811.
[18] 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'19)*. IEEE, 7280–7284.
[19] Xinxing Li and Edward Lin. 2020. A 43 language multilingual punctuation prediction neural network model. In *INTERSPEECH*. 1067–1071.
[20] Xin Liu, Yi Liu, and Xiao Song. 2018. Investigating for punctuation prediction in Chinese speech transcriptions. In *2018 International Conference on Asian Language Processing (IALP'18)*. IEEE, 74–78.
[21] 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'19)*. IEEE, 268–273.
[22] John Makhoul, Francis Kubala, Richard Schwartz, and Ralph Weischedel. 1999. Performance measures for information extraction. In *Proceedings of DARPA Broadcast News Workshop*. Herndon, VA, 249–252.
[23] Marcin Milkowski. 2010. Developing an open-source, rule-based proofreading tool. *Software: Practice and Experience* 40, 7 (2010), 543–566.
[24] Anna Moro and Gyorgy Szaszak. 2017. A phonological phrase sequence modelling approach for resource efficient and robust real-time punctuation recovery. In *INTERSPEECH*. 558–562.
[25] Md. Saddam Hossain Mukta, Md. Adnanul Islam, Faisal Ahamed Khan, Afjal Hossain, Shuvanon Razik, Shazzad Hossain, and Jalal Mahmud. 2021. A comprehensive guideline for Bengali sentiment annotation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 21, 2, Article 30 (Oct. 2021), 19 pages. https://doi.org/10.1145/3474363
[26] Alexandre Nanchen and Philip N. Garner. 2019. Empirical evaluation and combination of punctuation prediction models applied to broadcast news. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19)*. IEEE, 7275–7279.
[27] Alp Oktem, Mireia Farrus, and Leo Wanner. 2017. Attentional parallel RNNs for generating punctuation in transcribed speech. In *International Conference on Statistical Language and Speech Processing*. Springer, 131–142.
[28] Tuba Parlar, Selma Ozel, and Fei Song. 2019. Analysis of data pre-processing methods for sentiment analysis of reviews. *Computer Science* 20 (2019).
[29] Stephan Peitz, Markus Freitag, Arne Mauser, and Hermann Ney. 2011. Modeling punctuation prediction as machine translation. In *Proceedings of the 8th International Workshop on Spoken Language Translation: Papers*.
[30] Quang H. Pham, Binh T. Nguyen, and Nguyen Viet Cuong. 2019. Punctuation prediction for Vietnamese texts using conditional random fields. In *Proceedings of the Tenth International Symposium on Information and Communication Technology*. 322–327.
[31] Pau Rodriguez, Miguel Angel Bautista, Jordi Gonzalez, and Sergio Escalera. 2018. Beyond one-hot encoding: Lower dimensional target embedding. *CoRR* abs/1806.10805 (2018). arXiv:1806.10805. http://arxiv.org/abs/1806.10805.
[32] Wael Salloum, Gregory Finley, Erik Edwards, Mark Miller, and David Suendermann-Oeft. 2017. Deep learning for punctuation restoration in medical reports. In *BioNLP 2017*. 159–164.
[33] Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.
[34] Yuanhang Su and C.-C. Jay Kuo. 2019. On extended long short-term memory and dependent bidirectional recurrent neural network. *Neurocomputing* 356 (2019), 151–161.
[35] Gyorgy Szaszak and Mata Akos Tundik. 2019. Leveraging a character, word and prosody triplet for an ASR error robust and agglutination friendly punctuation approach. *Proc. Interspeech 2019* (2019), 2988–2992.
[36] Jorg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Lrec*, Vol. 2012. 2214–2218.
[37] Ottokar Tilk and Tanel AlumÃďe. 2016. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *Interspeech 2016*. 3047–3051. https://doi.org/10.21437/Interspeech.2016-1517
[38] Mate Akos Tundik and Gyorgy Szaszak. 2018. Joint word-and character-level embedding CNN-RNN models for punctuation restoration. In *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom'18)*. IEEE, 000135–000140.

[39] Andris Varavs and Askars Salimbajevs. 2018. Restoring punctuation and capitalization using transformer models. In *International Conference on Statistical Language and Speech Processing*. Springer, 91–102.
[40] Feng Wang, Wei Chen, Zhen Yang, and Bo Xu. 2018. Self-attention based network for punctuation restoration. In *2018 24th International Conference on Pattern Recognition (ICPR'18)*. IEEE, 2803–2808.
[41] Tian Wang and Kyunghyun Cho. 2015. Larger-context language modelling. *arXiv preprint arXiv:1511.03729* (2015).
[42] K. Xu, L. Xie, and K. Yao. 2016. Investigating LSTM for punctuation prediction. In *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP'16)*. 1–5. https://doi.org/10.1109/ISCSLP.2016.7918492
[43] 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'19)*. IEEE, 7270–7274.
[44] Zabir Al Nazi. 2020. Bangla Newspaper Dataset. https://doi.org/10.34740/KAGGLE/DSV/1576225
[45] Piotr Zelasko, Piotr Szymanski, Jan Mizgajski, Adrian Szymczak, Yishay Carmiel, and Najim Dehak. 2018. Punctuation prediction model for conversational speech. *arXiv preprint arXiv:1807.00543* (2018).

Received 7 August 2021; revised 27 May 2022; accepted 3 October 2022