Punctuation Prediction in Bengali Text

Md. Rafi
Department of CSE
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
190104041@aust.edu

Jerin Ahasan Kheya
Department of CSE
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
190104043@aust.edu

Asif Mamun Hridoy
Department of CSE
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
190104047@aust.edu

Syeda Annan Asrafi
Department of CSE
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
190104050@aust.edu

Abstract—Punctuation prediction is important because it can significantly improve the readability of speeches or writings that have been automatically transcribed by adding the proper punctuation. Additionally, unpunctuated texts produced by systems like Automatic Speech Recognizer (ASR) make it difficult for people to understand them and impair the effectiveness of numerous natural language processing (NLP) activities. These NLP-related tasks have been well studied for English, but relatively little work has been done for punctuation prediction in Bangla. This study aims to bridge this gap by proposing an effective method for accurately predicting punctuation in Bengali texts. In pursuit of this objective, we used a publicly accessible Bengali newspaper dataset to train both a Long Short-Term Memory (LSTM) network and a pre-trained transformer model known as Bangla-BERT. Our experimentation revealed that Bangla-BERT outperforms the LSTM network. This study contributes to the advancement of punctuation prediction techniques in the Bengali language.

Index Terms—punctuation prediction, Bengali text, LSTM, Bangla-BERT

I. Introduction

One of the main issues with artificial intelligence is Natural Language Processing (NLP), which enables communication between natural human languages and technological equipment. Parts-of-speech (PoS) tagging, machine translation, sentiment analysis, and other practical applications have all been made possible by NLP [1]–[3]. Due to the difficult nature of human-understandable words joining together to create specific meanings, natural language processing is exceedingly difficult for a machine to learn [4], [5]. But as a result of diligent study in this area, more and more systems are incorporating NLP-based programs that have improved the coherence of our daily interactions with machines.

Punctuation marks serve a crucial function in directing where one should halt, pause, and express emotions in order to understand the content of a sentence. When unpunctuated texts are produced by systems like Automatic Speech Recognition (ASR), adding punctuation makes the text easier to read and can help with further processing tasks like machine translation [6], sentiment analysis [7] and different NLP fields [6]. Additionally, the systems that need translations may have a significant impact on the creation of accurate machine translations.

Numerous research have dealt with the prediction and restoration of punctuation (e.g., [8]–[11]) but little work has been done previously for Bangla. Furthermore, the quantity of publicly accessible datasets that contain accurate Bangla punctuation is extremely small. For the purpose of predicting punctuation in Bangla text, we will be using recurrent neural networks. For now this is our goal for this study.

II. Literature Review

Makhija et al. [14] proposed an architecture for punctuation prediction that uses pre-trained BERT embeddings. Their model was trained and tested on IWSLT2012 dataset [15] which is an English ASR dataset. Their model achieved an overall F1 of 81.4% on the joint prediction of period, comma and question mark.

Rahman et al. [12] trained a bidirectional recurrent neural network (BRNN) along with Attention model to predict punctuation on Bengali text. They achieved highest F-1 score of 62.2% for question mark prediction on a balanced dataset.

Alam et al. [16] trained different transformer models for Bengali punctuation prediction. Their best performing model was XLM-RoBERTa-large with an overall F-1 score of 87.0%.

III. Dataset Description

Correctly punctuated sentences from books, newspapers, magazines etc. can be used as our dataset. Rather than collecting sentences from various sources to create our own dataset, we have used existing Bangla newspaper dataset [13].

TABLE I

Punctuation	Label	Split	Sentence
Period	0	Train	10,834
		Validation	2,709
		Test	3,385
Exclamatory	1	Train	10,501
		Validation	2,625
		Test	3,282
Question	2	Train	10,665
		Validation	2,665
		Test	3,333

The dataset [13] contains three different subsets. One of the dataset (Bangla OPUS dataset) is created by taking data from various sources like Bengali articles, conversations, translations etc. and another dataset is collected from top Bangladeshi newspaper Prothom Alo's archive. Both of the dataset were imbalanced i.e. number of different punctuation in the dataset were not same. Rahman et al. [12] took a portion of the Prothom Alo Dataset and made the dataset balanced. This study is based on the balanced Prothom Alo dataset. The actual number of data after pre-processing and their distribution in train and validation set can be found in Table I.

A. Data Pre-processing

Some data prepossessing techniques were applied to the dataset. At first, the dataset were separated into articles by articles. We first separate the dataset into sentences. We removed duplicate sentences, unnecessary white spaces, and numbers.

To prepare the final dataset, we take each sentences as individual row and label them according to the last punctuation of the sentence. A complete sentence can end with three punctuation- the period (।), the question mark (?), and the exclamation point (!). We labeled a sentence as 0 if the sentence is ended with a period, 1 if the sentence is ended with a exclamation mark and 2 if the sentence is ended with a question mark. After labeling all sentences we delete all the punctuation from the original sentences. For example the sentence "আপনারা কী বলেন, এটা ভালো গুণ নয়?" will be processed as "আপনারা কী বলেন এটা ভালো গুণ নয়" and the label of this sentence will be 2.

From the sentence length distribution of the dataset (see figure 2), it can be seen that most of the sentences has length greater than or equal to 50. Therefore, sentences with length grater than 50 are excluded from the dataset.

B. Exploratory Data Analysis

After removing large sentences the balanced Prothom Alo dataset contains total 448,285 sentences. The dataset contains an equal number of periods, commas, question marks and exclamation marks. The total number of each punctuation is roughly 149428. An overview of the dataset can be found in figure 1.

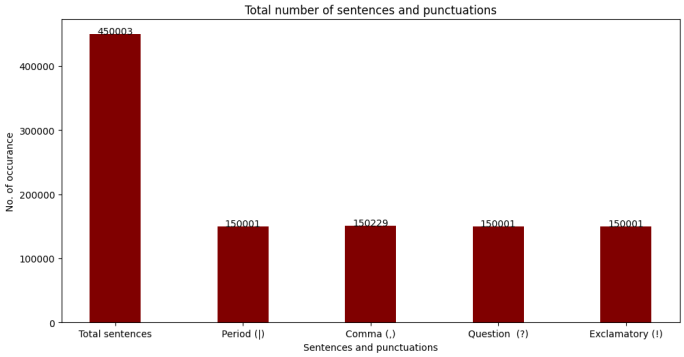

Fig. 1. Overview of Balanced Prothom Alo Dataset

From the sentence length distribution of the dataset can be found in figure 2. By analyzing the distribution, it can be seen that most sentences are in length five to fifteen words in range. It is also seen that very few sentences has lengths more than thirty words. Sentences of length forty or more than forty words are very rare.

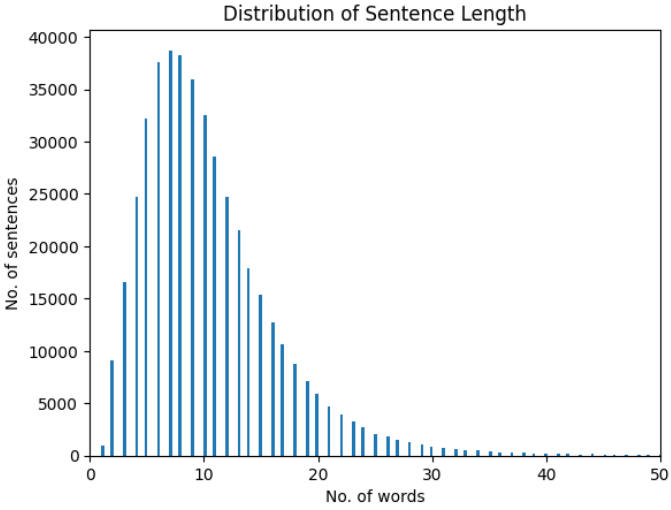

Fig. 2. Lengths of sentences in Prothom Alo Dataset

The mean, median, and standard deviation of sentence lengths before pre-processing were 9.58, 8.0, 11.75 respectively. After pre-processing the mean, median, and standard deviation of sentence lengths is 9.09, 8.0, 5.69 respectively. Figure 3 and figure 4 shows the data distribution with respect to sentence length before and after pre-processing respectively.

After analyzing the dataset the most used ten words with their number of occurrence in the dataset is shown in 5.

C. Feature Extraction

For this study two different approach is followed- LSTM and Bangla-BERT [17]. For LSTM network, the input text is tokenized, breaking it into individual words or

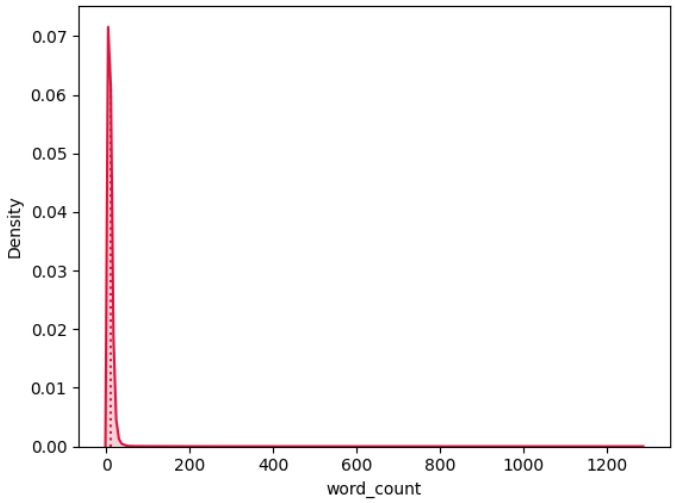

Fig. 3. Data distribution before pre-processing

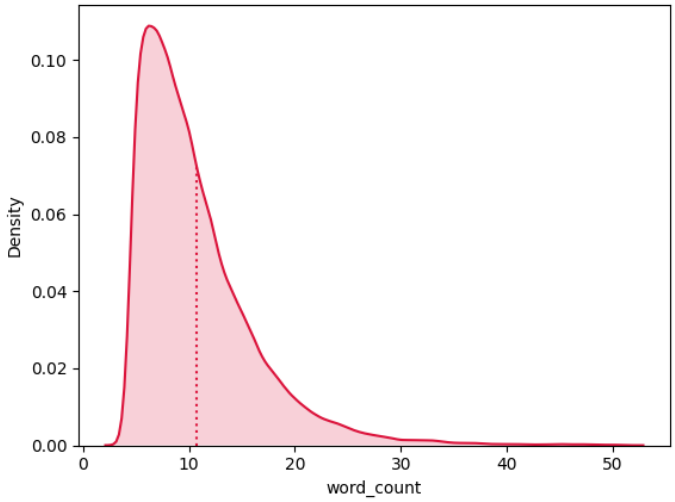

Fig. 4. Data distribution after pre-processing

subwords. Each token is represented as a unique integer based on a token vocabulary. LSTMs require inputs of the same length. Since sentences can vary in length, we padded the token sequences with special padding tokens to ensure uniform length across all inputs. After that, in order to represent tokens as continuous-valued vectors that capture semantic meaning, we used pre-trained word embeddings- GloVe from Stanford. The input token sequences, represented as word embeddings, are then fed into the LSTM model.

For the second approach we utilized the pre-trained Bangla-BERT model for our feature extraction process. Before feature extraction, the input text is tokenized into subwords or words by pre-tained 'sagorsarker/bangla-bert-base' tokenizer. Each token is then associated with an embedding vector. Additionally, a special [CLS] token is inserted at the beginning of the input and [SEP] token is

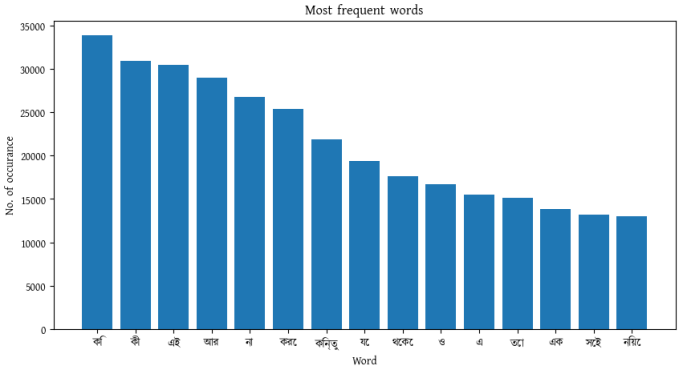

Fig. 5. Most Used Words of Prothom Alo Dataset

inserted at the end of the input to represent the entire sequence. These tokens are the input for Bangla-BERT transformer model.

IV. Methodology

A. LSTM

The key steps of proposed methodology for LSTM model is shown in figure 6. Short description of these steps are given here:

Input: Here, inputs are the sentences from the dataset. The raw sentences are considered as input.

Pre-processing: Each input sentences go through pre-processing step. Pre-processing steps are described in 'Dataset Description' section.

Tokenization: After pre-processing, each sentence is tokenized using a tokenizer from Keras. The length of each sentences is 50 words. Therefore, each sentence is represented with 50 tokens. Tokens are the numerical representation of words. If a sentence is less than 50 words then padding is added to the sentence to make all the sentences same length.

Word Embedding: Word embedding represents words in a vector space in a way that words with similar meaning can be closer. A popular method for word embedding is GloVe. In this step, GloVe embedding is implemented.

LSTM: Finally the embedded sequences are passed to the LSTM model. Finally, in the training phase, loss is calculated and parameters are updated. In the testing phase, prediction for the test data is calculated and model's performance is measured.

Hyper parameters used for LSTM model in this study are: batch size = 8, number of epochs = 5.

B. Bangla-BERT

The key steps of proposed methodology for Bangla-BERT is shown in figure 7. Short description of these steps are given here:

Input and Pre-processing: Input are the same as the inputs for LSTM model and sentences goes through the same pre-processing steps.

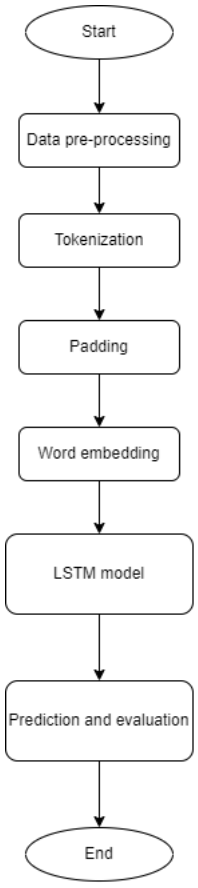

Fig. 6. Flow diagram of LSTM model

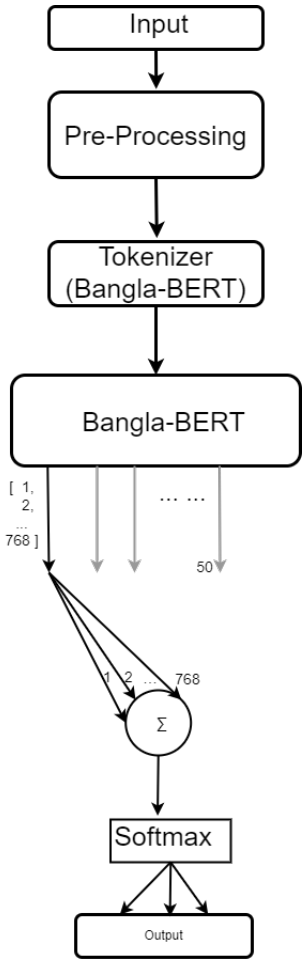

Fig. 7. Flow diagram of Bangla-BERT model

Tokenization: After pre-processing, each document is tokenized using Bangla-BERT tokenizer. Like the previous LSTM step, length of each sentences is 50. The BERT tokenizer adds [CLS] and [SEP] tokens at the start and end of of each documents respectively. .

Bangla-BERT Model: In this step the sequence of tokens is provided as the input for transformer model. We have used Bangla-BERT transformer model. The output of this step is a vector for each input tokens. The size of the vector is made with 768 float numbers.

Logistic Regression: From the previous output, only the first vector i.e. the output vector for [CLS] token is the input vector for the logistic regression. All other output vectors are left out. Softmax is used in output to as this is a multi-class classification problem.

Hyper parameters used for both BERT and RoBERTa models in this study are: batch size = 16, learning rate = 1e-5, number of epochs = 2.

V. Result Analysis

Confusion matrix for Bangla-BERT model is shown in figure 8.

Training Loss and F-1 score vs. Epoch Curve is shown in Figure 9.

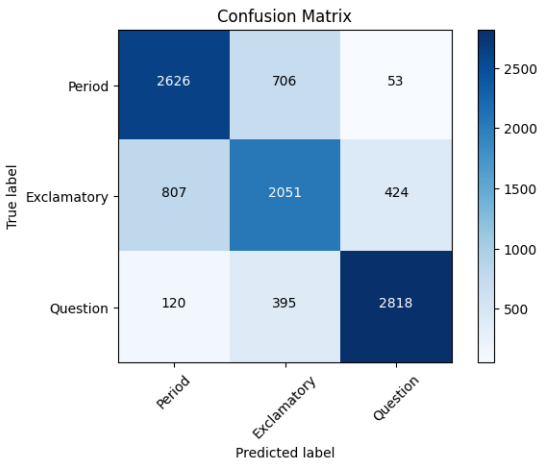

Fig. 8. Confusion matrix for Bangla-BERT

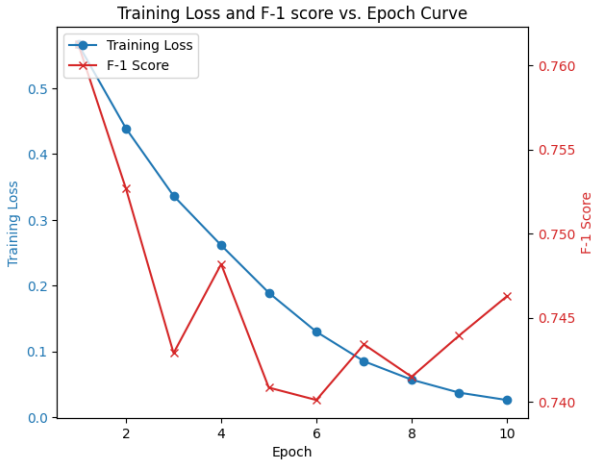

Fig. 9. Training Loss and F-1 score vs. Epoch Curve

TABLE II
Performance matrices for Bangla-BERT

Class	Accuracy	Precision	Recall	F-1 Score
Period	0.8486	0.7505	0.8200	0.7837
Exclamatory	0.7988	0.7261	0.6363	0.6782
Question	0.9087	0.8516	0.8780	0.8646
Overall	0.8528	0.7761	0.7781	0.7755

Table II shows the different performance score for all three punctuation classes along with overall performance. As we can see from the table, all the performance matrices for question class has better values. The reason is that the difference between a question and a statement or exclamatory sentence is higher than the difference between other classes.

The overall accuracy for LSTM model was 0.615 which is quite low than Bangla-BERT.

VI. Conclusion and Future Work

A typical post-processing issue with Automatic Speech Recognition (ASR) systems is punctuation restoration. It is necessary to make the transcribed text easier to read. In this paper, we divided our whole dataset into three different class labels to predict three separate punctuation. Our proposal concludes by using the pre-trained language representation model Bangla-BERT and also LSTM model. We fine tuned our models for our task and with our dataset we achieved satisfactory results as we didn't use various types of data. In our conclusion it can be said that Bangla-BERT outperformed LSTM. Till now our work can only predict three punctuation at the end of a Bangla sentence which are period, exclamatory and question mark. One of the limitations of this work that it can not predict comma or any other punctuation which are situated in the middle of a sentence. This work can be extended in future by adding this feature as well. Here only newspaper dataset is used. Dataset can be extended by adding various other types of dataset like using books, ASR etc. Also dataset can be increased as well. There are many opportunities to explore other models to improve performance.

References

[1] Hasan, H. M., & Islam, M. A. (2020, August). Emotion recognition from bengali speech using rnn modulation-based categorization. In 2020 third international conference on smart systems and inventive technology (ICSSIT) (pp. 1131-1136). IEEE.

[2] Islam, M. A., Anik, M. S. H., & Islam, A. A. A. (2021). Towards achieving a delicate blending between rule-based translator and neural machine translator. Neural Computing and Applications, 33, 12141-12167.

[3] Mukta, M. S. H., Islam, M. A., Khan, F. A., Hossain, A., Razik, S., Hossain, S., & Mahmud, J. (2021). A Comprehensive Guideline for Bengali Sentiment Annotation. Transactions on Asian and Low-Resource Language Information Processing, 21(2), 1-19.

[4] Islam, M. A., & Islam, A. A. A. (2016, November). Polygot: Going beyond database driven and syntax-based translation. In Proceedings of the 7th Annual Symposium on Computing for Development (pp. 1-4).

[5] Islam, M. A., Al Islam, A. A., & Anik, M. S. H. (2017, December). Polygot: An approach towards reliable translation by name identification and memory optimization using semantic analysis. In 2017 4th International Conference on Networking, Systems and Security (NSysS) (pp. 1-8). IEEE.

[6] Peitz, S., Freitag, M., Mauser, A., & Ney, H. (2011). Modeling punctuation prediction as machine translation. In Proceedings of the 8th International Workshop on Spoken Language Translation: Papers (pp. 238-245).

[7] Parlar, T., Ozel, S., & Song, F. (2019). Analysis of data preprocessing methods for sentiment analysis of reviews. Computer Science, 20.

[8] Ballesteros, M., & Wanner, L. (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).

[9] Fang, M., Zhao, H., Song, X., Wang, X., & Huang, S. (2019, December). Using bidirectional LSTM with BERT for Chinese punctuation prediction. In 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP) (pp. 1-5). IEEE.

[10] Szaszák, G., & Tündik, M. A. (2019). Leveraging a Character, Word and Prosody Triplet for an ASR Error Robust and Agglutination Friendly Punctuation Approach. In INTERSPEECH (pp. 2988-2992).

[11] Tündik, M. Á., & Szaszák, G. (2018, August). Joint word-and character-level embedding CNN-RNN models for punctuation restoration. In 2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom) (pp. 000135-000140). IEEE.

[12] Rahman, H., Rahin, M. R. S., Mahbub, A. M., Islam, M. A., Mukta, M. S. H., & Rahman, M. M. (2023). Punctuation Prediction in Bangla Text. ACM Transactions on Asian and Low-Resource Language Information Processing, 22(3), 1-20.

[13] Rahin, M.R.S. (2021). Bangla Text Dataset, Version 1. Retrieved June 21, 2023 from https://www.kaggle.com/datasets/rezwanrahin/bangla-text-dataset.

[14] Makhija, K., Ho, T. N., & Chng, E. S. (2019, November). Transfer learning for punctuation prediction. In 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (pp. 268-273). IEEE.

[15] Federico, M., Cettolo, M., Bentivogli, L., Michael, P., & Sebastian, S. (2012). Overview of the IWSLT 2012 evaluation campaign. In Proceedings of the international workshop on spoken language translation (IWSLT) (pp. 12-33).

[16] Alam, T., Khan, A., & Alam, F. (2020, November). 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) (pp. 132-142).
[17] Sarker S. (2020). BanglaBERT: Bengali Mask Language Model for Bengali Language Understanding. https://github.com/sagorbrur/bangla-bert