North South University

Department of Electrical and Computer Engineering

BACHELOR'S DIRECTED RESEARCH

Bangla POS Tagging Using Supervised Learning and Knowledge Distillation.

Md. Abu Ammar - 1821944042,
Sadia Afrin Tamanna - 1812030042

Supervised by

Dr. Nabeel Mohammed

Associate Professor

Department of Electrical and Computer Engineering

Summer 2022

Bangla POS Tagging Using Supervised Learning and Knowledge Distillation.

A directed research submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Computer Engineering

Submitted by

Md. Abu Ammar **1821944042**

Sadia Afrin Tamanna **1812030042**

Supervised by

Dr. Nabeel Mohammed

Associate Professor

Department of Electrical and Computer Engineering

Department of Electrical and Computer Engineering
North South University
Dhaka-1229, Bangladesh

July, 2026

Declaration

We, do, hereby, certify that the work presented in this directed research, titled, *"Bangla POS Tagging Using Supervised Learning and Knowledge Distillation."*, is the outcome of the investigation and research carried out by us under the supervision of *Dr. Nabeel Mohammed*, Associate Professor, Department of ECE, NSU. We also declare that neither this directed research nor any part thereof has been submitted anywhere else for awarding any degree, diploma, or other qualifications. Any material reproduced in this project has been properly acknowledged

Student's Names and Signatures :

1. Md. Abu Ammar

2. Sadia Afrin Tamanna

Dhaka, Bangladesh
Saturday 4th July, 2026

Approval

The directed research titled **"Bangla POS Tagging Using Supervised Learning and Knowledge Distillation."**, submitted by Md. Abu Ammar, Sadia Afrin Tamanna, in the Summer 2022 session, to the Department of Electrical and Computer Engineering, North South University, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Computer Engineering and approved as to its style and contents on Saturday 4th July, 2026.

Supervisor's Signature :

Dr. Nabeel Mohammed
Associate Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

Department Chair's Signature :

Dr. Rajesh Palit
Professor and Chair
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

Dedication

This work is dedicated to the Almighty Allah Subhanahu Ta'ala. Also our beloved parents and teachers. This possibly could not be achieved without their help and support.

Acknowledgement

First and foremost, we would like to express our deep and sincere gratitude to our supervisor Dr. Nabeel Mohammed, Associate Professor, Department of Electrical and Computer Engineering, North South University for giving us the opportunity to work with him and giving us the required guidance to complete this project. His motivation, vision and ideas allowed us to find this research topic and work on it. I would also like to thank him for his empathy toward us and for understanding the problems we had to face during the pandemic.

We also acknowledge our profound sense of gratitude toward all the teachers at North South University who have dedicated their time in enriching our knowledge in various philosophies. This enhanced our skills gradually which ultimately boosted our confidence leading to this final year project.

Finally we thank profusely all our friends and relatives that always found a way to support us emotionally, morally, physically or financially.

Dhaka, Bangladesh
Saturday 4th July, 2026

Contents

List of Tables

List of Figures

Abstract

The field of NLP research greatly benefits from the study of PoS Tagging. Establishing a competent POS tagger for a low-resource language like Bangla is still a difficult task. The Microsoft IL post dataset of Bangla PoS Tag contains numerous class imbalance difficulties as a result of the deep learning model's poor performance on this dataset since it is biased against the issue of class imbalance. We observed this problem by training a neural network on the contextual meaning embeddings of the word extracted from state of the art Bangla BERT language models. While a Decision Tree didn't perform as satisfactorily in contrast to the Neural Network, in our experiments, its performance was observed to be less affected by the class imbalance. In this paper, we have addressed the imbalance issue by developing a decision tree model as a teacher and distilling its knowledge to a neural network student model. This is done by regarding the number of classes in the terminal node as a probability distribution which acts like dark knowledge to the student model.

1 Introduction

Parts-of-speech (POS) tagging, a common NLP technique, refers to classifying words in a text in accordance with a specific part of speech, depending on the word's definition and its context. A part-of-speech tag, also known as a POS tag, is a unique label that is placed on each token in a corpus of text to indicate the part of speech and also other grammatical categories such as tense, number, gender, case, etc. POS tagging facilitates linguistic and statistical analysis in different contexts as they can provide the syntactic roles of different words in a corpus. Hence, it has a wide range of applications such as Name Entity Recognition (NER), question answering, grammar correction and sentiment analysis. As a result, its potential can be extended to support practically all branches of NLP. The performance of POS tagging has reached previously unheard-of heights in recent years thanks to the revival of neural network-based techniques. Unfortunately, this development has generally been restricted to languages that have an abundance of resources and datasets with the right annotations because of the nature of the technology. Such datasets are just unavailable for a large number of low resource languages, particularly Bangla. As a result, relatively little progress has been accomplished in this area. As we show in this paper, innovative approach like Knowledge Distillation can be leveraged to alleviate the low resource problem for Bangla language.

Bangla is the state language of Bangladesh and is one of the most widely spoken languages in the world, particularly in Bangladesh and parts of India such as Tripura, West Bengal, and southwestern Assam. In fact, it occupies the seventh position in terms of the number of speakers around the world. However, it still lacks enough resources for achieving comparatively good results in contrast with the other abundantly spoken languages. To contribute in the NLP field for Bangla, creating an automatic POS tagger will be crucial

[1]. Character-level characteristics, rule-based systems, or sequence modeling techniques make up the bulk of the research on Bangla PoS tagging [2] [3] [4] . Although a dataset's imbalance is not always a problem for rule-based systems, they have generally been shown to be less successful. The class imbalance problem is a concern for machine-learning-driven systems, particularly those that use neural networks.

For instance, the Microsoft IL POST dataset, which was used in this study, has samples for 32 tags, the majority of which are grouped into 5 groups. When using general sequence learning models with this dataset, the trained model will invariably favor the majority classes while ignoring the minority classes. The work of [2] uses a Conditional Random Field (CRF) on top of a BERT model and trains using class-specific weights as recent attempts to resolve the imbalance of this dataset. Besides, using inconsequent model with oversampling techniques has overcome this class imbalance problem as cite. They employ cost sensitive learning and oversampling strategies, respectively, and while they show good results, the goal of this research is to determine whether we can apply a knowledge distillation approach in this context.

The solution proposed in this research uses a decision tree model to classify PoS tags using contextual BERT embeddings on a word-by-word basis, followed by training a neural network on the same data to incorporate the decision tree model's knowledge into the neural network model. The main contributions of this paper can be summarized as follows:

- We demonstrate that regarding the number of classes found in a terminal node of a Decision Tree as a probability distribution can retain semantic information that is less biased by class imbalance with respect to Deep Learning approaches. - We improved the performance of a neural network on the task of POS tagging by leveraging dark knowledge gained from a decision tree.

1.1 Organization of the directed research

The research work performed in this study is divided into different topics and presented within six chapters including this chapter. The first Chapter of this research describes the introduction of the study.

The Second Chapter provides an extensive literature review based on the recent work that has been done on Bangla Parts Of Speech Tagging area.

The Third Chapter consists of the methodology of our research. Why and how we have build the models.

Chapter Four delivers the experimental setup of this research.

The Fifth Chapter of this research is about the result analysis of our research where we will be folding some research question related with our research approach.

And finally the Sixt Chapter of this research will conclude the conclusion of our research.

2

Related Work

Rule-based, stochastical, and neural network tagging are the three main types of prevalent techniques to PoS tagging. The first PoS tagging system used a rule-based approach, in which PoS tags were assigned in accordance with some pre-established rules. Stochastic techniques include frequency, statistics, or probability. This method clarifies the words based on the likelihood that they are associated with a particular tag. The artificial neural network is the foundation of the third PoS tagging method. PoS tagging is a sequence labeling problem in the context of neural network approaches, and as such, sequence modeling techniques like LSTM and GRU have demonstrated superior performance to earlier methods and may be able to mitigate the vanishing gradient issues in conventional recurrent neural networks. The most crucial element of sequence label modeling is word embeddings, which are necessary to obtain cutting-edge performance. In order to learn the word representation, a shallow network, such Word2Vec or glove embeddings, is often trained on a sizable unlabeled text corpus. BERT and ELMO embeddings, which obtain the contextual representation of a word, are frequently utilized in sequence labeling and produce cutting-edge outcomes. Developed a Bangla Stemmer and a set of rules for word-level tagging in the Bangla PoS context. Proposed a classical strategy for the assignment. [3] In addition to this, a number of traditional methodologies have been used, including the allomorphic [5] method and rule-based with layered tagging [6]. With a limited training set of 5,000 words and 41 tag sets, the author investigated a statistical technique and transformation-based approach for it and reached an accuracy of 55% [7]. A BiLSTM and CRF were used to predict the Bangla tags on the dataset in the author's knowledge-poor method to Bangla PoS tagging. [7] This method's Bangla PoS tagging accuracy was 86Unfortunately, a significant amount of Bangla PoS Tagging research has focused

mostly on datasets gathered for specific publications and made public, which eliminates the chance. Although this collection does not offer the typical train, validation, and test partitions, we nevertheless used it for all of our studies despite this fact. Intrinsic characteristics constructed from discrete character level representations are used in one of the first articles that report the findings from the IL POST dataset. These representations were utilized by the authors to create a Deep Belief Network that forecasts the PoS tags. Recent research combined a Conditional Random Field (CRF) with class-specific weights with contextual BERT embeddings. The Macro F1 score for the paper is 79.84%. Another recent study [8] identifies the problem of class imbalance and addresses it by employing oversampling techniques in which the minority class was only oversampled, not the other classes, to avoid bias. They claim a Macro F1 score of 78.35% on the test set. To solve the class disparity problem, none of them adopted the knowledge distillation strategy.

3
Methodology

In this chapter, we will discuss about our research methodology.

3.1 Methond with System Diagram

This paper comprise decision tree models and neural network models for the three BERT Bangla language model mentioned in Table 5.1 to leverage the word embedding to transform more contextual than other word embedding methodologies i.e word2vec, globe . Bert language model has 12 hidden layers subsequently and each of them has the ability to produce contextual word embedding. Our aim was to leverage the hidden layers into that extends that we can get the best, average, and worst layer for the contextual word embedding for each of the BERT Bangla language models. This evaluation of layers gave us the three desired layers from 12 hidden BERT layers.

A decision tree model and a convolutional neural network model have been implemented upon these 3 layers. Then finally we will leverage the idea of knowledge distillation to build a teacher and student model. In our research, the decision tree model is the teacher model and the convolutional neural network model is the student model. All the models use the Scikit Learn and Pytorch platform in the background to generate the parts of speech tag for the particular Bangla word.

3.2 Architecture

The overall block diagram of our Bangla Parts of Speech tagging is shown in Figure 1. As can be seen from the block diagram, the original Bangla word goes through the Bangla

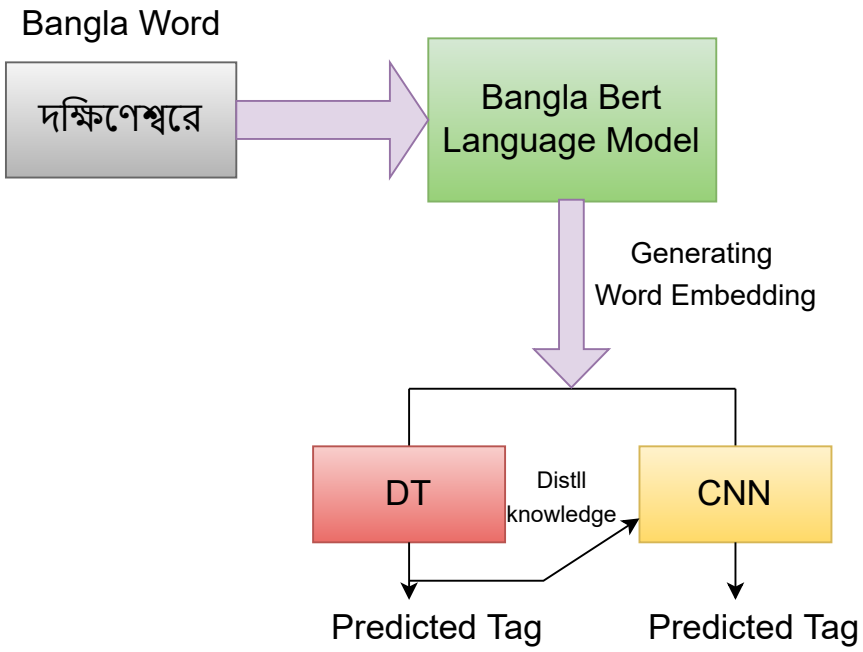

Figure 3.1: BanglaPoS Tagging Pipeline.

bert language model that extracts the word into the word embedding from the selected layer then the generated word embeddings are matched with its corresponding PoS tags. Correspondingly a decision tree model and a convolutional neural network model train on the generated word embedding and each model will predict a distinct PoS tag for the given word embedding as input. We will then build a teacher-student model. The architecture of the neural network is a dense neural network and the decision tree is a general decision tree with a softmax score as an output for both types of models.

4

Experimental Setup

4.1 Dataset

Although the semantics and etymology of the Bangla language are quite rich, there hasn't been sufficient data collection for the PoS tagging task, which makes it challenging to undertake model benchmarking for this task. We use the Indian Language Part-of-Speech Tagset: Bengali dataset[9] from Microsoft Research in India to carry out all the experiments described in this paper. This is the only dataset for the Bangla language that is accessible for academic study and might be used to test a PoS tagging model. 7.168 sentences from the sample have 102,933 hand-added tags. There are other levels of PoS tag annotations; however, for the sake of this paper, we only use the first level, or universal lexical categories, which consists of 32 tags, as done in [4] and [2]. Due to the lack of a typical train, test, and validation partition in the dataset, we follow [8] to train, test, and validation at the sentence level in the following proportions: 60:20:20. We made sure samples from each class were present in each of the three partitions. Table 4.1 displays the tag IDs and the names that correspond to them.

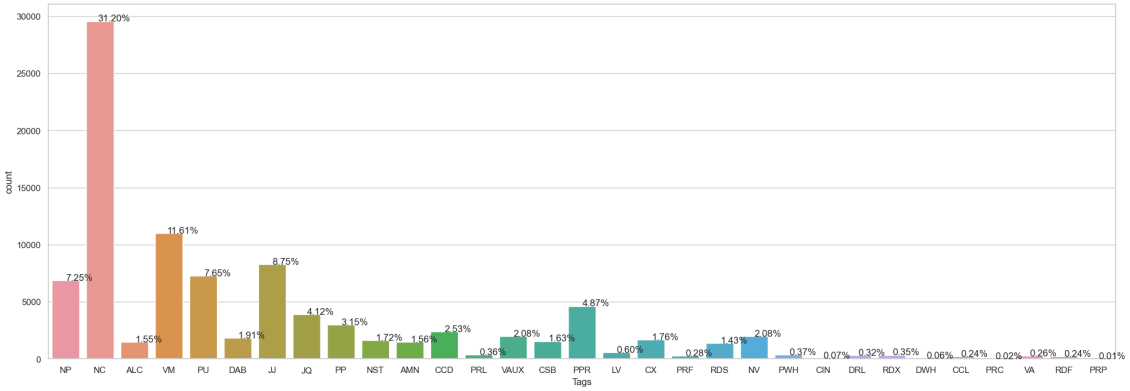

Figure 4.1: Data Distribution of Microsoft IL-POST.

Table 4.1: Names of PoS tags and the identifiers that are incorporated into the dataset.

Identifier	**Name**	**Identifier**	**Name**
NC	Common Noun	PRC	Reciprocal Pronoun
NP	Proper Noun	PRL	Relative Pronoun
NV	Verbal Noun	PWH	Wh Pronoun
NST	Spatio-temporal Noun	JJ	Ajective Nominal Modifier
AMN	Manner Adverb	CCL	Classifier Particles
ALC	Location Adverb	CIN	Interjection Particles
LV	Verbal Participle	RDF	Foreign Word Residual
VM	Main Verb	JQ	Quantifier Nominal Modifier
VA	Auxiliary Verb	DAB	Absolute Demonstratives
PPR	Pronominal Pronoun	DRL	Relative Demonstratives
PRF	Reflexive Pronoun	DEWH	Wh Demonstratives
PP	Postposition	RDS	symbol Residual
PU	Punctuation	RDX	Other Residual
CCD	Coordinating Particles	VAUX	Auxiliary Verbs
CSV	Subordinating Particles	PRP	Pronominal Pronoun
CX	Other Particles	LRL	Relative Participle

The difficulty in using this dataset is primarily brought on by the enormous class imbalance, which is depicted in the bar chart in Fig 4.1. The most frequent tag, NC, is more than two times as common as the second-most frequent phrase, VM, as seen in Fig!4.1. Due to VA's lesser frequency than VAUX and the fact that both tags VA and VAUX refer to the Auxiliary Verb, they were both included in the VAUX class. Due to the aforementioned filtering, the 32 tags were subsequently reduced to 30 tags.

4.2 BERT Layer Comparison

Bidirectional Encoder Representations from Transformers, or BERT, is a deep learning model that is based on Transformers. In Transformers, each output element is connected to each input element, and the weightings between them are dynamically determined based upon their connection.

BERT uses text as an input and creates vectors or embeddings of each word; they are then used as high-quality feature inputs to subsequent models. NLP models like LSTMs and CNNs need inputs in the form of numerical vectors, which often entails converting lexical and grammatical elements into numerical representations. Words were previously either represented as singularly indexed values (one-hot encoding) or, more usefully, as neural word embeddings, where vocabulary words were matched against the fixed-length feature embeddings produced by models like Word2Vec or Fasttext. Compared to models like Word2Vec, BERT has an advantage since it creates word representations that are dynamically influenced by the words surrounding them. Under Word2Vec, each word has a fixed representation regardless of the context in which it appears.

Contextual word embeddings can be generated by BERT language models from all 12 of their hidden layers. We wanted to determine each hidden layers performance in Bangla sentences by evaluating the cos

$$\text{similarity}(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.1}$$

similarity using equation 1 of a particular word in two sentences where that particular word hold distinct and different meaning in each sentence. For this experiment, we have built our own

custom dataset of a total of 30 pairs of Bangla sentences where each pair has the same word but it holds a different meanings.The result of this experiment visualizes in the table. Then based on the evaluation we selected three layers out of the twelve layers the best, worst, and the average layer. Which will be used in our further experiments to produce Bangla contextual word embeddings.

4.3 Tokenization and Embedding Extraction

We have chosen BERT embeddings for the contextual representations of the words, as was described in the preceding section. We employed a total of three BERT Bangla language models including the BERT pre-trained multilingual model, Kosher Bangle BERT model, and Sagor Sarkar Bangla BERT model of Table 5.1, which recognizes words in Bangla but has a very small word fragment repertoire. As seen in Fig 4.2, we send the sentence into the BERT tokenizer to extract the BERT head word piece embeddings. If the entire word is not found in the dictionary, it will be broken up into many word pieces or sub-words.

The 768-dimensional hidden state vectors of the select layers mentioned in the preceding section for each tokenized word piece are then extracted from the tokenized word pieces using the BERT model forward propagation approach. The difficulty arises when choosing an embedding to represent a word since every word is represented by several word fragments.

Since BERT embeddings are a by-product of self-attention, we hypothesize that any

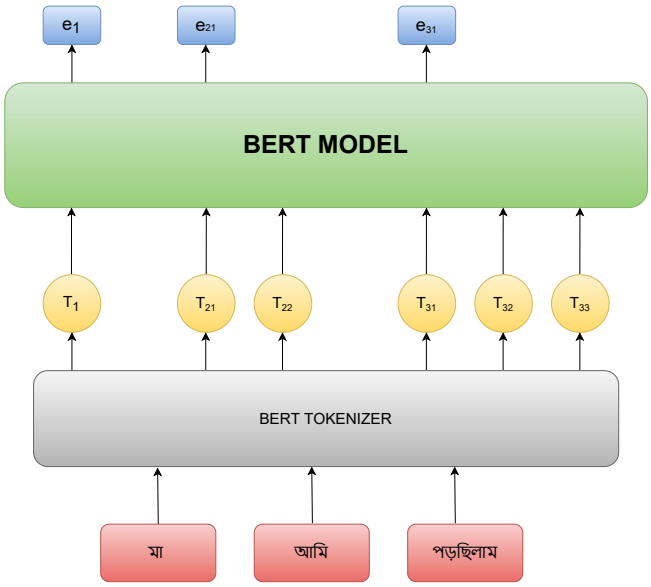

Figure 4.2: BERT Models Embedding Generator Pipeline.

one of the word fragments' embeddings may be sufficient to represent a word. We have decided to choose the embeddings of the headword piece in order to impose consistency. The method is depicted in Fig 4.2. By employing this method, we can make sure that contextual information of each word is preserved in the sentence.

4.4 Knowledge Distillation Approach to Handle Class Imbalance Issue

As discussed in section 4.1, the Microsoft IL-POST dataset has a lot of class imbalance issues. A classifier that is biased in favor of the majority class may be produced by a training set that contains varying numbers of examples from each class. When applied to a test set that is similarly skewed, this classifier provides an optimistic accuracy estimate. To achieve an accuracy equal to the percentage of test cases belonging to the majority class, the classifier may, in the most extreme scenario, assign every test case to the majority class. Due to the possibility of inflated performance estimations caused by an unbalanced dataset, this is a significant problem. This could result in erroneous inferences about the relevance of the algorithm's performance above chance.

There has been worked to handle this data imbalance issue using oversampling strategies. However, we didn't want to increase the data for minority classes; instead, we wanted to address the minority class issue without developing or generating new data by merely using the existing dataset of Microsoft IL-POST. So we had not gone to the oversampling strategy. We have selected Knowlege Distillation to solve this class imbalance issue.

As far as class imbalance issues go, Decision Tree models are less biased than deep learning models. As a result, we first train a decision tree model on our PoS tag dataset to see how it performs for the minority classes. Since it performs better than a deep

learning model for the minority classes, we then apply decision tree knowledge to our deep learning model to enhance its performance for the minority class. This will improve the deep learning model's performance across the board for the Bangla PoS tagging in Microsoft I-POST dataset.

4.5 Training

The Neural Network system in Fig 3.1 was trained with a learning rate of Ir = 0.01 and a weight decay rate of $\beta = 0.00001$, using the Adam optimizer. All of our models were trained using an early halting technique over 100 epochs. During training, we employed the categorical cross-entropy Eq 2 as the loss function initially before distill the knowledge of decision tree.

$$L_{\mathrm{CE}} = -\sum_{i=1}^{n} t_i \log\left(p_i\right) \tag{4.2}$$

for n classes, where t_i is the truth label and p_i is the Softmax probability for the i^{th} class.

We assess the model on our validation partition at the conclusion of each period to determine which model is the best. The following section contains further details regarding our validation metric and model selection criteria.

4.6 Metric and Model Selection Criterion

Due to the imbalanced class situation, we used the macro-F1 score as our assessment metric during the training. On Eq (4.3), the formula for the macro-F1 score is displayed. Where Eq (4.4) (4.5) gives the macro-precision and macro-recall.

$$F1_{\text{macro}} = \frac{\left(\beta^2+1\right) \times \text{ Precision }_{\text{macro}} \times \text{ Recall }_{\text{macro}}}{\left(\beta^2\right) \times \text{ Precision }_{\text{macro}} + \text{ Recall }_{\text{macro}}} \tag{4.3}$$

The β parameter can be adjusted to focus on particular classes in macro-precision and macro-recall. β is set to 1 for all of our tests. We also provide an overall accuracy score for our models Eq (4.6), which helps us determine whether our model is being over-trained or over-fitted.

$$\text{Precision }_{\text{macro}} = \frac{\sum_{i=1}^{C} \frac{tp_i}{tp_i+fp_i}}{l} \tag{4.4}$$

$$\text{Recall }_{\text{macro}} = \frac{\sum_{i=1}^{C} \frac{tp_i}{tp_i+fn_i}}{l} \tag{4.5}$$

$$Acc = \frac{tp+tn}{tp+tn+fp+fn} \tag{4.6}$$

The validation set's macro-F1 scores will serve as the basis for this model selection.

5

Result and Discussion

We primarily compare three things of classical and quantum machine learning model for both of our model.

1. Either classical or quantum model reach the optimum accuracy state faster.

2. Either the classical or quantum model reaches the optimum loss faster.

3. And lastly their test accuracry.

5.1 BERT Layer-wise Comparison

As mentioned in section 4.2 we have built our own custom dataset of a total 30 pair of Bangla sentences where each pair has a same word but it holds different meaning. First, we prepared the bangla sentence for BERT model's format then tokenize it with suitable tokenizer for the selected type of BERT model then pass the tokens of the sentence to input layer of BERT model which gvae us the contextual embeddings for the corresponding words or token. Lastly ,we had find the cos similarity for the target token or word. By evaluiting the BERT layer's performance on cos similarity we have selected the best, average and worst layer for embedding generator for the rest of the research these 3 type layers of each bangla BERT model has been used.

From fig(5.1, 5.2, 5.3) we have evaluated BERT layers and select the best, average and worst layer for each BERT model showed in Table 5.1.

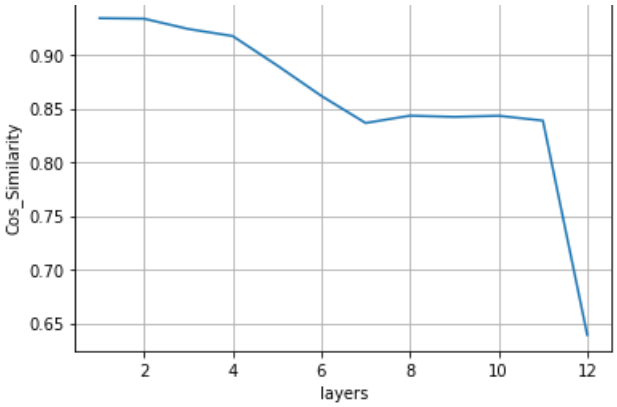

Figure 5.1: Target Word Embedding Cos Similarity Of BERT Multilingual

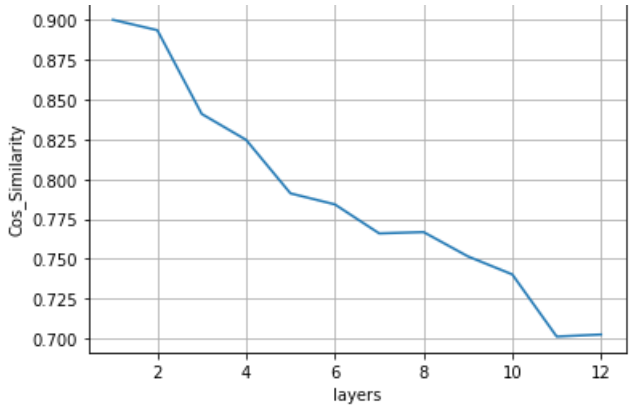

Figure 5.2: Target Word Embedding Cos Similarity Of Sagorsarker/bangla-BERT-base

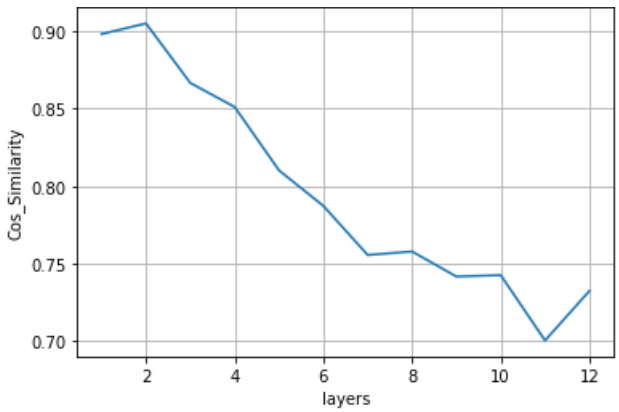

Figure 5.3: Target Word Embedding Cos Similarity Of Kowsher Bangla BERT

Table 5.1: BERT Models Layer Evaluation

Model Name	**Best Layer**	**Average Layer**	**Worst Layer**	**No. Of Embedding Extracted**
BERT-Multilingual	12	7	1	92522
Sagorsarker Bangla BERT	12	7	1	94717
Kowsher Bangla BERT	11	6	1	94717

5.2 Model Evaluation

The research that was carried out using the standard dataset we chose uses a variety of different methodologies. As our benchmark for training, validating, and testing, which is 60:20:20 respectively, we used the Koushhik Roy study. Using their oversampling strategy with this dataset, they were able to achieve a macro-F1 of 87.41

On the testing data of our teacher decision tree model, we present our accuracy and macro F1 score in the Table(5.5 ,5.6, 5.7).

Table 5.2: BERT-Multilingual NN Models Evaluation

Model Name	**Val m-F1**	**Val Acc**	**Test m -F1**	**Test Acc**
BERT-Multilingual 1st Layer	0.54	0.68	0.54	0.68
BERT-Multilingual 7th Layer	0.54	0.69	0.53	0.68
BERT-Multilingual 12th Layer	0.56	0.70	0.56	0.69

On the basis of our validation and testing data for the student neural network model, we present our accuracy and macro F1 score in the Table(5.2, 5.3, 5.4). We get some insights from Table(5.5, 5.6, 5.7, 5.2, 5.3, 5.4). First insight is that for all the decision tree

Table 5.3: Sagorsarker NN Models Evaluation

Model Name	**Val m-F1**	**Val Acc**	**Test m -F1**	**Test Acc**
Sagorsarker 1st Layer	0.60	0.73	0.59	0.73
Sagorsarker 7th Layer	0.70	0.79	0.69	0.79
Sagorsarker 12th Layer	0.70	0.78	0.69	0.78

Table 5.4: Kowsher NN Models Evaluation

Model Name	**Val m-F1**	**Val Acc**	**Test m -F1**	**Test Acc**
Kowsher 1st Layer	0.61	0.74	0.60	0.74
Kowsher 6th Layer	0.69	0.79	0.64	0.79
Kowsher 11th Layer	0.66	0.78	0.65	0.78

model of all BERT embeddings with increasing the layer of BERT models the decision tree's performance decreases but for the neural network models it seems completely reverse means with increasing the BERT layer for embedding generator neural network models perform better. Moreover another insight is that for decision tree model embeddings from Kowsher Bangla BERT got better performance and for neural network model embeddings from Sagor Sarker bought the better result both in terms of Test macro-F1 and Test accuracy.

Table 5.5: BERT-Multilingual DT Models Evaluation

Model Name	**Test Macro-F1**	**Test Acc**
BERT-Multilingual 1st Layer	0.40	0.55
BERT-Multilingual 7th Layer	0.40	0.54
BERT-Multilingual 12th Layer	0.18	0.37

Table 5.6: Sagorsarker DT Models Evaluation

Model Name	**Test Macro-F1**	**Test Acc**
Sagorsarker 1st Layer	0.46	0.60
Sagorsarker 7th Layer	0.34	0.49
Sagorsarker 12th Layer	0.24	0.38

Table 5.7: Kowsher DT Models Evaluation

Model Name	**Test Macro-F1**	**Test Acc**
Kowsher 1st Layer	0.43	0.60
Kowsher 6th Layer	0.32	0.48
Kowsher 11th Layer	0.26	0.41

6

Conclusion

We examine the issue of data imbalance on a mainstream Bangla PoS dataset in this research. We decided to perform knowledge distillation on our training dataset partition utilizing Decision Tress as Teacher and Neural Network model as Student in order to address the imbalance problem. In order to separate the words from the sentences, we have to find contextualized word embeddings using BERT embeddings. The teacher student models were then trained to predict PoS tags on the prevalent Microsoft IL-POST dataset using these separated word embeddings. With this revised formulation of the issue, we were able to apply the well-known knowledge distillation method.

Appendix

A block of sample **python** codes are given below.

```
# Loading the pre-trained BERT model
###################################
# Embeddings will be derived from
# the outputs of this model
model = BertModel.from_pretrained("sagorsarker/bangla-bert-base",
                                  output_hidden_states = True,
                                  )

# Setting up the tokenizer
###################################
# This is the same tokenizer that
# was used in the model to generate
# embeddings to ensure consistency
tokenizer = BertTokenizer.from_pretrained("sagorsarker/bangla-bert-base")

```

```
#Sample function to prepare the text for bert model
def bert_text_preparation(text,tag, tokenizer):
    """Preparing the input for BERT

    Takes a string argument and performs
    pre-processing like adding special tokens,
    tokenization, tokens to ids, and tokens to
    segment ids. All tokens are mapped to seg-
    ment id = 1.

    Args:
        text (str): Text to be converted
        tag (str): Tag associate with text
        tokenizer (obj): Tokenizer object
            to convert text into BERT-re-
            adable tokens and ids

    Returns:
        list: List of BERT-readable tokens
        obj: Torch tensor with token ids
        obj: Torch tensor segment ids


    """
    m1_tokenized_text = []
    marked_text = "[CLS] " + text + " [SEP]"
    m1_marked_text = marked_text.split()

    marked_tag = "[CLS] " +  tag + " [SEP]"
    m1_marked_tag = marked_tag.split()

    tokenized_text = tokenizer.tokenize(marked_text)
    copy_tokenized_text = tokenized_text.copy()

    while tokenized_text:
      token = tokenized_text.pop(0)

      word = m1_marked_text.pop(0)

      if (token == word) :
        m1_tokenized_text.append(token)
```

```
42
43
44          else:
45              word_token = tokenizer.tokenize(word)
46
47              if token == word_token.pop(0):
48                m1_tokenized_text.append(token)
49
50                while word_token:
51                  token = tokenized_text.pop(0)
52                  poped_words = word_token.pop(0)
53
54              else:
55                continue
56
57        while '[UNK]' in m1_tokenized_text:
58          index_of_UNK_Token = m1_tokenized_text.index('[UNK]')
59
60          del_tag =  m1_marked_tag.pop(index_of_UNK_Token)
61
62          del_token = m1_tokenized_text.pop(index_of_UNK_Token)
63
64
65
66
67        indexed_tokens = tokenizer.convert_tokens_to_ids(m1_tokenized_text)
68        segments_ids = [1]*len(indexed_tokens)
69
70        # Convert inputs to PyTorch tensors
71        tokens_tensor = torch.tensor([indexed_tokens])
72        segments_tensors = torch.tensor([segments_ids])
73
74        return m1_tokenized_text, tokens_tensor, segments_tensors
75
```

```python
#Sample function to get bert embeddings
def get_bert_embeddings(tokens_tensor, segments_tensors, model):
    """Get embeddings from an embedding model

    Args:
        tokens_tensor (obj): Torch tensor size [n_tokens]
            with token ids for each token in text
        segments_tensors (obj): Torch tensor size [n_tokens]
            with segment ids for each token in text
        model (obj): Embedding model to generate embeddings
            from token and segment ids

    Returns:
        list: List of list of floats of size
            [n_tokens, n_embedding_dimensions]
            containing embeddings for each token

    """

    # Gradient calculation id disabled
    # Model is in inference mode
    with torch.no_grad():
        outputs = model(tokens_tensor, segments_tensors)
        # Removing the first hidden state
        # The first state is the input state
        hidden_states = outputs[2][1:]

    # Getting embeddings from the final BERT layer
    token_embeddings = hidden_states[-12]
    # Collapsing the tensor into 1-dimension
    token_embeddings_col = torch.squeeze(token_embeddings, dim=0)
    # Converting torchtensors to lists
    list_token_embeddings = [token_embed.tolist() for token_embed in token_embed

    return list_token_embeddings, token_embeddings, token_embeddings_col


```

Bibliography

[1] M. N. Hoque and M. H. Seddiqui, "Bangla parts-of-speech tagging using bangla stemmer and rule based analyzer," in *2015 18th International Conference on Computer and Information Technology (ICCIT)*, 2015, pp. 440–444.

[2] I. Ashrafi, M. Mohammad, A. S. Mauree, G. M. A. Nijhum, R. Karim, N. Mohammed, and S. Momen, "Banner: A cost-sensitive contextualized model for bangla named entity recognition," *IEEE Access*, vol. 8, pp. 58 206–58 226, 2020.

[3] M. N. Hoque and M. H. Seddiqui, "Bangla parts-of-speech tagging using bangla stemmer and rule based analyzer," in *2015 18th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2015, pp. 440–444.

[4] M. F. Kabir, K. Abdullah-Al-Mamun, and M. N. Huda, "Deep learning based parts of speech tagger for bengali," in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*. IEEE, 2016, pp. 26–29.

[5] M. H. Seddiqui, A. Rana, A. Al Mahmud, and T. Sayeed, "Parts of speech tagging using morphological analysis in bangla," in *Proceeding of the 6th International Conference on Computer and Information Technology (ICCIT)*, 2003.

[6] D. Chakrabarti and P. CDAC, "Layered parts of speech tagging for bangla," *Language in India, www. languageinindia. com, Special Volume: Problems of Parsing in Indian Languages*, 2011.

[7] K. Elleithy, *Advances and innovations in systems, computing sciences and software Engineering*. Springer Science & Business Media, 2007.

[8] K. Roy, M. Hasan, K. Fuhad, N. Mohammed, A. Rabby, N. Hasan, J. Nahar, and F. Rahman, "Bangla part of speech tagging using contextual embeddings and oversampling techniques," in *Proceedings of the Future Technologies Conference*. Springer, 2020, pp. 648–663.

[9] *Indian Language Part-of-Speech Tagset: Bengali*.

Index