


ARTICLE

Part-of-speech tagger for Bodo language using deep learning approach

Dhrubajyoti Pathak , Sanjib Narzary, Sukumar Nandi and Bidisha Som

Centre for Linguistic Science and Technology, IIT Guwahati, Guwahati, Assam, India

Corresponding author: Dhrubajyoti Pathak; Email: drbj153@iitg.ac.in

(Received 29 November 2022; revised 8 November 2023; accepted 4 January 2024)

Special Issue on 'Natural Language Processing Applications for Low-Resource Languages'

Abstract

Language processing systems such as part-of-speech (POS) tagging, named entity recognition, machine translation, speech recognition, and language modeling have been well-studied in high-resource languages. Nevertheless, research on these systems for several low-resource languages, including Bodo, Mizo, Nagamese, and others, is either yet to commence or is in its nascent stages. Language model (LM) plays a vital role in the downstream tasks of modern natural language processing. Extensive studies are carried out on LMs for high-resource languages. However, these low-resource languages are still underrepresented. In this study, we first present BodoBERT, an LM for the Bodo language. To the best of our knowledge, this work is the first such effort to develop an LM for Bodo. Second, we present an ensemble deep learning-based POS tagging model for Bodo. The POS tagging model is based on combinations of BiLSTM with conditional random field and stacked embedding of BodoBERT with BytePairEmbeddings. We cover several LMs in the experiment to see how well they work in POS tagging tasks. The best-performing model achieves an F1 score of 0.8041. A comparative experiment was also conducted on Assamese POS taggers, considering that the language is spoken in the same region as Bodo.

Keywords: part-of-speech tagging; Bodo language; Bodo BERT; Bodo POS tagger; low-resource language

1. Introduction

Part-of-speech (POS) tagging is one of the building blocks of modern natural language processing (NLP). POS tagging automatically assigns grammatical class types to words in a sentence, such as noun, pronoun, verb, adjective, adverb, conjunction, and punctuation. Various approaches are there for automatic POS tagging; however, the deep neural network approach has achieved state-of-the-art accuracy for resource-rich languages. POS tagging plays a vital role in various text processing tasks, such as named entity recognition (NER) (Aguilar *et al.* 2019), machine translation (Niehues and Cho 2017), information extraction (IE) (Bhutani *et al.* 2019), question answering (Le-Hong and Bui 2018), and constituency parsing (Shen *et al.* 2018). Therefore, a well-performing POS tagger is inevitable in developing a successful NLP system for a language.

Bodo (Boro) is a Tibeto-Burman^a morphologically rich language, mainly spoken in Assam, a state in northeastern India. The Bodoland Territorial Region, an independent entity in Assam,

^aThe Sino-Tibetan language family includes the Tibeto-Burman languages. Tibeto-Burman language does not include Chinese languages.

uses it as its official language. The Devanagari script is used to write Bodo. As per the 2011 Indian census, Bodo has about 1.5 million speakers, making it the 20th most spoken language in India among the 22 scheduled languages. Even though most Bodo speakers are ethnic, Bodos, Assamese, Rabha, Koch Rajbongshi, Santhali, Garo, and the Muslim community in the Bodoland Territorial Region also speak the language.

However, the state of the NLP systems for Bodo is in a very nascent stage. The study of fundamental NLP tasks in Bodo language, such as lemmatization, dependency parser, language Identification, language modeling, POS tagging, and NER has yet to start. While word embedding or language models (LMs) play a crucial role in a deep learning approach, we observe that no existing pretrained language models cover the Bodo language. As a result, we could not find any downstream NLP tools developed using a deep learning method.

With 1.5 million Bodo speakers, the need for developing NLP resources for the Bodo language is highly demanding. Motivated by this gap in research of the NLP areas of the Bodo language, we present the first LM for Bodo language—BodoBERT based on BERT architecture (Devlin *et al.* 2018). We prepare a monolingual training corpus to train the LM by collecting and acquiring corpus from various sources. After that, we develop a POS tagger for Bodo language by employing the BodoBERT.

We explore different sequence labeling architectures to get the best POS tagging model. We conduct three experiments to train the POS tagging models: (a) fine-tuning based, (b) conditional random field (CRF) based, and (c) long-short term memory (BiLSTM)-CRF based. Among the three, the BiLSTM-CRF-based POS tagging model performs better than the others.

Bodo language uses the Devanagari script, which the Hindi language also uses. Therefore, we conducted the POS tagging experiment with LMs that were trained in Hindi to compare the performance with the BodoBERT. We cover LMs such as Fasttext, Byte Pair Embeddings (BPE), Contextualise Character Embedding (FlairEmbedding), MuRIL, XLM-R, and IndicBERT. We employ two different methods to embed the words in the sentence to train POS tagging models using BiLSTM-CRF architecture—Individual and Stacked methods. In the individual method, the model trained using BodoBERT achieves the highest F1 score of 0.7949. After that, we experiment with the stacked method to combine the performance of the BodoBERT with other LMs. The highest F1 score in the stacked method reached 0.8041. We believe this is not only the first neural Network-based POS tagging model for the language but also the first POS tagger in Bodo. Our contributions can be summarized as follows:

- Proposed a LM for Bodo language based on the BERT framework. To the best of our knowledge, this is the first LM for Bodo.
- Presented comparison of different POS tagging models for Bodo by employing state-of-the-art sequence tagger frameworks such as CRF, fine-tuned LMs, and BiLSTM-CRF.
- Comparison of POS tagging performance of different LMs in POS tagging models such as Fasttext, BPE, XLM-R, FlairEmbedding, IndicBERT, and MuRIL embedding using Individual and Stacked embedding methods.
- The top-performing Bodo POS tagging model and BodoBERT are made publicly available.^b

This paper is organized as follows- Section 2 describes related works on POS tagging for similar language. The details about the pretraining of BodoBERT and Bodo corpus used in training are presented in Section 3. Section 4 presents the experiments carried out to develop the neural network POS tagger. The section also includes the description of the annotated dataset and POS

^b<https://anonymous.4open.science/status/BodoPoS-4C10>

tagset in the experiment. In Section 5, we present all the experiment results of all three models using different sequence tagging architectures. Finally, we conclude our paper in Section 6.

2. Related work

This section presents related works about POS tagging in different languages. In our literature study, we do not find any prior work on language modeling for Bodo language. Furthermore, to the best of our knowledge, there is no previous work on POS tagging for Bodo language. Therefore, we discuss recent research works reported on neural network-based POS tagging in various low-resource languages belonging to the northeastern part of India.

In paper (Pathak *et al.* 2022), an Assamese POS tagger based on deep learning (DL) approach is proposed. Various DL architectures are used to develop the POS model, which includes CRF, bidirectional long short-term memory (BiLSTM) with CRF, and gated recurrent unit (GRU) with CRF. The BiLSTM-CRF model achieved the highest tagging F1 score of 86.52. Pathak *et al.* (2023) presented an ensemble POS tagger for Assamese, where two DL-based taggers and a rule-based tagger are combined. The ensemble model achieved an F1 score of 0.925 in POS tagging. Warjri *et al.* (2021) presented a POS-tagged corpus for Khasi language. The paper also presented a DL-based POS tagger for the language using different architectures, including BiLSTM, CRF, and character-based embedding with BiLSTM. The top performing tagger achieved an accuracy of 96.98% using BiLSTM with CRF technique. Alam *et al.* (2016) proposed a neural network-based POS tagging system for Bangla using BiLSTM and CRF architecture. They also used a pretrained word embedding model during training. The model achieved an accuracy of 86.0%. In paper (Pandey *et al.* 2022), reported work on POS tagging of the Mizo language. They employed classical LSTM and quantum-enhanced long short-term memory (QLSTM) in training and reported a tagging accuracy of 81.86% using the LSTM network. Kabir *et al.* (2016) presented their research on POS tagger for Bengali language using a DL-based approach. The tagger achieves an accuracy of 93.33% using deep belief network (DBN) architecture. There are other works that have been reported over the years on POS tagging for a variety of languages, including Assamese, Mizo, Manipuri, Khasi, and Bengali. However, they are based on traditional methods such as rule-based and HMM-based models.

3. BodoBERT

The transformer-based LM BERT achieves state-of-the-art performance on many NLP tasks. However, the success of BERT models on downstream NLP tasks is mostly limited to high-resource languages such as English, Chinese, Spanish, and Arabic. The BERT model is trained in English (Devlin *et al.* 2018). After that, the multilingual pretrained models were released for 104 languages. However, Bodo language is not covered. We could not find any pretrained LMs that cover the Bodo language. So, this motivates us to develop a LM for Bodo using the vanilla BERT (Devlin *et al.* 2018) architecture.

3.1. Bodo raw corpus

A large monolingual corpus is required to train a well-performed LM like BERT. Moreover, training BERT is a computationally intensive task, and it requires substantial hardware resources. On the other hand, getting decent monolingual raw corpus for Bodo has been an enduring problem for NLP research community. Although Bodo has a rich oral literary tradition, however, there was no standard writing script for writing until the year 2003. After a long history of the Bodo script movement, the Bodo language is recognized as a scheduled language of India by the Government of India, and the Devanagari script for writing is officially adopted.

We have curated the corpus after acquiring it from the Linguistic Data Consortium for Indian Languages (LDC-IL) (Ramamoorthy *et al.* 2019; Choudhary 2021). The text of the raw corpus is from different domains such as Aesthetics (Culture, Cinema, Literature, Biographies, and Folklore), Commerce, Mass media (Classified, Discussion, Editorial, Sports, General news, Health, Weather, and Social), Science and Technology (Agriculture, Environmental Science, Textbook, Astrology, Mechanical Engineering, and Environmental Science) and Social Sciences (Economics, Education, Political Science, Linguistics, Health and Family Welfare, History, Text Book, Law, etc). We also acquired another corpus from the work (Narzary *et al.* 2022). The final consolidated corpus has 1.6 million tokens and 191k sentences.

3.2. BodoBERT model training

The architecture of the BodoBERT model is based on a multilayer bidirectional transformer framework (Vaswani *et al.* 2017). We use the BERT framework (Devlin *et al.* 2018) to train the LM using the masked LM and next-sentence prediction tasks. WordPiece tokenizer (Wu *et al.* 2016) is used for embeddings with 50,000 vocabularies. The BERT model architecture is described in the guide “The Annotated Transformer,”^c and the implementation details are provided in “Tensorflow code and BERT model”.^d

The model is trained with six layers of transformers block with a hidden layer size of 768 and the number of self-attention heads as 6. There are approximately 103 M parameters. The model was trained on Nvidia Tesla P100 GPU (3584 Cuda Cores) for 300K steps with a maximum sequence length of 128 and batch size of 64. We used the Adam optimizer (Kingma and Ba 2014) with a warm-up of the first 3000 steps to a peak learning rate of 2e-5. The pretraining took approximately seven days to complete.

4. Bodo POS tagger

Part-of-speech tagging belongs to the sequence labeling tasks. There are different stages in developing a DL-based POS tagger. This section presents various stages of development of the Bodo POS tagger.

4.1. Annotated dataset

A DL-based method requires a large size of properly annotated corpus to train a POS tagging model. The performance of a well-performed tagging model depends upon the quality of the annotated dataset. On the other hand, the availability of annotated corpus in the public domain is very rare. Moreover, it is also a tedious and time-consuming task to annotate a corpus manually. Therefore, it is a challenging task to build a deep learning model for a low-resource language. In our literature study, we could find only one Bodo annotated corpus—Bodo Monolingual Text Corpus (ILCI-II, 2020b). The corpus is tagged by language experts manually as part of the project Indian Languages Corpora Initiative Phase-II (ILCI Phase- II), initiated by the Ministry of Electronics and Information Technology (MeitY), Government of India, and Jawaharlal Nehru University, New Delhi. The corpus contains approximately 30k sentences and 240k tokens comprised of different domains. The statistics of the annotated dataset are reported in Table 1. The corpus is tagged according to the Bureau of Indian Standards (BIS) tagset. We use this dataset for all our experiments to train neural-based taggers for Bodo.

We prepare the tagged dataset in CoNLL-2003 (Sang and De Meulder 2003) column format in which each line contains one word, and the corresponding POS tag is separated by tab space. An

^c<http://nlp.seas.harvard.edu/2018/04/03/attention.html>

^d<https://github.com/google-research/bert>

Table 1. Statistics of Bodo POS annotated dataset

| File name | Sentence count | Token count |
|--------------|----------------|-------------|
| Training set | 24003 | 192k |
| Dev set | 2325 | 23k |
| Test set | 3161 | 23k |

Table 2. Dataset format

| Word | Tag |
|------------|---------|
| बियो | PR_PRP |
| 88 | QT_QTC |
| सानावनो | N_NST |
| सानखौ | N_NN |
| खेबसे | QT_QTC |
| गिदिखनो | V_VM |
| (| RD_PUNC |
| Revolution | RD_RDF |
|) | RD_PUNC |
| | RD_PUNC |

empty line represents the sentence boundary in the dataset. In Table 2, the sample column format is shown.

For training, we first randomized the dataset; after that, the dataset was divided into 80:10:10 train, development, and test sets, respectively.

4.2. POS taggset

The TDIL dataset is tagged according to the Bureau of Indian Standards (BIS) tagset, which is considered the national standard for annotating Indian languages. The dataset contains eleven (11) top-level categories that include 34 tags. The complete tagset used in our experiment is reported in Table 3.

4.3. Language model

Language models are a crucial component in a deep learning-based model. These models are typically trained on large unlabeled text corpus to capture both semantic and syntactic meanings. The size of the training corpora (Bojanowski *et al.* 2017) impacts the quality of the language models. In our literature survey, we could not find any other LMs that covered the Bodo language. The lack of availability of text corpus is one of the main factors behind this. Due to this, the BodoBERT is the first LM that has ever been trained on Bodo text. Therefore, to accomplish our comparative experiment, we consider pretrained language models for Hindi as it shares the same written script with Bodo-Devanagari.

Table 3. Tagset used in the dataset

| S.No | Category | Type | Tag |
|------|---------------|---------------------|------------|
| 1 | Noun | Proper Noun | N_NNP |
| | | Noun (Location) | N_NST |
| | | Noun (unclassified) | N_NN |
| 2 | Pronoun | Personal | PR_PRP |
| | | Reflexive | PR_PRF |
| | | Reciprocal | PR_PRC |
| | | Relative | PR_PRL |
| | | Wh-words | PR_PRQ |
| | | Indefinite | PR_PRI |
| 3 | Demonstrative | Deictic | DM_DMD |
| | | Relative | DM_DMR |
| | | Wh-words | DM_DMQ |
| | | Indefinite | DM_DMI |
| 4 | Verb | Auxiliary Verb | V_VAUX |
| | | Main Verb | V_VM |
| | | Finite | V_VM_VF |
| | | Non-Finite | V_VAUX_VNF |
| 5 | Adjective | Adjective | JJ |
| 6 | Adverb | | RB |
| 7 | Post Position | | PSP |
| 8 | Conjunction | Conjunction | CC_CCD |
| | | Co-ordinator | CC_CCS |
| 9 | Particles | Classifier | RP_RPD |
| | | Interjection | RP_INJ |
| | | Negation | RP_NEG |
| | | Intensifier | RP_INTF |
| 10 | Quantifiers | General | QT_QTF |
| | | Cardinals | QT_QTC |
| | | Ordinals | QT_QTO |
| 11 | Residuals | Foreign word | RD_RDF |
| | | Symbol | RD_SYM |
| | | Punctuation | RD_PUNC |
| | | Echowords | RD_ECH |
| | | Unknown | RD_UNK |

Table 4. Training dataseize of different language models

| Language model | Trained corpus | Trained dataseize |
|---|------------------------------------|-------------------|
| FastTextEmbeddings (Bojanowski et al., 2017) ^e | Wiki | < 29 M |
| Byte Pair (Heinzerling and Strube, 2018) ^f | Wiki | 29M |
| BodoBERT (Bodo) ^g | Bodo corpus (Narzary et al., 2022) | 1.6M |
| Flair Embeddings (Akbik et al., 2018) ^h | Wiki+OPUS | ≈ 29 M |
| MuRIL (Khanuja et al., 2021) ⁱ | CommonCrawl + Wiki | 788M |
| XLm-R Embedding (Conneau et al., 2020) ^j | CC-100 corpus | 1.7B |
| IndicBERT (Kakwani et al., 2020) ^k | Scraping | 1.84B |

Table 4 provides the details about the LMs that are used for training the Bodo POS tagging model. A brief description of these models is described below.

FastText embedding (Bojanowski *et al.* 2017) uses subword embedding technique and skip-gram method. It is trained on character n -grams of words to get the internal structure of a word. Therefore, it has the ability to get the word vectors for out-of-vocabulary (OOV) words by using the subword information from the previously trained model.

Byte-Pair embedding (Heinzerling and Strube 2018) are precomputed on subword level. They can embed by splitting words into subwords or character sequences, looking up the precomputed subword embeddings. It has the capability to deal with unknown words and has the ability to infer meaning from unknown words.

Flair embedding (Akbik, Blythe, and Vollgraf, 2018) is a type of character-level embedding. It is a contextualized word embedding that captures word semantics in context, meaning that a word has different vector representations under different contexts. The embedding method based on recent advances in neural language modeling (Sutskever, Vinyals, and Le, 2014; Karpathy, Johnson, and Fei-Fei, 2015) that provides sentences to be modeled as distributions over sequences of characters instead of words (Sutskever, Martens, and Hinton, 2011; Kim *et al.* 2016).

Multilingual Representations for Indian Languages (MuRIL) (Khanuja *et al.* 2021) is a multilingual language model based on BERT architecture. It is pretrained in 17 Indian languages.

XLm-R (Conneau *et al.* 2020) uses self-supervised training techniques in cross-lingual understanding, a task in which a model is trained in one language and then used with other languages without additional training data.

IndicBERT (Kakwani *et al.* 2020) based on Fasttext-based word embedding and ALBERT-based language models for 11 languages trained on the IndicCorp dataset.

4.4. Experiment on POS models

In this section, we describe the experiments conducted to develop POS tagging models using different LMs, including BodoBERT. The experiment can be divided into three phases. In the first phase, we employ three different sequence labeling architectures that are shown to be a well

^e<https://fasttext.cc/docs/en/pretrained-vectors.html>

^f<https://github.com/bheinzerling/bpemb>

^g<https://anonymous.4open.science/status/BodoPoS-4C10>

^hhttps://github.com/flairNLP/flair/blob/master/resources/docs/embeddings/FLAIR_EMBEDDINGS.md

ⁱ<https://huggingface.co/google/muril-base-cased>

^j<https://tinyurl.com/XLM-R-Embed>

^k<https://indictlp.ai4bharat.org/indic-bert/>

Table 5. POS tagging performance in the stacked method using BiLSTM-CRF architecture

| Stacked embeddings | F1 score |
|--------------------------------------|---------------|
| BodoBERT + FastTextEmbeddings | 0.7928 |
| BodoBERT + BytePairEmbeddings | 0.8041 |
| BodoBERT + mBERT | 0.799 |
| BodoBERT + FlairEmbeddings | 0.801 |
| BodoBERT + MuRIL | 0.785 |
| BodoBERT + XLM-R | 0.8003 |
| BodoBERT + IndicBert | 0.793 |

performer in other languages, namely - Fine-tuning BodoBERT model for POS tagging, CRF (Lafferty, McCallum, and Pereira 2001) and the third one with BiLSTM-CRF (Hochreiter and Schmidhuber 1997; Rei, Crichton, and Pyysalo 2016). The pretrained BodoBERT is used to get the embedding vector of the words present in the sentences during training with CRF and BiLSTM-CRF (Huang, Xu, and Yu 2015) architecture. It is observed that the BiLSTM-CRF-based model outperforms the other two models.

Therefore, the BiLSTM-CRF architecture is used in the second phase to develop POS tagging models employing different language models, including BodoBERT.

Apart from Bodo, we also conducted the same experiment on Assamese, another low-resource language. Existing Assamese pretrained LMs are used for the experiment. To conduct the training for Assamese POS model, we acquired dataset (ILCI-II, 2020a) from TDIL, which was tagged by language experts manually. There are 35k sentences from different domains and 404k words in the POS datasets. The Assamese annotated dataset also follows the BIS tagset and contains 41 tags with 11 top-level categories.

In the third phase, we further experiment with the Stacked embedding method to develop the Bodo POS tagging mode. The Stacked method allows us to learn how well one embedding performs when combined with others during the training process. The top-performing LM (BodoBERT) in the second phase is selected for further training. In the Stacked method, each one of the LMs is combined with BodoBERT to get the final word embedding vector.

Compared to the best individual method, the stacked embedding approach using BiLSTM-CRF improves the performance score for POS tagging by around 2–7%. The model with **BodoBERT + BytePairEmbedding** attains the highest F1 score of 0.8041. The results of the experiment are listed in Table 5. The POS model architecture is illustrated in Figure 1. In all experiments, the Flair framework¹ is used to train the sequence tagging model.

We explored different hyperparameters to optimize the configurations concerning the hardware constraint. After that, we use the same set of hyperparameters in all three experiments. We use a fixed mini-batch size of 32 to account for memory constraints. The early stopping technique is used if there is no improvement in the accuracy of the validation data. We use the learning rate annealing factor for early stopping.

¹<https://github.com/flairNLP/flair>

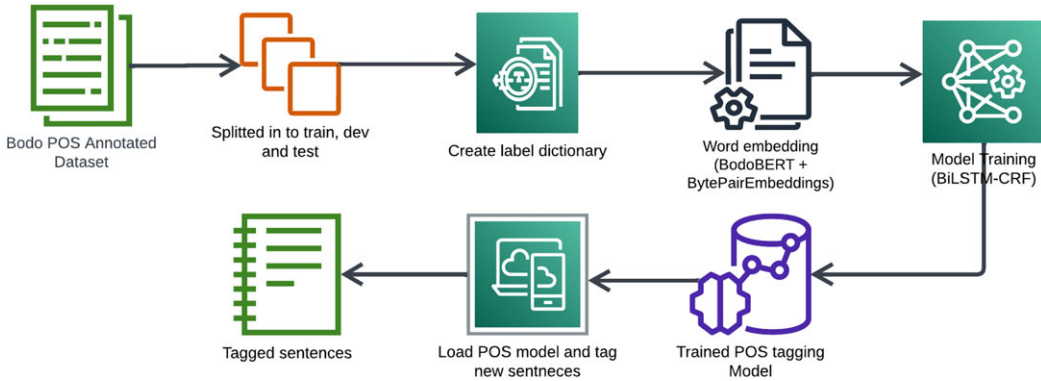


Figure 1. Block diagram of POS tagging model.

5. Result and analysis

In this section, we present an analysis of our experiment and the performance of taggers. We evaluated the performance of the three sequence tagging methods: CRF, Fine-tuning of BodoBERT, and BiLSTM-CRF by measuring the micro F1 score. The weighted average of the tagging performance is also reported.

Table 6 illustrates an overview of the tagging performance in F1 score of the three models. We observe that the BiLSTM-CRF based model with BodoBERT performs the best with an F1 score of 0.7949 and weighed the average F1 score as 0.7898. In contrast, the fine-tune-based and CRF-based tagging model achieves an F1 score of 0.7754 and 0.7583, respectively. The performance comparison result of different LMs is reported in Table 7. We observe BodoBERT outperforms all the other pretrained models. The Flair Embedding model achieves almost similar performance in tagging with an F1 score of 0.7885.

We also cover a similar low-resource language, Assamese spoken in the same region. The same set of experimental setups is used for the experiment. It provides us with an overview of how the models work on similar languages with almost the same size as the annotated dataset. It has been observed that the highest F1 score achieved in the case of Assamese is 0.7293 using IndicBERT. This is almost ≈ 7 less than the highest of the Bodo POS tagging model. It could be because of the difference in the number of tagsets used in Assamese (41 tags) versus Bodo (34 tags).

Data augmentation experiment: The overall best-performing model (BodoBERT+BytePairEmbedding) is employed to annotate a new set of sentences from another corpus. The corpus is taken from the Bodo corpus created by Narzary *et al.* (2022). The annotated dataset is further manually corrected. A new dataset of 10k has been added to the existing training dataset. In order to evaluate the performance of the model when increasing the training dataset, the same architectures (BodoBERT+BytePairEmbedding in BiLSTM-CRF) are employed in further training the POS model with the same set of parameters. The test and dev are kept the same. It is observed that the model performance is enhanced by $1 \approx 2\%$. The model achieves an F1 score of 0.8494.

The tag-wise performance score of precision, recall, and F1 score, along with support consolidated micro, macro, and weighted score, are reported in Table 8. The learning curve of training and validation for the best-performing model is shown in Figure 2.

The learning curve implies that the dataset needs improvement as the training dataset does not provide sufficient information to learn the sequence tagging problem relative to the validation dataset used to evaluate it. Whatsoever, we get the first POS tagging model for the Bodo language. Eventually, it becomes the de facto baseline for the Bodo tagging.

Table 6. Performance of POS tagging model in different methods

| Embeddings | Tagging model | F1-score (micro) | F1 score (weighted) |
|------------|---------------------|------------------|---------------------|
| BodoBERT | CRF | 0.7583 | 0.7454 |
| | Fine-tuned BERT | 0.7754 | 0.7775 |
| | BiLSTM + CRF | 0.7949 | 0.7898 |

Table 7. The F1 score of different language models in POS tagging task on Bodo and Assamese language

| Embeddings | Bodo | Assamese |
|-------------------------|---------------|---------------|
| FastTextEmbeddings | 0.7686 | 0.6981 |
| BytePairEmbeddings | 0.7669 | 0.7099 |
| BodoBERT | 0.7949 | 0.7033 |
| FlairEmbeddings (Multi) | 0.7885 | 0.7076 |
| MuRIL | 0.7708 | 0.7286 |
| XLM-R | 0.7638 | 0.7001 |
| IndicBert | 0.7235 | 0.7293 |

The reference sentences and their corresponding tagged results from the proposed tagger are given below.

- Reference sentences

Sentence 1: तिकेन < N_NNP > बर'आ < N_NNP > सासे < QT_QTC > मोजां < JJ >
फोरोंगिरि < N_ANN > | < RD_PUNC >

IPA: tiken boroa sase mojaŋ
foroŋgiri

English: *Tiken Bodo is a good teacher*

Sentence 2: बडलेन्ड < N_NNP > मुलुगसोलोसालिआ < N_NNP > कक्राझारआव
< N_NNP > दं < V_VAUX > | < RD_PUNC >

IPA: boroland muluɡɔsolosaliya kokrajharɑ
dɔŋ

English: *Bodoland University is situated in Kokrajhar*

- Tagged by the proposed DL-based tagger

Sentence 1: तिकेन < N_NNP > बर'आ < N_NN > सासे < QT_QTC > मोजां < JJ >
फोरोंगिरि < N_NN > | < RD_PUNC >

IPA: tiken boroa sase mojaŋ
foroŋgiri

English: *Tiken Bodo is a good teacher*

Sentence 2: बडलेन्ड < N_NNP > मुलुगसोलोसालिआ < N_NN > कक्राझारआव < N_NN >
दं < V_VAUX > | < RD_PUNC >

IPA: boroland muluɡɔsolosaliya kokrajharɑ
dɔŋ

English: *Bodoland University is situated in Kokrajhar*

Table 8. Tag-wise performance of best-performing Bodo POS tagging model

| Tags | Precision | Recall | F1 score | Support |
|---------|-----------|--------|----------|---------|
| N_NN | 0.7439 | 0.7826 | 0.7628 | 11560 |
| V_VM | 0.8945 | 0.9366 | 0.9150 | 9005 |
| RD_PUNC | 0.9927 | 0.9960 | 0.9944 | 6815 |
| N_NNP | 0.7180 | 0.6727 | 0.6946 | 5264 |
| JJ | 0.6665 | 0.5554 | 0.6059 | 1975 |
| N_NST | 0.5703 | 0.5194 | 0.5436 | 1421 |
| CC_CCD | 0.9059 | 0.9634 | 0.9338 | 1039 |
| DM_DMD | 0.9699 | 0.9593 | 0.9646 | 908 |
| PSP | 0.6492 | 0.7624 | 0.7012 | 665 |
| QT_QTC | 0.7319 | 0.8743 | 0.7968 | 565 |
| RB | 0.7295 | 0.4958 | 0.5904 | 593 |
| PR_PRP | 0.8584 | 0.8491 | 0.8537 | 464 |
| RD_UNK | 0.6613 | 0.0861 | 0.1524 | 476 |
| RD_ECH | 0.2155 | 0.4266 | 0.2864 | 143 |
| QT_QTF | 0.3673 | 0.1949 | 0.2547 | 277 |
| PR_PRI | 0.6552 | 0.6683 | 0.6617 | 199 |
| CC_CCS | 0.4789 | 0.5574 | 0.5152 | 122 |
| RP_INTF | 0.3699 | 0.4154 | 0.3913 | 65 |
| V_VAUX | 0.4493 | 0.5082 | 0.4769 | 61 |
| PR_PRF | 0.1205 | 0.2222 | 0.1562 | 45 |
| RD_SYM | 0.9600 | 0.6486 | 0.7742 | 74 |
| QT_QTO | 0.5758 | 0.7308 | 0.6441 | 52 |
| DM_DMI | 0.2292 | 0.1864 | 0.2056 | 59 |
| RD_RDF | 1.0000 | 0.9455 | 0.9720 | 55 |
| PR_PRC | 0.1379 | 0.2963 | 0.1882 | 27 |
| PR_PRL | 0.5000 | 0.1346 | 0.2121 | 52 |
| PR_PRQ | 0.3704 | 0.3571 | 0.3636 | 28 |
| RP_RPD | 0.0400 | 0.0526 | 0.0455 | 19 |
| DM_DMQ | 0.4615 | 0.8000 | 0.5854 | 15 |
| RP_NEG | 0.0000 | 0.0000 | 0.0000 | 5 |
| RP_INJ | 0.0000 | 0.0000 | 0.0000 | 4 |

Table 8. Continued

| Tags | Precision | Recall | F1 score | Support |
|--------------|-----------|--------|----------|---------|
| DM_DMR | 0.0000 | 0.0000 | 0.0000 | 4 |
| micro avg | 0.8041 | 0.8041 | 0.8041 | 42056 |
| macro avg | 0.5320 | 0.5187 | 0.5076 | 42056 |
| weighted avg | 0.8021 | 0.8041 | 0.7990 | 42056 |

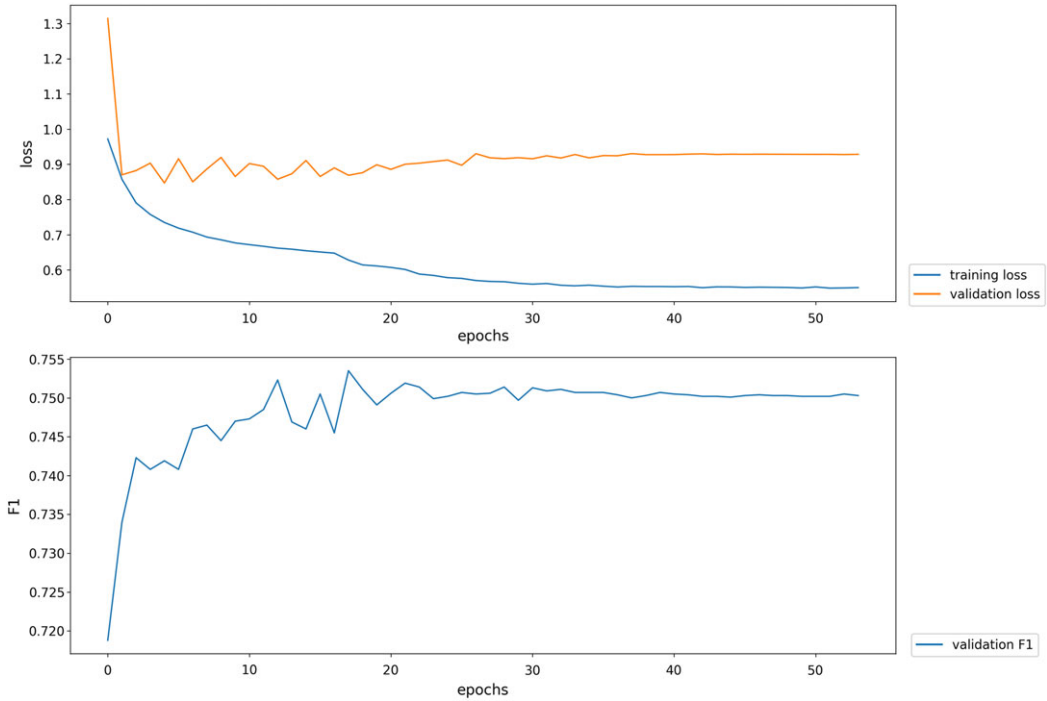


Figure 2. Learning curve of BodoBERT + BytePairEmbeddings based POS model.

In the above example, the word ‘बर’आ/boroa/ (‘Bodo’) in sentence 1, and मुलुगसो-लौसालिआ/mulugosoloᅇsalia (University + nominative marker ‘-ᅇ’) and कक्राझारआव kokrajharᅇᅇ (Kokrajhar + locative marker ‘-ᅇᅇ’) in sentence 2 are proper nouns (*N_NNP*). But the tagger considers them as NOUN (generalized form) and tags them accordingly as *N_NN*. Likewise, फोरोगिरि/ foroᅇgiri (Teacher) is an abstract noun (*N_ANN*). However, it is tagged as *N_NN* by the tagger. In other cases, the POS tagged the words correctly. If we consider only the top-level tag, the tagger performance increases.

The reported highest score is arguably lower than the state-of-the-art score on resource-rich languages. This could be due to a variety of factors.

1. The size of the annotated corpus may not be adequate.
2. The training data size of BodoBERT may not be sufficient enough to capture the linguistic features of the language.

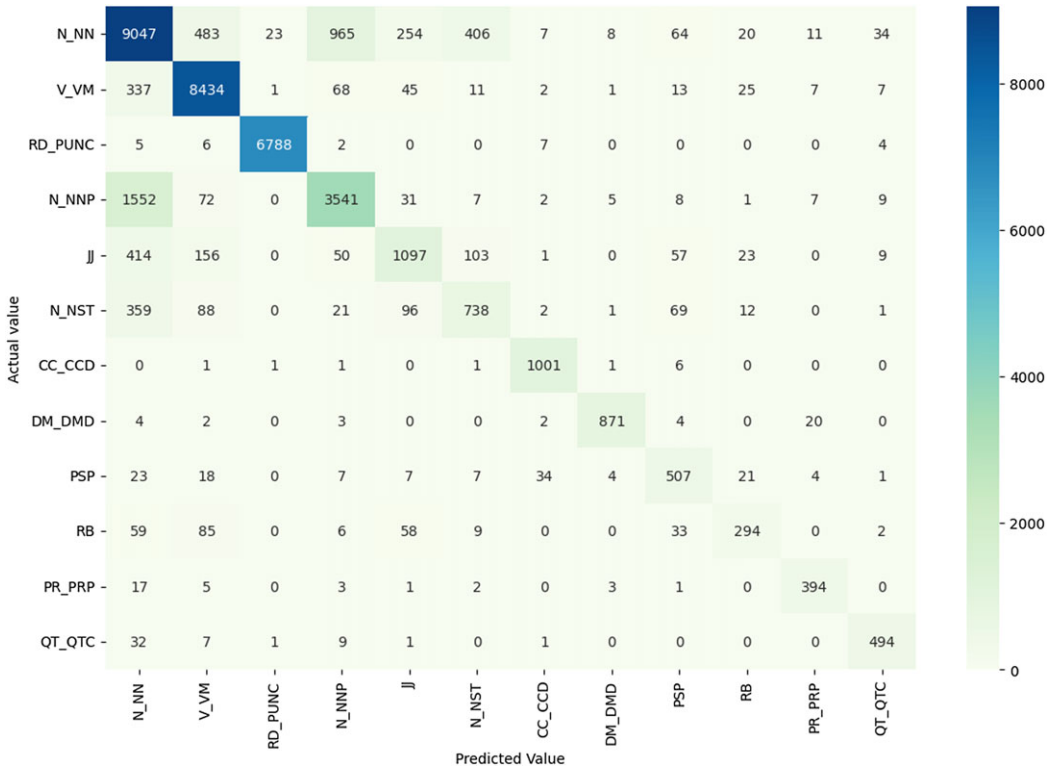


Figure 3. Confusion matrix of BodoBERT + BytePairEmbeddings POS model.

3. The LM BodoBERT may need more improvement in capturing the linguistic characteristics of the Bodo language.
4. The assignment of the tags to the words in the dataset may need some correction.

Observation from confusion matrix: The confusion matrix of the top-performing Bodo POS tagging model is reported in Figure 3. The confusion matrix covers the top twelve POS tags in terms of frequency counts in the test set. The diagonal entries of the matrix represent the correctly predicted tags, and the off-diagonal entries represent incorrectly classified tags.

It is observed that the common error occurs in Noun (N_NN), Proper Noun (N_NNP), Locative Noun (N_NST), VERB (V_VM), Adjective (JJ), and Adverb (RB) in the taggers. We can draw the following observation from the confusion matrix.

- The most common error in the dataset is intra-class confusion, i.e., confusing Noun (N_NN) with Proper (N_NNP) and Noun (N_NN) with Locative noun (N_NST). This might have occurred due to the similarity in the attached features of nouns, pronouns, and locative nouns.
- In some instances, the tagger incorrectly predicts a noun when its class-type changes to an adjective in a sentence. It happens when it describes another noun, e.g., ‘Bodo language’; in this case, although *Bodo* is a proper noun, it is being used to describe the noun- *language*. Therefore, it becomes an adjective.
- Sometimes, it becomes difficult to figure out the correct tag- JJ or N_NN, V_VM or N_NN, and RB or N_NN. So, many times, the tagger incorrectly tags as N_NN for JJ, V_VM, and RB.

- Furthermore, Bodo has no similar orthographic conventions to differentiate the proper nouns as done using capitalization in English. Therefore, it is difficult for a machine to differentiate a proper noun from other nouns.

6. Conclusion

In this work, we presented a LM, BodoBERT, for the Bodo language based on BERT architecture.

We develop a POS tagging model employing three distinct sequence tagging architectures using the BodoBERT. Upon applying the BodoBERT LM in POS tagging, we obtained two outcomes: first, an evaluation of the pretrained BodoBERT's performance on a downstream task in different architectures; second, we obtained a model for POS tagging in the Bodo language. We also compared the performance of BodoBERT to that of other LMs using two methods: Individual and Stacked. The stacked method improves the performance of the POS tagging model. In our experiment, the model that uses BodoBERT and BytePairEmbeddings together in a stacked method does better.

Despite the fact that the Bodo POS tagger is unable to attain state-of-the-art accuracy in comparison to resource-rich languages, we feel that our POS model can serve as a baseline for future studies. Our contributions may be useful to the research community in terms of using the LM BodoBERT and the POS tagging model for various downstream tasks.

References

- Aguilar G., Maharjan S., López-Monroy A. P. and Solorio T. (2019). A multi-task approach for named entity recognition in social media data. arXiv preprint arXiv: 1906.
- Akbik A., Blythe D. and Vollgraf R. (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pp. 1638–1649.
- Alam F., Chowdhury S. A. and Noori S. R. H. (2016). Bidirectional lstms–crfs networks for bangla pos tagging. In *2016 19th International Conference on Computer and Information Technology (ICCIT)*, IEEE, pp. 377–382.
- Bhutani N., Suhara Y., Tan W.-C., Halevy A. and Jagadish H. V. (2019). Open information extraction from question-answer pairs. arXiv preprint arXiv: 1903.
- Bojanowski P., Grave E., Joulin A. and Mikolov T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5, 135–146.
- Choudhary N. (2021). Ldc-il: the indian repository of resources for language technology. *Language Resources and Evaluation* 55(3), 855–867.
- Conneau A., Khandelwal K., Goyal N., Chaudhary V., Wenzek G., Guzmán F., Grave É., Ott M., Zettlemoyer L. and Stoyanov V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451.
- Devlin J., Chang M.-W., Lee K. and Toutanova K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv: 1810.04805.
- Heinzerling B. and Strube M. (2018). BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan: European Language Resources Association (ELRA).
- Hochreiter S. and Schmidhuber J. (1997). Long short-term memory. *Neural Computation* 9(8), 1735–1780.
- Huang Z., Xu W. and Yu K. (2015). Bidirectional LSTM-CRF models for sequence tagging, arXiv preprint arXiv: 1508.01991.
- ILCI-II, J. (2020a). Assamese monolingual text corpus ILCI-II.
- ILCI-II, J. (2020b). Bodo monolingual text corpus ILCI-II.
- Kabir M. F., Abdullah-Al-Mamun K. and Huda M. N. (2016). Deep learning based parts of speech tagger for Bengali. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, IEEE, pp. 26–29.
- Kakwani D., Kunchukuttan A., Golla S., Gokul N., Bhattacharyya A., Khapra M. M. and Kumar P. (2020). IndicNLPsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 4948–4961.
- Karpathy A., Johnson J. and Fei-Fei L. (2015). Visualizing and understanding recurrent networks, arXiv preprint arXiv: 1506.02078.
- Khanuja S., Bansal D., Mehtani S., Khosla S., Dey A., Gopalan B., Margam D. K., Aggarwal P., Nagipogu R. T., Dave S., Gupta S., Gali S. C. B., Subramanian V. and Talukdar P. (2021). MuriL: Multilingual representations for indian languages.

- Kim Y., Jernite Y., Sontag D. and Rush A. (2016). Character-aware neural language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30.
- Kingma D. P. and Ba J. (2014). Adam: a method for stochastic optimization, arXiv preprint arXiv: [1412.6980](https://arxiv.org/abs/1412.6980).
- Lafferty J., McCallum A. and Pereira F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- Le-Hong P. and Bui D.-T. (2018). A factoid question answering system for vietnamese. In *Companion Proceedings of the The Web Conference 2018*, pp. 1049–1055.
- Narzary S., Brahma M., Narzary M., Muchahary G., Singh P. K., Senapati A., Nandi S. and Som B. (2022). Generating monolingual dataset for low resource language bodo from old books using google keep. In *Proceedings of the Language Resources and Evaluation Conference*, Marseille, France: European Language Resources Association, pp. 6563–6570.
- Niehues J. and Cho E. (2017). Exploiting linguistic resources for neural machine translation using multi-task learning. arXiv preprint arXiv: [1708.00993](https://arxiv.org/abs/1708.00993).
- Pandey S., Dadure P., Nunsanga M. V. and Pakray P. (2022). Parts of speech tagging towards classical to quantum computing. In *2022 IEEE Silchar Subsection Conference (SILCON)*, IEEE, pp. 1–6.
- Pathak D., Nandi S. and Sarmah P. (2022). Aspos: Assamese part of speech tagger using deep learning approach. In *2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA)*, IEEE, pp. 1–8.
- Pathak D., Nandi S. and Sarmah P. (2023). Part-of-speech tagger for assamese using ensembling approach. *ACM Transactions on Asian and Low-Resource Language Information Processing* **22**(10), 1–22.
- Ramamoorthy L., Choudhary N., Basumatary B. and Daimary F. (2019). Gold standard Bodo raw text corpus.
- Rei M., Crichton G. K. and Pyysalo S. (2016). Attending to characters in neural sequence labeling models, arXiv preprint arXiv: [1611.04361](https://arxiv.org/abs/1611.04361).
- Sang E. F. and De Meulder F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition, *arXiv preprint cs/0306050*.
- Shen Y., Lin Z., Jacob A. P., Sordoni A., Courville A. and Bengio Y. (2018). Straight to the tree: Constituency parsing with neural syntactic distance arXiv preprint arXiv: [1806.04168](https://arxiv.org/abs/1806.04168).
- Sutskever I., Martens J. and Hinton G. E. (2011). Generating text with recurrent neural networks. In *ICML*.
- Sutskever I., Vinyals O. and Le Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pp. 3104–3112.
- Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł. and Polosukhin I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, vol. 30.
- Warjri S., Pakray P., Lyngdoh S. A. and Maji A. K. (2021). Part-of-speech (pos) tagging using deep learning-based approaches on the designed khasi pos corpus. *Transactions on Asian and Low-Resource Language Information Processing* **21**(3), 1–24.
- Wu Y., Schuster M., Chen Z., Le Q. V., Norouzi M., Macherey W., Krikun M., Cao Y., Gao Q., Macherey K., et al. (2016). Google’s neural machine translation system: Bridging the gap between human and machine translation, arXiv preprint arXiv: [1609.08144](https://arxiv.org/abs/1609.08144).