IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 July 2022, accepted 27 July 2022, date of publication 8 August 2022, date of current version 7 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3197662

RESEARCH ARTICLE

Bangla-BERT: Transformer-Based Efficient Model for Transfer Learning and Language Understanding

M. KOWSHER[1], **ABDULLAH AS SAMI**[2], **NUSRAT JAHAN PROTTASHA**[3], **MOHAMMAD SHAMSUL AREFIN**[3,4], **(Senior Member, IEEE), PRANAB KUMAR DHAR**[4], **AND TAKESHI KOSHIBA**[5], **(Member, IEEE)**

[1]Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ 07030, USA
[2]Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chattogram 4349, Bangladesh
[3]Department of Computer Science and Engineering, Daffodil International University, Dhaka 1207, Bangladesh
[4]Chittagong University of Engineering and Technology, Chattogram 4349, Bangladesh
[5]Waseda University, Shinjuku-ku, Tokyo 169-8050, Japan

Corresponding author: Mohammad Shamsul Arefin (sarefin@cuet.ac.bd)

ABSTRACT The advent of pre-trained language models has directed a new era of Natural Language Processing (NLP), enabling us to create powerful language models. Among these models, Transformer-based models like BERT have grown in popularity due to their cutting-edge effectiveness. However, these models heavily rely on resource-intensive languages, forcing other languages into multilingual models(mBERT). The two fundamental challenges with mBERT become significantly more challenging in a resource-constrained language like Bangla. It was trained on a limited and organized dataset and contained weights for all other languages. Besides, current research on other languages suggests that a language-specific BERT model will exceed multilingual ones. This paper introduces Bangla-BERT,[a] a monolingual BERT model for the Bangla language. Despite the limited data available for NLP tasks in Bangla, we perform pre-training on the largest Bangla language model dataset, BanglaLM, which we constructed using 40 GB of text data. Bangla-BERT achieves the highest results in all datasets and vastly improves the state-of-the-art performance in binary linguistic classification, multilabel extraction, and named entity recognition, outperforming multilingual BERT and other previous research. The pre-trained model is assessed against several non-contextual models such as Bangla fasttext and word2vec the downstream tasks. Finally, this model is evaluated by transfer learning based on hybrid deep learning models such as LSTM, CNN, and CRF in NER, and it is observed that Bangla-BERT outperforms state-of-the-art methods. The proposed Bangla-BERT model is assessed by using benchmark datasets, including Banfakenews, Sentiment Analysis on Bengali News Comments, and Cross-lingual Sentiment Analysis in Bengali. Finally, it is concluded that Bangla-BERT surpasses all prior state-of-the-art results by 3.52%, 2.2%, and 5.3%.

INDEX TERMS Bangla NLP, BERT-base, large corpus, transformer.

I. INTRODUCTION

Pre-trained language models based on the transformer architecture have become an absolute standard for state-of-the-art performance on a wide variety of natural language processing applications [1].BERT, a renowned transformer-based technique, brought a great revolution that had huge impacts on the evolution of NLP [2]. Since its release as an academic research paper, this technologically pioneering NLP model has amazed the AI world. It's the first-ever deeply bidirectional and fully unsupervised technique for language representation that was pre-trained just using a plain text corpus [3]. Numerous advancements are happening

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Oliva.

[a]https://huggingface.co/Kowsher/bangla-bert

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

in BERT nowadays. One notable modification over BERT by Facebook is named ROBERTA, which uses a more robust architecture with massive computational power and an enormous dataset [4]. Another method invented called XLNet was inspired by BERT's autoregressive formation [5]. Both these models require substantial computational power, which becomes a problem for a particular aspect. For this power with less computation, another comprehensive model comes from BERT, compromising only 5% performance degradation named DistilBERT [6]. Despite its small size, it gives a faster performance, and DistilBERT results in almost identical performance on similar tasks. Google announces ALBERT, a lite version of BERT. Even though it has fewer parameters than BERT, it produces significant outcomes [7]. BERT establishes its supremacy over all other language processing units. The authors included a "multilingual" version of BERT(mBERT) pre-trained on the Wikipedia articles of 104 distinct languages, including the Bangla language, to serve as a resource for languages other than English. This renowned variation of BERT emphasizes the contextual representation for several multilingual tasks [8]. This model showed promising results and obtained state-of-the-art performance on cross-lingual benchmarks by optimizing for language-specific tasks. This consequence yields a wave of implementing BERT on monolingual data. The monolingual implementation of the BERT model for a resource-constrained language like Bangla can create a new era of language modeling for Bangla.

The majority of the latest BERT models are only available in English and other resource-rich languages such as Chinese, Arabic, and Spanish. When it comes to low resources like Bangla, it is still at the bottom of the heap, and it leads to the lack of availability of many downstream task datasets and pre-trained language models. Additionally, mBERT, trained in 104 languages, has two significant gaps. It was prepared using only relatively structured and limited language data from Wikipedia, and another being the aggregate weights of all 104 languages. This article has addressed those deficiencies and proposed a monolingual BERT for Bangla language based on a developed large Bangla language dataset (BanglaLM).[1]

In addition, this paper also discusses the process of the pretraining architecture of the BERT transformer model for Bangla, which we refer to as Bangla-BERT. This model has been trained from scratch, and its performance is compared to mBERT and other Bangla pre-trained word embedding models on some published datasets for sentiment analysis, binary and multilabel text classification, and NER. We have developed the largest Bangla language modeling dataset to train the proposed model. The dataset is 40 GB, with three variants containing around 20 million samples for each variant. The proposed model has been trained on a substantial quantity of unsupervised developed data(BanglaLM) before fine-tuning. However, it is initiated using the parameters that have already been trained before utilizing labeled data from downstream tasks. Most of the research in Bangla didn't examine the power of the transformer. Furthermore, none of them used an extensive dataset for any pre-trained model as the resource is constrained.

This work examines the potential by fine-tuning a range of Bangla downstream tasks. The proposed pre-trained model has been compared to a range of non-contextual neural models, including Bangla fasttext[2] (skip-gram and CBOW models) and word2vec. We used some NLP datasets for the proposed BERT model's performance analysis. We compared them with classical machine learning and hybrid deep learning models [9], including LSTM [10], CNN [11], CRF, and proved that the Bangla-BERT model outperformed them. When we compared the outcomes to the current state of the art in performance, Bangla-BERT came out on top [12]. As a summary, our contributions are as follows:

- This work proposes a massive Bangla unsupervised language dataset (BanglaLM) for language modeling.
- This paper presents the whole mechanism for pre-training the context-aware BERT model using BanglaLM.
- This work includes training a language model with the largest dataset ever created for Bangla and exploring the possibility of fine-tuning a transformer model for a low-resource language like Bangla.
- This work resolves the mBERT's limitation for Bangla (trained on limited and more structured data only) and mixed weights issues among 104 languages.
- We examine Bangla-BERT and show its effectiveness on four NLP downstream tasks: Sentiment Analysis, Named Entity Recognition, Binary, and Multilevel Text Classifications. Apart from that, compared to mBERT and other non-contextual models such as Bangla fasttext (including skip-gram and CBOW models), word2vec, in these downstream tasks, we showed that the proposed model outperformed them all by a wide margin.
- We make Bangla-BERT available on popular site Huggingface so that it can be adopted as the new baseline and to advance Bangla NLP research.

The rest of this paper is organized as follows. Section 2 contains a brief overview of the previous research on language representation. The architecture of BERT is described in Section 3, and the method used to construct Bangla-BERT is described in Section 4. Section 5 illustrates the technique for developing vocabulary. The following section describes the downstream NLP tasks and benchmark datasets in-depth and the acquired results. Section 7 details the comparison with the previous work. Section 8 discusses future work, and section 9 concludes this paper.

[1] https://www.kaggle.com/gakowsher/bangla-language-model-dataset

[2] https://github.com/Kowsher/Bangla-Fasttext

II. RELATED WORK

Word2vec [13] has begun the modern era of language processing and was proposed in 2013 to find the most meaningful word representations. As the transition continues, The follow-up of word2vec appears as GloVe [14] and fastText [15]. While none of these models contained context-specific knowledge, ELMo [16] dealt with the issue. A similar LSTM-based model, ULMFit [17], sparked a revolution by including the transfer learning method into NLP tasks. While they perform admirably in polysemy, the input method poses a problem. Since Elmo demands character-based input, UlmFit's word-based input reveals a lack of vocabulary, which BERT resolves through sub-words solutions. ELMo takes a concatenated approach to both directions, but UlmFit operates with a unidirectional approach [18]. BERT outperforms both of these methods due to its bidirectional strategy. However, an identical transformer-based architecture known as GPT with a unidirectional topology emerges. It undergoes three (3) transformations over two years. Though the initial model required fine-tuning, the latest model, the GPT-3, does not. Additionally, Open Ai, the originator of GPT, permits it exclusively in a commercial context through API, but Google revolutionizes the NLP industry by making the BERT an open-source model [19], [20], [21].

When Google published the 104-language mBERT model, it generated great interest. As in 2019, [22], [23] demonstrated the effectiveness of mBERT by experimenting with it on various natural language processing tasks. This phenomenon led researchers to work on the BERT model, which can capture multiple languages. Simultaneously, [24] published a model for cross-lingual BERT that includes 15 languages for cross-lingual signal exploitation. [25] compared monolingual versions of BERT (English and German) to mBERT. It has been stated that mBERT performed more challenging tasks poorly regarding language generation. It is one of the factors contributing to the recent wave of BERT monolingualism. BERT significantly empowers monolingualism. Through its monolingual implementation, this model not only achieves cutting-edge efficiency but also establishes a benchmark in numerous languages by utilizing an attention strategy. Though BERT is English-centric, pushing other languages to resource-constrained multilingual models (mBERT), researchers and scientists work to include this into their language. Their substantial contribution shows CamemBERT [26] and FlauBERT [27] for French, BERTje [28], and RobBERT [29] for Dutch, AraBERT [30] for Arabic, AlBERTo [31] for Italian, PersBERT [32]for Persian, FinBERT [33] for Finnish, and other 30 languages such as Chinese [34], Spanish [35], Romanian [36], Russian [37], etc. Even though these languages are highly diverse, the context-based approach yields promising results in all variants.

The evaluation of self-attention mechanisms departs from conventional recurrent architectures, consisting of token prediction followed by masked language modeling (MLM) [38], [39]. In a masked language modeling (MLM) test Goldberg [39] shows how BERT consistently assigns higher scores to correct verb forms than to incorrect verb forms.NSP (next sentence prediction), a binary classification task that allows the model to capture relationships between phrases easily, is another of BERT's underlying mechanisms [28]. Another robust upgrade of BERT, RoBERTa, eliminates the NSP task from the dynamic masking instead of static masking [4]. Rather than relying entirely on BERT's internal architecture, researchers are increasingly likely to exploit this. For example, while the French and Spanish imply a dynamic masking method, the Chinese take it a step further by implementing a novel whole-word masking method recently developed by the inventor of BERT [26], [34], [35].

Researchers created these models with less difficulty due to their language's abundant resources. However, we encounter many obstacles compared to their work because the Bangla language has limited resources. We address this as well as the mBERT issue of accumulating the weight of additional languages and the problem of limited, organized data.

III. BERT ARCHITECTURE

Since Universal Language Model Fine-tuning (ULMFIT) launched, transfer learning has instantly established the gold standard for state-of-the-art results in NLP-related tasks. Following then, significant advancements have been made by merging the Transformer with transfer learning. OpenAI's GPT and Google AI's BERT are two notable examples of this coupling. The Encoder utilized in BERT is an attention-based Natural Language Processing (NLP) architecture introduced a few years ago in the paper Attention Is All You Need. The paper presents the Transformer architecture, which is formed of two components: the Encoder and the Decoder. Since BERT only employs the Encoder, we will discuss that in this paper. We will look at the Encoder architecture outlined in Attention Is All You Need. Then, in BERT Specifics, we will inquire into the innovative alterations that contribute to the effectiveness of BERT.

A. INPUT EMBEDDING

The input is processed in three stages: tokenization, mapping tokens for numeric representation, and embedding. Following tokenization, each token is mapped to a distinct integer of the corpus vocabulary, referred to as mapping tokens. Each token obtains a unique numeric representation. Besides, padding is required to ensure that the input sequences in a batch are identical in length. Tokenization, mapping, and word embeddings all refer to the process of converting words to vectors, which is similar to how neural word embedding accomplishes it. Given the following toy sentence: "Bangladesh is a beautiful country." To begin, tokenizing it:

"Bangladesh is a beautiful country." Then we get tokens as - ["Bangladesh," "is," "a," "beautiful," "country," "."]

This is proceeded by mapping, in which each token is assigned a unique integer number in the lexicon of the corpus. Such as -

TABLE 1. **The representative matrix Z of dimension (input_length) x (emb_dim).**

$$\begin{bmatrix} Bangladesh \\ is \\ a \\ beautiful \\ country \\ . \end{bmatrix} \begin{bmatrix} < & - & d_{emb\_dim} & - & > \\ 123 & 0.32 & \ldots & 94 & 32 \\ 83 & 34 & \ldots & 77 & 19 \\ 0.2 & 50 & \ldots & 33 & 30 \\ 289 & 432.98 & \ldots & 150 & 92 \\ 80 & 46 & \ldots & 23 & 32 \\ 41 & 21 & \ldots & 74 & 33 \end{bmatrix}$$

[“Bangladesh,” “is,” “a,” “beautiful,” “country,” “.”]→ [34, 90, 15, 684, 55, 193]. Then, for each word in the sequence, we obtain its embedding. Every phrase in the sequence is associated with an embedding (emb_dim) dimensional vector that the model will discover throughout learning. Consider it a vector look-up for each token. The members of those vectors are handled as model parameters and adjusted via back-propagation in the same way that other weights are optimized.

As a result, we search up the vector associated with each token. For example, this is depicted below equation:

$$\begin{aligned} 34 &\rightarrow E[\,34] = [\,123, 0.32, \ldots, 94, 32] \\ 90 &\rightarrow E[\,90] = [\,83, 34, \ldots, 77, 19] \\ 15 &\rightarrow E[\,15] = [\,0.2, 50, \ldots 33, 30] \\ 684 &\rightarrow E[\,684] = [\,289, 432.98, \ldots, 150, 92] \\ 55 &\rightarrow E[\,55] = [\,80, 46, \ldots, 23, 32] \\ 193 &\rightarrow E[\,193] = [\,41, 21, \ldots, 74, 33] \end{aligned}$$

Then we generate a matrix Z of dimension that is: $(input\_length)x(emb\_dim)$ by stacking each of the vectors.It is shown in table 1.

It is critical to note that padding was utilized to ensure that all input sequences in a batch were identical in length. Such that, we lengthen a few of the sequences by including ’pad’ tokens. The sequence following padding for the 9th length will be as: [“ $< pad >$,″ “ $< pad >$,″ “ $< pad >$,″ “$Bangladesh$,″ “is,″ “a,″ “$beautiful$,″ “$country$,″ “.″] → [5, 5, 5, 34, 90, 15, 684, 55, 193]

B. POSITIONAL ENCODING

BERT algorithm gets an advantage by learning positional embedding. The generated sequence of texts is represented as a matrix, although these representations do not consider the fact of a word’s existence in a variety of places. But it needs to be able to change the representational meaning of a word based on its position. Though it is not intended to alter the word’s complete representation; rather, it aims to alter it slightly to encode its placement.

This analysis adopted a strategy of adding numbers between [-1,1] to the token embeddings using non-learnable sinusoidal functions. The remainder of the encoder represents the word slightly differently based on its place (even if it is the same word).

Additionally, the encoder uses the fact that some words are in a given position while additional words are in a different specific position within the same sequence. We want the network to comprehend both absolute and relative positions. In [38], the authors’ choice of sinusoidal functions enables the representation of locations as linear combinations of one another, allowing the systems to learn relevant relationships among token positions.

We add a matrix P with positional encoding to Z to incorporate this information. Then it becomes $P + Z$.

BERT employs a synthesis of sinusoidal functions. In terms of mathematics, the token’s location in the sequence is denoted by I, and the position of the embedding feature is denoted by j.The sinusoidal function is described in the below equation.

$$p_{i,j} = \begin{cases} \sin(\frac{i}{10000^{\frac{j}{d_{emb\_dim}}}}) \;\; if\ j\ is\ even \end{cases}$$
$$\begin{cases} \cos(\frac{i}{10000^{\frac{j-1}{d_{emb\_dim}}}}) \;\; if\ j\ is\ odd \end{cases}$$

More precisely, the positional embedding matrix for a given text P would be as table 2:

This deterministic approach possessed a number of distinct advantages over learned positional representations. For example, the input length parameter can be increased endlessly because the functions can be calculated for any arbitrary place. Additionally, fewer parameters had to be learned. Thus the model could be trained more quickly.

The resulting matrix is $X = Z + P$ and it has the size $(input\_length)x(emb\_dim)$. It is the input of the first encoder block.

C. ENCODER BLOCK

The BERT Encoder is a transformer-based encoding method based on the combination of attention mechanism and a feed-forward neural network. The Encoder consists of multiple encoder blocks stacked on top of one another. Each encoder block comprises two feed-forward layers and a bidirectional self-attention layer [40].

When data passes through encoder blocks, a matrix of dimensions (Input length) x (Embedded dimension) is returned for a given input sequence generating positional information by positional encoding. Mainly these total N blocks of the Encoder are attached to obtain the output. A particular block is responsible for establishing relationships between the input representations and encoding them in the output. The architecture is illustrated in the figure 1.

D. MULTI-HEAD ATTENTION

The Encoder’s architecture is built around multi-head attention. It calculates attention h multiple times using various weight matrices and then concatenates the results [38]. A head is the outcome of each of these parallel computations of attention [12]. The subscript i will be used to signify a particular head and its corresponding weight matrices. Concatenation will occur once all the heads have been computed.This produces a matrix with the dimensions $Input\_Length * x(h * d\_v)$. Eventually a linear layer consisting of the weight

TABLE 2. The positional embedding matrix.

$$\begin{bmatrix}
< & - & d_{emb_{dim}} & - & - & > \\
Bangladesh & \sin(\frac{0}{10000^{\frac{0}{emb_{dim}}}}) & \cos(\frac{0}{10000^{\frac{0}{emb_{dim}}}}) & \sin(\frac{0}{10000^{\frac{2}{emb_{dim}}}}) & \cos(\frac{0}{10000^{\frac{2}{emb_{dim}}}}) & \ldots \\
is & \sin(\frac{1}{10000^{\frac{0}{emb_{dim}}}}) & \cos(\frac{1}{10000^{\frac{0}{emb_{dim}}}}) & \sin(\frac{1}{10000^{\frac{2}{emb_{dim}}}}) & \cos(\frac{1}{10000^{\frac{2}{emb_{dim}}}}) & \ldots \\
a & \sin(\frac{2}{10000^{\frac{0}{emb_{dim}}}}) & \cos(\frac{2}{10000^{\frac{0}{emb_{dim}}}}) & \sin(\frac{2}{10000^{\frac{2}{emb_{dim}}}}) & \cos(\frac{2}{10000^{\frac{2}{emb_{dim}}}}) & \ldots \\
beautiful & \sin(\frac{3}{10000^{\frac{0}{emb_{dim}}}}) & \cos(\frac{3}{10000^{\frac{0}{emb_{dim}}}}) & \sin(\frac{3}{10000^{\frac{2}{emb_{dim}}}}) & \cos(\frac{3}{10000^{\frac{2}{emb_{dim}}}}) & \ldots \\
country & \sin(\frac{4}{10000^{\frac{0}{emb_{dim}}}}) & \cos(\frac{4}{10000^{\frac{0}{emb_{dim}}}}) & \sin(\frac{4}{10000^{\frac{2}{emb_{dim}}}}) & \cos(\frac{4}{10000^{\frac{2}{emb_{dim}}}}) & \ldots \\
. & \sin(\frac{5}{10000^{\frac{0}{emb_{dim}}}}) & \cos(\frac{5}{10000^{\frac{0}{emb_{dim}}}}) & \sin(\frac{5}{10000^{\frac{2}{emb_{dim}}}}) & \cos(\frac{5}{10000^{\frac{2}{emb_{dim}}}}) & \ldots
\end{bmatrix}$$

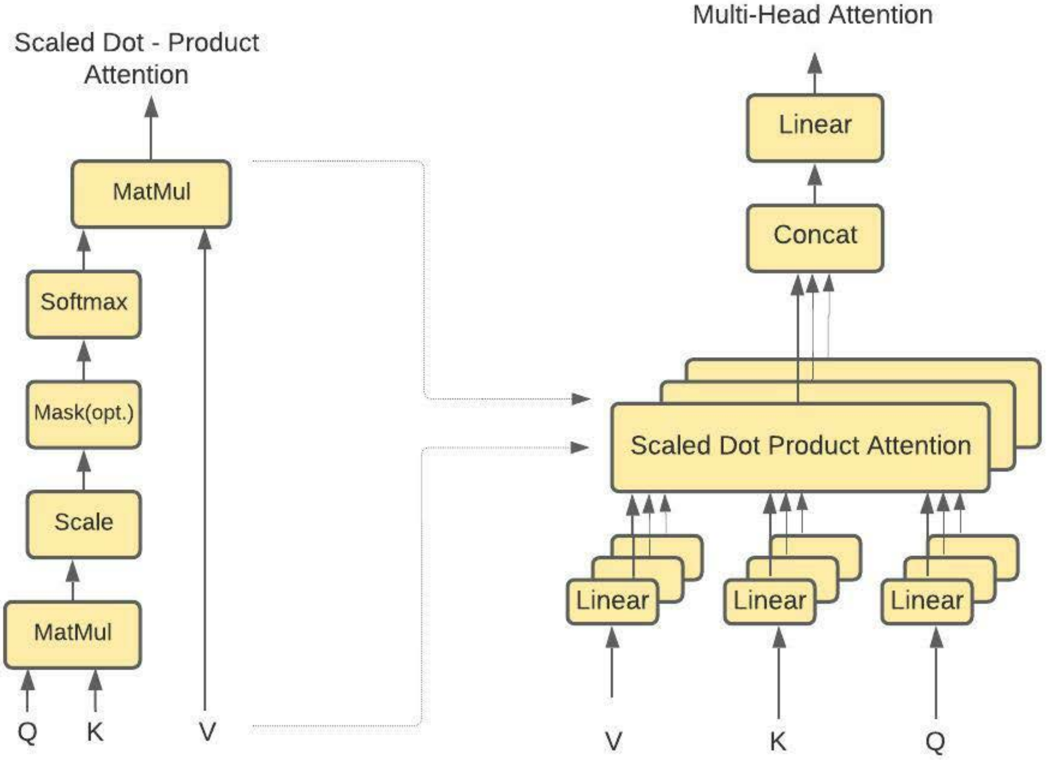


FIGURE 1. Multi-Head Attention Computation.

matrix W^0 of dimension $(h * d\_v) * Embedding\_dimension$ is added, producing an ultimate output with the dimensions $Input\_Length * Embedding\_dimension$. In terms of mathematics:

$$Multihead(Q, K, V) = Concat(head_1, \ldots head_h)W^0$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$

In this case, Q, K, and V serve as placeholders for various input matrices.

E. SCALED DOT-PRODUCT ATTENTION

At the mechanism of scaled Dot-Product Attention, each head is defined by three distinct projections (matrix multiplications) specified by matrices:

$$\begin{aligned}
&W_i^k \text{ with the dimensions } d_{emb\_dim} \times d_k, \\
&W_i^Q \text{ with the dimensions } d_{emb\_dim} \times d_k, \\
&W_i^v \text{ with the dimensions } d_{emb\_dim} \times d_v
\end{aligned}$$

The input matrix X is projected separately via these weight matrices to compute the head.

$$\begin{aligned}
XW_i^K &= K_i \text{ with the dimensions input\_length } \times d_k \\
XW_i^Q &= Q_i \text{ with the dimensions input\_length } \times d_k \\
XW_i^V &= V_i \text{ with the dimensions input\_length } \times d_v
\end{aligned}$$

We use these K_i Q_i and V_i to determine the scaled dot product attention.

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

Here,The dot product of these K_i and Q_i projections can be used to quantify the similarity of token projections. Considering m_i and n_j as the i_th and j_th token's projections via K_i and Q_i, correspondingly, the dot product is as follows:

$$m_i n_j = cos(m_i, n_j)||m_i||2||n_j||2$$

TABLE 3. Illustration of previous example where decimal positive numbers sum to 1.

$$\begin{bmatrix} & Bangladesh & is & a & beautiful & country & . \\ Bangladesh & 0.1 & 0 & 0.006 & 0.1 & 0.6 & 0.14 \\ is & \dots & \dots & \dots & \dots & \dots & \dots \\ a & \dots & \dots & \dots & \dots & \dots & \dots \\ beautiful & \dots & \dots & \dots & \dots & \dots & \dots \\ country & \dots & \dots & \dots & \dots & \dots & \dots \\ . & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

TABLE 4. V_i's multiplication production.

$$\begin{bmatrix} & Bangladesh & is & a & beautiful & country & . \\ Bangladesh & 0.1 & 0 & 0.006 & 0.1 & 0.6 & 0.14 \\ is & \dots & \dots & \dots & \dots & \dots & \dots \\ a & \dots & \dots & \dots & \dots & \dots & \dots \\ beautiful & \dots & \dots & \dots & \dots & \dots & \dots \\ country & \dots & \dots & \dots & \dots & \dots & \dots \\ . & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} d_v \\ v_{Bangladesh} \\ v_{is} \\ v_a \\ v_{beautiful} \\ v_{country} \\ v_. \end{bmatrix}$$

TABLE 5. A matrix in which each row is a composition of the token's representations projected via V_i.

$$\begin{bmatrix} -< & - & - & d_v & - & - & -< \\ Bangladesh+0.1v_{Bangladesh} & +0v_{is} & +0.06v_a & +0.1v_{beautiful} & +0.6v_{country} & +0.14v_. & \\ is & \dots & \dots & \dots & \dots & \dots & \dots \\ a & \dots & \dots & \dots & \dots & \dots & \dots \\ beautiful & \dots & \dots & \dots & \dots & \dots & \dots \\ country & \dots & \dots & \dots & \dots & \dots & \dots \\ . & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

It denotes the similarity in direction between n_i and m_j. Following this, the matrix is scaled by dividing it element-wise by the square root of d_k. The next stage involves implementing softmax row-by-row. As a result, the row value of the matrix converges to a value between 0 and 1, which sums it to 1. Lastly, V_i multiplies this result to get the head [3].

Considering our dummy example: Bangladesh is a beautiful country. Then, the resulting representation of "Bangladesh" could look something like table 3.

Then multiplying this by v_i we get the outcome of table 4.

This generates a matrix in which each row is composed off the token's representations projected via V_i showing at table 5.

A unique head here symbolizes the cohesion of "Bangladesh" and "country." We can calculate this h amount of times (h heads) where each encoder block is required for storing these different relationships. Taking the earlier case as the first head.

$$\begin{aligned} V_{Bangladesh,1} = {} & 0.1v_{Bangladesh} + 0.0v_{is} + 0.06v_a \\ & + 0.1v_{beautiful} + 0.6v_{country} + 0v \end{aligned}$$

At this stage, "Bangladesh" would be represented as

$$Concat(V_1, V_2, V_3, \dots, V_h)W_0$$

Concatenating h weighted variations of token expressions using h distinct learned projections yield the token representation.

The following layers comprise the position-based Feed Forward Network. Such that, for every row in the preceding layer's output.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

where W_1 and W_2 are $(emb\_dim)x(d\_F)$ and $(d\_F)x(emb\_dim)$, respectively. Token vector representations do not "interact" with one another. It is equal to performing the computations row by row and then stacking the rows in a matrix. This step's output has the dimensions $(input\_length)x(emb\_dim)$.

The output of this step is then passed to the dropout, add, and norm layers. Between position-aware feed-forward networks and dropout, add, and norm networks, there is always a layer named sublayer. A sublayer is a layer with identical inputs and outputs (Multi-Head Attention or Feed-Forward). Dropout is applied with a 10% probability following each Sublayer. This is referred to as Dropout(Sublayer(x)).

This result is applied to the input x of the Sublayer, yielding $x + Dropout(Sublayer(x))$

This is accomplished in the Multi-Head Attention layer by supplementing the representation of a token x with its original representation based on its relationship to other tokens.

Finally, using the mean and standard deviation for every row, a token-wise/row-wise normalization is constructed. This increases the network's stability.

These layers produce the following:

$$LayerNorm(x + Dropout(Sublayer(x)))$$

This is the architecture that underpins all of the magic in cutting-edge NLP.

IV. METHODOLOGY

Bangla, the seventh most widely spoken language globally, continues to be resource-constrained, resulting in a shortage of downstream task datasets and pre-trained language models. Hence, this paper will revolutionize Bangla NLP by adopting BERTs monolingualism, leaving behind the multilingual phenomenon implemented on a restricted subset. We, therefore, introduce a pre-trained BERT model (Bangla-BERT) for Bangla natural language processing. Bangla-BERT is an optimized BERT variant that achieves state-of-the-art performance in Bangla NLP downstream tasks. On a large-scale Bengali corpus, it is highly compatible with the Bengali word dimension and lexicality. Having a similar model architecture as BERT, we execute additional pre-processing actions to ensure that the architecture easily fits within our massive Bangla Corpora. The presented methodology consists of five main tasks or two phases. The first two tasks, or the first phase, comprise collecting and processing data relating to the dataset. The following three tasks focus on model architecture, including training setup, parameter estimation, and model training.

A. DATA COLLECTION

The initial step in training the Bangla-BERT model is to build a suitable unlabeled text corpus. Since BERT is a transformer-based mechanism, it needs a huge corpus for perfect training. BERT was initially trained on 3.3 billion words retrieved from the enormous English Wikipedia and the Book Corpus. Since the Bengali Wikipedia dumps are rather modest compared to the English ones, we developed the largest Bangla language modeling dataset to resolve this issue. It is an enormous corpus of internet sources, including news, web discussion, blog sites, government journals, TED Talks, subtitles, newspapers, articles, and an internet crawl to generate a sufficiently large and unannotated corpus for pre-training. Consequently, the dataset contains recent news articles from various prominent Bangla newspapers, including Prothom Alo, BD News, Jugantor, and Jaijaidin. Table 6 contains data samples.

B. DATA PROCESSING

It is essential to have a high-quality and structural Bangla corpus to train Bangla-BERT. Consequently, We made structural to BanglaLM from raw data as the conduction of this work. The BanglaLM dataset is available in three variants: raw, pre-processed V1, and pre-processed V2. While the raw version can be pre-processed to meet the requirements of any specific task, we used Preprocessed V1 for pre-training the model and Pre-processed V2 for fine-tuning. The exact size of the whole dataset is 39 GB, including 3 versions, V1 and V2 variants, each containing approximately 20 million observations. In addition, the training corpus consists of around 821 million words and 1.7 million unique words. Mainly the text data in strings of varying lengths were dealt with. An intense cleaning and filtration process have been employed for each subcorpus. Moreover, the noise, emoticons, URL tags, HTML tags, and all the non-meaningful stuff such as telephone/fax numbers, email addresses, and so on have been eliminated. Any advanced linguistic operation like Stemming and lemmatization has not been applied to the training. Since BERT is context-based and has syntactic abilities, changing words to root words by these operations (lemmatization, stemming) reduces the syntactic abilities and context. All foreign languages from the dataset except English were removed because their attendance has less than 0.01% and had no meaningful impact. Punctuations have not been removed in the pre-processed V1 rather than the V2 since it aids in recognizing the word relation. Additionally, it has been ensured that all sentences adhere to a minimum and maximum word length by applying a minimum of 3 and a maximum of 512 as a threshold. Table 7 summarizes the dataset's properties before fitting into the pre-trained model v2.

C. TRAINING SETUP

The monolingual BERT procedure is nearly identical in all languages. The pre-training process begins with forming a vocabulary based on the available corpora. Then, Byte-pair-encoding (BPE) is mainly used to produce cased and uncased vocabulary. Proper execution of these steps considerably improves the model's performance. In addition, the model works better if sentences are tokenized (i.e., the fewer parts each word is split into), as tokenized sentences are more accurate [29]. Our pre-training process is divided into two essential activities. The first is masked language modeling, whereas the second is next sentence prediction. We have used Cross-entropy loss to train a Masked Language Model (MLM) for predicting random masked tokens. Given N tokens, 15% of them are randomly chosen for this purpose. These are derived from 80 percent of selected tokens are replaced with an exclusive [MASK] token, 10% with a random token, and 10% remain untouched.

Our process is depicted in figure 2.

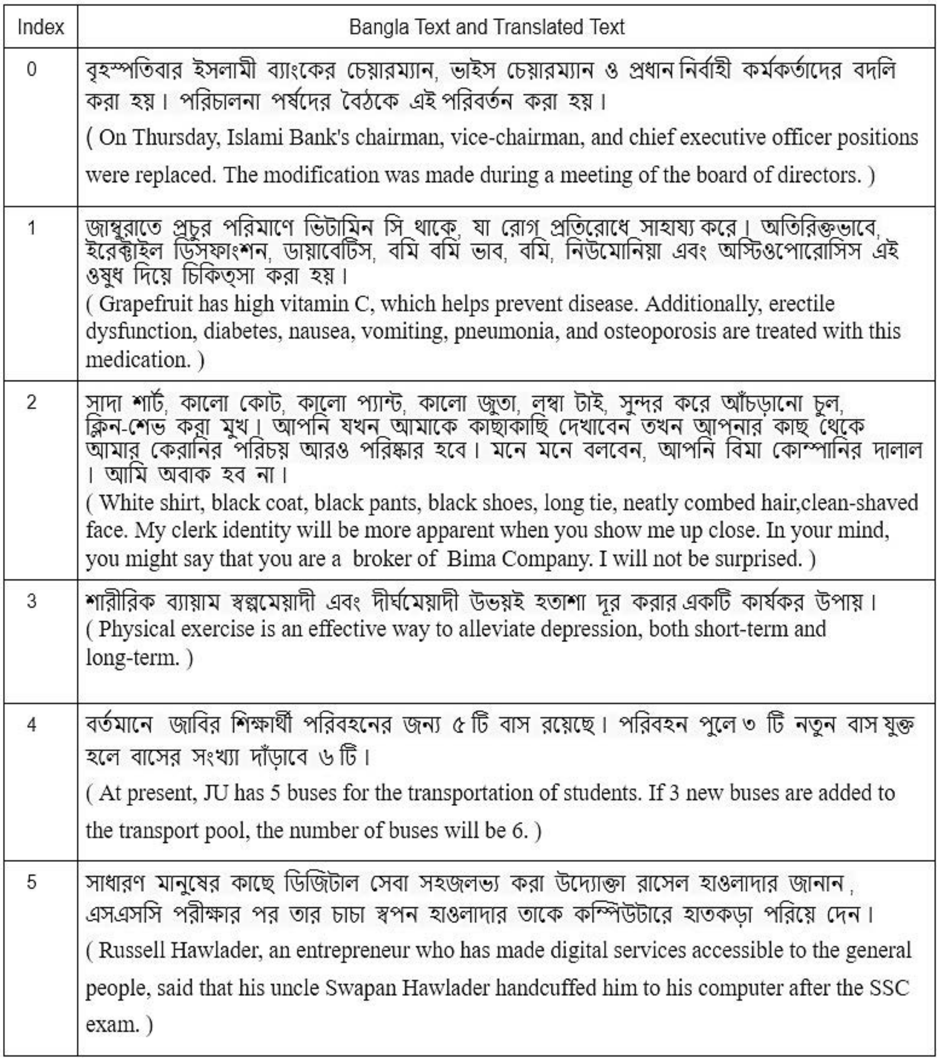
D. PARAMETER ESTIMATION

Our model's setting is critical to obtaining the desired output. That is why we carefully select the model configuration value. The size of the feedforward layer, or intermediate size, is 3072. We have set the pad token id to 0. The encoder and pooler's non-linear activation function (function or string) is gelu. The standard deviation of the truncated normal initializer used to initialize all weight matrices is 0.02. We have set the use_cache to True to indicate whether or not the model must supply the model's most recent key/value attention. We detail each parameter and its value. The whole parameter estimation or model configuration is presented in the table 8.

E. MODEL TRAINING

Our model is based on the BERT architecture, and in the training setup, this work mainly uses the original BERT

TABLE 6. Data samples from BanglaLM corpus that used as input in the training bangla-BERT model.



Index	Bangla Text and Translated Text
0	বৃহস্পতিবার ইসলামী ব্যাংকের চেয়ারম্যান, ভাইস চেয়ারম্যান ও প্রধান নির্বাহী কর্মকর্তাদের বদলি করা হয়। পরিচালনা পর্ষদের বৈঠকে এই পরিবর্তন করা হয়। (On Thursday, Islami Bank's chairman, vice-chairman, and chief executive officer positions were replaced. The modification was made during a meeting of the board of directors.)
1	জাম্বুরাতে প্রচুর পরিমাণে ভিটামিন সি থাকে, যা রোগ প্রতিরোধে সাহায্য করে। অতিরিক্তভাবে, ইরেক্টাইল ডিসফাংশন, ডায়াবেটিস, বমি বমি ভাব, বমি, নিউমোনিয়া এবং অস্টিওপোরোসিস এই ওষুধ দিয়ে চিকিত্সা করা হয়। (Grapefruit has high vitamin C, which helps prevent disease. Additionally, erectile dysfunction, diabetes, nausea, vomiting, pneumonia, and osteoporosis are treated with this medication.)
2	সাদা শার্ট, কালো কোট, কালো প্যান্ট, কালো জুতা, লম্বা টাই, সুন্দর করে আঁচড়ানো চুল, ক্লিন-শেভ করা মুখ। আপনি যখন আমাকে কাছাকাছি দেখাবেন তখন আপনার কাছ থেকে আমার কেরানির পরিচয় আরও পরিষ্কার হবে। মনে মনে বলবেন, আপনি বিমা কোম্পানির দালাল । আমি অবাক হব না। (White shirt, black coat, black pants, black shoes, long tie, neatly combed hair,clean-shaved face. My clerk identity will be more apparent when you show me up close. In your mind, you might say that you are a broker of Bima Company. I will not be surprised.)
3	শারীরিক ব্যায়াম স্বল্পমেয়াদী এবং দীর্ঘমেয়াদী উভয়ই হতাশা দূর করার একটি কার্যকর উপায়। (Physical exercise is an effective way to alleviate depression, both short-term and long-term.)
4	বর্তমানে জাবির শিক্ষার্থী পরিবহনের জন্য ৫ টি বাস রয়েছে। পরিবহন পুলে ৩ টি নতুন বাস যুক্ত হলে বাসের সংখ্যা দাঁড়াবে ৬ টি। (At present, JU has 5 buses for the transportation of students. If 3 new buses are added to the transport pool, the number of buses will be 6.)
5	সাধারণ মানুষের কাছে ডিজিটাল সেবা সহজলভ্য করা উদ্যোক্তা রাসেল হাওলাদার জানান , এসএসসি পরীক্ষার পর তার চাচা স্বপন হাওলাদার তাকে কম্পিউটারে হাতকড়া পরিয়ে দেন। (Russell Hawlader, an entrepreneur who has made digital services accessible to the general people, said that his uncle Swapan Hawlader handcuffed him to his computer after the SSC exam.)

configuration and techniques. This task ensures that the configuration setup produces the same performance as the main BERT by keeping the parameters near the original implementation. We frequently setups the original implementations and hyperparameters value. In some cases, we have made modifications to our required model. Here, the settings are 12 encoder blocks, 768 dimensions, and 12 attention heads comprise our model. It contains a vast vocabulary of 102k, nearly three times the size of the original BERT. This phenomenon has made the model more robust and computationally challenging, even though it yields a meaningful result. Our foundation is built on hugging face transformer version 4.2.2. We have set the value 1e-12 to the denominator for numerical stability in the normalization layers. We have selected the gradient checkpoint false to conserve memory due to the slower backward pass. For attention probabilities and all fully connected layers in the embeddings, encoder, and pooler, we have the layer drop rate at 0.1. We have optimized the objective function using the adaptive moment estimation (ADAM) optimizer [41], which is well-suited for cases involving a large amount of data or parameters, as BERT [3] demonstrate. The learning rate as 1e-6, $\beta 1 = 0.900$ and $\beta 2 = 0.999$ was chosen and 1e-6 epsilon was used for numeric stability.

Pre-training was conducted entirely on Google's Cloud TPU V3, and it took 120 hours to complete the phase.

V. VOCABULARY BUILDING

Tokenization is breaking down a phrase, sentence, paragraph, or even an entire text document into small chunks called tokens. Tokens are mainly instances of a linguistic unit in speech or writing, instead of the type or class of linguistic unit of which they are an occurrence. Sub-word tokenization is the most effective approach among several tokenization techniques because it tackles the Out Of Vocabulary (OOV) problem and considerably reduces the number of model parameters. Mainly it is based on the principle that frequently recurring terms should be included in the vocabulary, while

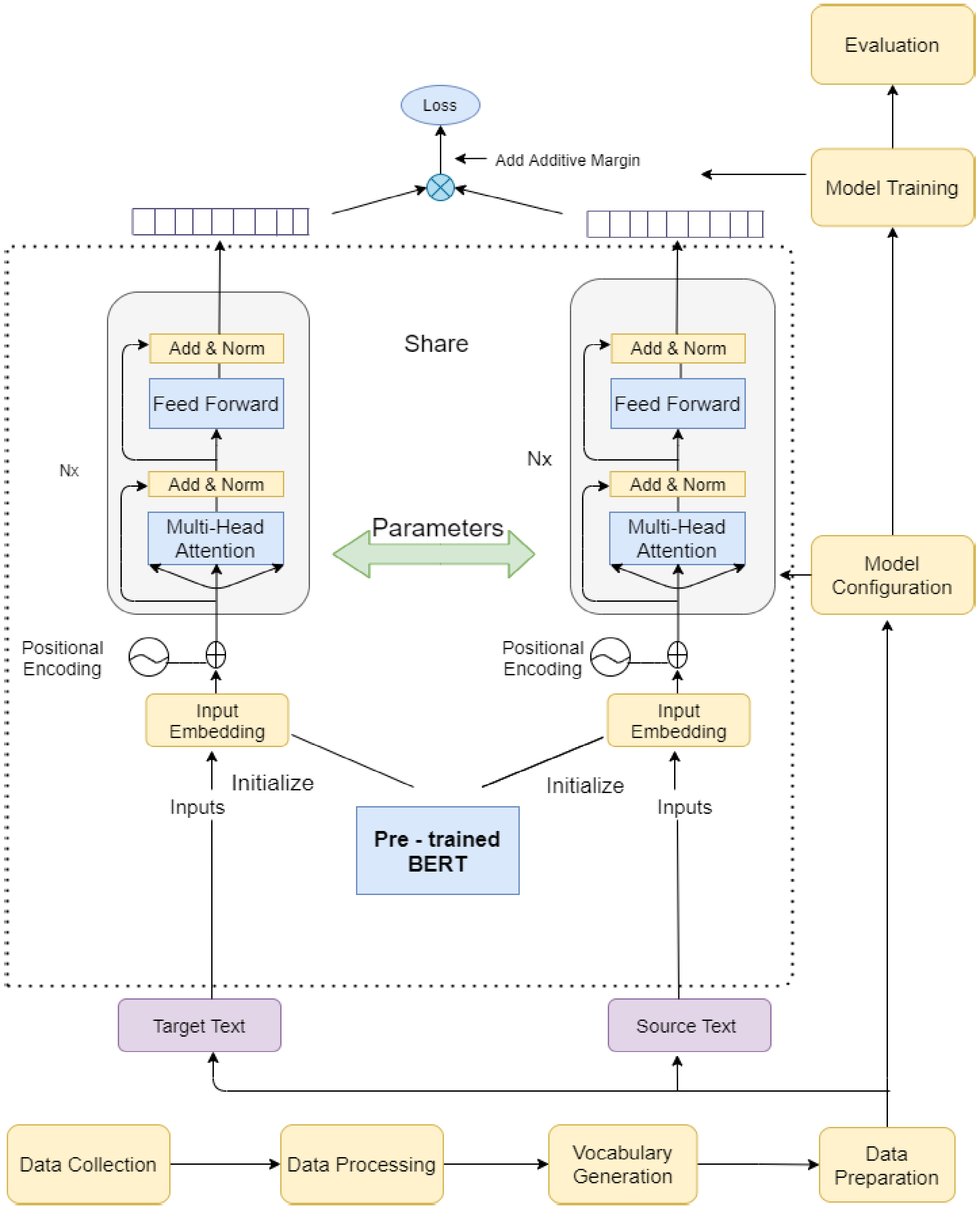


FIGURE 2. Encoder's Structure of Bangla-BERT and weight sharing mechanism. The right encoder is to develop the Bangla-BERT pre-training model using BanglaLM unsupervised dataset. The left encoder accepts the trained parameters from the pre-trained model(right encoder) and is used as fine-tuning for downstream tasks.

uncommon words should be divided into repeated sub-words. There are several methods of sub-word tokenization, and one of them is the word piece tokenizer [42]. In the instance of the proposed model, this work employs the wordpiece tokenizer.

WordPiece begins by incorporating all characters and symbols into its base vocabulary. After establishing the desired vocabulary size, the strategy is to continue inserting subwords until the intended vocabulary size is obtained. WordPiece picks the one representing the maximum probability of the training data while expanding the vocabulary. Additionally, WordPiece determines the frequency of appearance of individual symbols and integrates them into the vocab depending on the count below [43].

$$Count(x, y) = frequency\ of\ (x, y)/frequency(x) * frequency(y)$$

The symbol pair with the highest count will be selected for incorporation into the vocab. Whenever a pair is introduced to the vocab, the model is retrained with the new vocabulary.

TABLE 7. The pre-processing steps and property of the data-set before fitting into pre-trained model V2.

Attribute	Action
Total sentences	1,99,25,396
Min sentence length	3
Max sentence length	512
Total words	82,10,07,301
Unique words	17,10,431
Total char length	5,36,77,01,734
Noise	No
Emoticon	No
URL tag	No
HTML tag	No
Punctuation	No
Stop words	Yes
Stemming	No
Lemmatization	No

This procedure is conducted till the required vocab is attained. This procedure is conducted till the required vocab is attained.

VI. EVALUATION AND RESULT

We assessed Bangla-BERT on four downstream tasks for Bangla language comprehension and these are cross-lingual sentiment analysis, named entity recognition, binary Text Classification, and multi-class sentiment analysis. In addition, we have evaluated Bangla-BERT to the multilingual variant of BERT, including other enhanced neural techniques such as fasttext [44], word2vec [45] for findings for each task as a baseline.

A. EVALUATION METRIC

1) ACCURACY

Accuracy is a performance parameter for machine learning classification models that is defined as the proportion of true positives and negatives to the total number of positive and negative observations. In other words, accuracy is the proportion of times we anticipate our machine learning model to predict a result correctly out of the total number of times it has made predictions. Mathematically, it defines the ratio between the sum of all true positives (TP) and true negatives (TN).

$$Accuracy\ Score = \frac{(TP + TN)}{(TP + FN + TN + FP)}$$

2) PRECISON

The model precision score estimates the proportion of accurately predicted positive labels. Precision is also referred to as the predictive value of the positive. It indicates the mathematical ratio of true positives to the sum of true positives and false positives (FP).

$$Precision\ Score = \frac{TP}{(FP + TP)}$$

3) RECALL

The score for model recall reflects the model's ability to correctly forecast the positives from the actual positives. It depicts the mathematical ratio of true positives to the sum of true positives and false negatives (FN).

$$Recall\ Score = \frac{TP}{(FN + TP)}$$

4) F1 SCORE

F1 score indicates the model score as a function of the recall and precision scores. F1 score is an alternative to Accuracy metrics that provides equal weight to both Precision and Recall when analyzing the performance of a machine learning model in terms of accuracy. It can be mathematically expressed as a harmonic mean of precision and recall score.

$$F1\ Score = \frac{2 * Precision\ Score * Recall\ Score}{(Precision\ Score + Recall\ Score)}$$

5) AUC

As a summary of the Receiver Operator Characteristic (ROC) curve, the Area Under the Curve (AUC) quantifies the ability of a classifier to differentiate between classes. ROC curve is an evaluation metric for binary classification problems. It is a probability curve that compares the True Positive Rate (TPR) to the False Positive Rate (FPR) at different threshold settings. The greater the AUC, the greater the model's ability to differentiate between positive and negative classifications.

6) HAMMING LOSS

The Hamming loss is the proportion of wrongly predicted labels.

$$H = average(y\_true * (1 - y\_pred) + (1 - y\_true) * y\_pred)$$

Where, y_true is the actual labels, and y_pred is the probability.

B. NAMED ENTITIY RECOGNITION

Named Entity Recognition (NER) categorizes various tokens in a text using pre-defined categories. It is structured as a categorization (or tagging) task at the word level, with classes referring to pre-defined groups such as persons, places, institutions, occurrences, and time expressions. While most machine learning approaches have been used previously to solve the Bangla named entity task, including Hidden Markov Model (HMM), Conditional Random Fields (CRF), Support Vector Machine (SVM), Maximum Entropy (ME), and Multi-Engine Method, the BERT approaches have yet to demonstrate their ability.

We have used a dataset created by [46] for this task. This dataset contains train, test sets, and the F1 score is used to assess the effectiveness. Around 96697 tokens from prominent newspapers were used to construct the annotation, 67554 tokens for training purposes, and 29143 words for testing purposes.

TABLE 8. Parameter estimation.

Name	Value	Parameter Description
attention_probs_dropout_prob	0.1	Dropout rate for probability of drawing attention.
gradient_checkpointing"	false	Memory-saving approach as the consequence of a slower backward pass.
hidden_act	gelu	Encoder and pooler's non-linear activation function (function or string).
hidden_dropout_prob	0.1	Likelihood of a layer dropping out in the embeddings, encoder, or pooler.
hidden_size	768	Encoder and pooler layers's dimensionality
initializer_range	0.02	Truncated normal initializer's standard deviation to initialize every weight matrices.
intermediate_size	3072	Dimension of the Transformer encoder's "intermediate" (commonly referred to as feed-forward) layer.
layer_norm_eps	1e-12	Epsilon value utilised by layer normalising layers.
max_position_embeddings	512	Maximum length of a sequence that this model can ever be used with. Usually, this is set to a high value just in case (e.g., 512 or 1024 or 2048).
model_type	BERT	Model type of the architecture
num_attention_heads	12	Number of attention heads in the Transformer encoder's attention layers.
num_hidden_layers	12	Transformer encoder's number of hidden layers.
pad_token_id	0	Token list of IDs for padding purposes such as when batching sequences of varying lengths is done.
position_embedding_type	absolute	Position embedding of a specific type.
transformers_version	4.2.2	Version of the Transformer mechanism
type_vocab_size	2	Vocabulary size of the token_type_ids which is delivered to the model during the call.
use_cache	true	Signifies whether or not the model has to provide last key/value attention.
vocab_size	101975	BERT model's vocabulary size which specifies the number of distinct tokens that can be represented by the inputs_ids supplied to the model when it is invoked.

TABLE 9. Performance of NER. All other models except Bangla-BERT are derived from [46] for comparison.

Model	Precision	Recall	F1
BLSTM+CNN-dropout	0.7700	0.6822	0.7192
BLSTM+CNN+dropout	0.8547	0.6527	0.7190
BLSTM+CNN+CRF	0.7971	0.6541	0.7079
BLSTM-dropout	0.8035	0.6082	0.68.00
BGRU+CNN	0.7332	0.7227	0.7266
BGRU+CNN+CRF	0.8163	0.6380	0.7044
mBERT	0.9587	0.9429	0.9514
Bangla-BERT	**0.9999**	**0.9980**	**0.9995**

The corpus annotations contain eight distinct types of named entities: B-PER (the beginning of a person's name), I-PER (the person's multiword name), B-ORG (the beginning of an institution's name), I-ORG (the inside of a multiword organization's name), B-LOC (the front of a location name), I-LOC (the inside of a multiword location name), TIM (the time expression), and O- (anything other than the categories as mentioned earlier).

We provide the optimal NER approach that produces State-of-the-art (SOTA) outcomes and outperforms all previous methods on this criterion.

The results in Table 9 demonstrate that our model far exceeds all previous work in this domain, obtaining 0.9995 F1 scores. It outperforms the preceding BGRU+CNN model by 0.2659 and the multilingual variant by a factor of 0.0481. mBERT has an F1 score of 0.9514, which is 0.22 percent better than the previous best model. The precision and recall of the mBERT are 0.9587 and 0.9429, respectively, while Bangla-BERT reaches 0.9980 and 0.9999. As a result, the Bangla-BERT model becomes the new state-of-the-art for NER on the Bengali NER corpus.

C. CROSS LINGUAL SENTIMENT ANALYSIS

Sentiment analysis is a sub-field of Natural Language Processing that focuses on examining individual views or emotions about a particular case acquired from various resources [47]. Due to a lack of annotated data and a scarcity of language processing tools, research on sentiment analysis in low-resource languages such as Bangla remains undiscovered. Salim Sazzed [48] produced and annotated an extensive corpus of approximately 12000 Bangla reviews, which became the benchmark for the Bangla sentiment analysis. This corpus has 11807 annotated reviews, including about 2-300 Bengali words per review. All 12000 reviews were categorized as good, negative, or non-subjective. This corpus is unbalanced in content, with 3307 adverse reactions and 8500 favorable ones. The shape of the training data is (9444, 524) in dimension. We chose this dataset for the Sentiment Analysis research.

Table 10 illustrates the sharp distinction in performance between all previous word embedding approaches and the BERT architecture. The Word2vec approach has the lowest effectiveness, with an accuracy of 0.8587 percent and an F1 score of 0.73. It contains a 0.1413 error which is also the maximum among other methods. While the two variations of BanfastText (CBOW, skip-gram) achieve nearly identical accuracy of 0.9441 and 0.9373, respectively, the resultant F1 score is 0.9101 and 0.9134. The multilingual BERT variation performed marginally better than earlier techniques, earning 0.92 accuracies and a 0.93 F1 score. Bangla-BERT significantly surpasses these approaches, reaching 0.9703 accuracies and a 0.9621 F1 score.

The BanfastText indicates a loss of approximately 0.0526, but the word2vec model yields a loss of almost three times the

TABLE 10. Cross-lingual sentiment analysis on mendeley dataset.

Feature Extraction Methods	Algorithm	Accuracy	Hamming Loss	Recall	Precision	F1 score	AUC
Word2vec	KNN	0.8104	0.1897	0.5166	0.7277	0.6042	0.741
	XGB	0.8587	0.1414	0.6178	0.8346	0.7100	0.749
	SVM	0.8396	0.1605	0.5649	0.8043	0.6637	0.746
	RF	0.8006	0.1994	0.5715	0.9897	0.4504	0.724
	LR	0.8396	0.1604	0.8036	0.6811	0.7373	0.729
BanFastText(CBOW)	KNN	0.9001	0.0999	0.7401	0.8844	0.8059	0.821
	XGB	0.9292	0.0707	0.8625	0.8825	0.8724	0.831
	SVM	0.9441	0.0508	0.8403	0.8903	0.9003	0.851
	RF	0.9098	0.0901	0.7673	0.8059	0.8266	0.812
	LR	0.8793	0.1206	0.6419	0.8985	0.7488	0.783
	LSTM	0.9221	0.0752	0.8511	0.9012	0.9101	0.838
	CNN	0.7238	0.2091	0.5203	0.7049	0.6027	0.751
BanFastText (SkipGram)	KNN	0.8927	0.1092	0.6937	0.8796	0.7831	0.796
	XGB	0.9229	0.0773	0.8429	0.8773	0.8697	0.831
	SVM	0.9373	0.0526	0.8836	0.8917	0.8978	0.846
	RF	0.8988	0.1003	0.6752	0.9033	0.8126	0.783
	LR	0.8823	0.1287	0.6042	0.9049	0.8246	0.784
	LSTM	0.9123	0.0701	0.8712	0.9101	0.9134	0.816
	CNN	0.7308	0.2104	0.5813	0.7191	0.6107	0.773
mBERT		0.9491	0.0428	0.9382	0.9489	0.9291	0.899
Bangla-BERT		**0.9703**	**0.0263**	0.9485	0.9785	**0.9621**	**0.939**

BanfastText(CBOW, Skip-gram) value of 0.1414. A decrease in the losses implies an improvement in the classifier's performance. Hence we get a loss value of 0.0263 and the best result from the BERT model. The AUC score is a good measurement of the classifier as it is not biased on the dataset. The word2vec shows 0.749 AUC, whereas the BanfastText(CBOW, skip-gram) shows improvement by 10% and 9% correspondingly. The mBERT and Bangla-BERT models show higher AUC scores of 0.899 and 0.939, respectively. The error is minimum in Bangla BERT, showing only 0.0297, whereas the mBERT shows 0.0509 error.

D. BINARY TEXT CLASSIFICATION ON BANFAKE NEWS DATASET

We have evaluated our model's performance using the Bangla fake news dataset, which comprises 50K Bangla news items and can also construct automated fake news detection systems. This dataset establishes a new baseline in the Bangla language for binary text categorization by considering a broad range of linguistic features. They gathered legitimate news from Bangladesh's 22 most famous and widely reliable news portals. They employed misleading, clickbait, and satirical contexts to arrange all of the content in the dataset under 12 categories, which are further divided into authentic and fake news. There are 48678 accurate news sources out there, yet there are also 1299 sources spreading fake information [49].

The results of fine-tuning the BanfakeNews dataset are shown in table 11. The accuracy of Word2vec is 0.9455, the F1 score is 0.75, and the hamming loss is 0.1814. In addition, it includes a 0.0545 error which is also the maximum among all other techniques. On the other hand, the BanfastText(CBOW) performs considerably better than the Word2vec, with 0.9814 accuracies and a 0.8812 F1 score with 0.1801 hamming loss. The Banfasttext, both CBOW and Skip-Gram demonstrated the same level of accuracy. However, the skip-gram approach produces a higher F1 score of 0.8923, and the hamming loss is nearly equivalent to that of CBOW, containing 0.1802. The mBERT technique has an accuracy of 0.9809 and an F1 score of 0.9201. The hamming loss is 0.1123, more minor than 0.0679 in the previous Skip Gram Model. The Bangla-BERT delivers superior results across all three factors, with an accuracy rate of 0.99.41, an F1 score of 0.9421, and the lowest hamming rate 0f 0.1013. In addition, it shows a 0.0059 error which is the minimum among all other techniques. Bangla-BERT's 0.979 AUC score is the best overall performance among all other methods. Thus, Bangla-BERT became the new state-of-the-art binary text classification model.

E. BANGLA NEWS COMMENT BASELINE(MULTICLASS SENTIMENT ANALYSIS)

The Dataset For Sentiment Analysis On Bengali News Comments is a genuine and trustworthy dataset that is freely accessible to everyone to evaluate various models. For multi-class sentiment analysis, we apply our model to this dataset. The data was gathered from a well-known online news portal called Prothom-Alo, containing 13809 posts. The top ten most often appearing comment topics were used to describe the dataset. Opinions, sports, the Bangladesh economy, entertainment, and technology are just a few examples. The data set is classified into five traditional sentiment categories: strongly positive, positive, neutral, negative, and strongly negative, with each input being tagged three times to assure the data set's validity and reliability. The slightly positive label contains 1436 observations, the positive, neutral, and negative labels have 2279, 2955, and 3936 observations, and the slightly negative label includes 3203 observations. The data set is not biased in any direction. Typical model

TABLE 11. Binary text classification on BanFake news dataset.

Feature Extraction Methods	Algorithm	Accuracy	Hamming Loss	Recall	Precision	F1 score	AUC
Word2vec	KNN	0.8828	0.2671	0.6792	0.7745	0.6821	0.825
	XGB	0.9455	0.1814	0.6789	0.8723	0.7545	0.892
	SVM	0.7678	0.2318	0.7865	0.8124	0.7728	0.726
	RF	0.7213	0.2436	0.5267	0.9013	0.6876	0.696
	LR	0.8013	0.2764	0.6014	0.7067	0.6312	0.646
BanFastText(CBOW)	KNN	0.9321	0.2507	0.7401	0.9421	0.7217	0.884
	XGB	0.9645	0.0274	0.6267	0.9024	0.8245	0.905
	SVM	0.7581	0.2318	0.9003	0.8918	0.7728	0.717
	RF	0.7157	0.2542	0.5073	0.9333	0.7855	0.694
	LR	0.8824	0.2031	0.6419	0.7418	0.6946	0.826
	LSTM	0.9814	0.1801	0.7121	0.8942	0.8812	0.953
	CNN	0.9425	0.1759	0.9412	0.8019	0.7121	0.917
BanFastText(SkipGram)	KNN	0.9664	0.1131	0.6309	0.9060	0.7733	0.937
	XGB	0.9723	0.1105	0.6309	0.9298	0.8517	0.958
	SVM	0.7687	0.2112	0.7012	0.9666	0.8214	0.718
	RF	0.7245	0.2345	0.5623	0.8733	0.6564	0.698
	LR	0.8766	0.1233	0.7421	0.7235	0.7823	0.849
	LSTM	0.9896	0.1802	0.7462	0.8344	0.8923	0.969
	CNN	0.9623	0.1968	0.8504	0.9272	0.8292	0.947
mBERT		0.9809	0.1123	0.9452	0.9625	0.9201	0.968
Bangla-BERT		**0.9941**	**0.1013**	0.9524	0.9783	**0.9421**	**0.979**

TABLE 12. Bangla news comment baseline(multiclass sentiment analysis).

Feature Extraction Methods	Algorithm	Accuracy	Hamming Loss	Recall	Precision	F1 score	AUC
Word2vec	KNN	0.5678	0.4322	0.6620	0.5908	0.5908	0.517
	XGB	0.5953	0.4041	0.7969	0.5953	0.6815	0.535
	SVM	0.6058	0.3942	0.7288	0.6155	0.6674	0.584
	RF	0.5823	0.4175	0.9471	0.5693	0.7111	0.532
	LR	0.5989	0.4013	0.6369	0.6294	0.6326	0.568
BanFastText(CBOW)	KNN	0.6418	0.3581	0.6469	0.6532	0.6499	0.624
	XGB	0.6838	0.3032	0.7495	0.6815	0.7137	0.656
	SVM	0.7033	0.2966	0.7901	0.6966	0.7491	0.683
	RF	0.6803	0.3196	0.8215	0.6663	0.7361	0.656
	LR	0.6295	0.3709	0.6387	0.6646	0.6514	0.605
	LSTM	0.7625	0.2124	0.6667	0.7245	0.6734	0.724
	CNN	0.7411	0.2245	0.5612	0.7876	0.6467	0.715
BanFastText(SkipGram)	KNN	0.6328	0.3672	0.6578	0.6534	0.6435	0.615
	XGB	0.6825	0.3191	0.7583	0.6729	0.7296	0.646
	SVM	0.7195	0.2879	0.8035	0.7829	0.7427	0.701
	RF	0.6712	0.3245	0.8198	0.6721	0.7382	0.663
	LR	0.6287	0.3711	0.6455	0.6712	0.6629	0.608
	LSTM	0.7505	0.2142	0.6845	0.7269	0.6921	0.738
	CNN	0.7327	0.2421	0.5245	0.7427	0.6456	0.715
mBERT		0.7992	0.1942	0.7426	0.7294	0.7621	0.798
Bangla-BERT		**0.8417**	**0.1194**	0.8393	0.8294	**0.8104**	**0.826**

evaluation methods fail to quantify model performance adequately when confronted with unbalanced data sets. The characteristics of the minority class are frequently dismissed as noise. As a result, there may be a considerable risk of the minority class being classified incorrectly compared to the dominant class. There are 248562 words in all, 244432 of which are in Bengali, 4130 of which are in other languages, and the remainder is in numeric language [50].

The following table 12 summarizes the results of multiclass sentiment analysis. As we descend from the word2vec to Bangla-BERT, the accuracy changes regularly. The gap between the earliest feature extraction method and the most recent transfer learning methodology is relatively large, at 0.24. The Word2vec approach produces a 0.71 F1 score with a 0.4175 hamming loss. BanFastText CBOW and BanfastText skip-gram obtain 0.76 and 0.75 accuracy, respectively, whereas Word2vec achieves 0.5824 accuracies. Word2vec comprises 0.4177 error which is also maximum among all other techniques. The F1 scores for BanfastText CBOW and Skip-gram are nearly equal, containing 0.7491 and 0.7427, respectively. CBOW has a hamming loss of 0.2966, while skip gram has a hamming loss of 0.2879. There is a minor improvement in the BanfastText two variants, with roughly 0.74 F1 scores, compared to the Word2vec's 0.71. The BERT's mBERT form has a greater accuracy of 0.7992 and an F1 score of 0.7621. The hamming loss is minimal than all previous methods, which is 0.1942. The BERT approach yields a significantly lower

TABLE 13. Comparison of different models with the dataset.

DataSet	Model	Accuracy	F1 Score
BanFakeNews [49]	L+POS+E(F)+MP		0.9100
	L+POS+E(N)+MP		0.9100
	Bangla-BERT	**0.9925**	**0.9421**
Data Set For Sentiment Analysis On Bengali News Comments [50]	LSTM	0.7474	0.7929
	Bangla-BERT	**0.8417**	**0.8104**
Cross-lingual Sentiment Analysis in Bengali [48]	TextBlob (Unsupervised approach)	0.8279	0.7760
	SVM (Supervised Approach)	0.9300	0..9160
	Bangla-BERT	**0.9703**	**0.9621**

hamming score of as little as 0.1013. Moreover, it outperforms all preceding methods in accuracy and F1 score, obtaining 0.84 and 0.81, respectively. The AUC score is also the highest in Bangla-BERT, comprising 0.826. It is better than the skip-gram LSTM model by 9% and word2vec by 24%. It also shows a 0.1584 error, the minimum among all other techniques.

VII. COMPARISON WITH THE PREVIOUS STUDIES

Table 13 highlights the performance of all state-of-the-art (SOTA) approaches and Bangla-BERT on some of the most renowned datasets on Bangla text classification ever created. For example, for Binary Sentiment analysis on the Banfakenews dataset, the previous best performance is 0.91 F1 when they [49] combine all standard linguistic features with an SVM classifier. However, Bangla-BERT outperforms this with an F1 score of 0.9421.

In the dataset for multiclass sentiment analysis on Bengali News Comments And Its Baseline Evaluation, the previous state-of-the-art result obtained an accuracy of 0.7474 and 0.79 F1 scores using the LSTM model. However, the Bangla-BERT model improves this result by establishing new state-of-the-art results of 0.8417 accuracies and 0.8104 F1 scores for this dataset. Another dataset, Cross-lingual Sentiment Analysis in Bengali Using a New Benchmark Corpus, confirms Bangla-BERT's superiority. The previous best result cannot exceed Bangla-BERT's result. Using the LR classifier in transfer learning and the best-unsupervised technique, TextBlob, yields nearly identical accuracy and F1 scores of approximately 0.82 and 0.78, respectively. However, their supervised method in conjunction with SVM makes an improved performance of 0.93 accuracies and 0.91 F1 scores. Bangla-BERT outperforms all three approaches and can be designated the state-of-the-art model for these datasets in Bangla.

VIII. FUTURE WORK

Using this as an experiment for a resource-constrained language like Bangla, we have illustrated how powerful a pre-trained deep model could be. The following stage will evaluate RoBERTa and other BERT architectures such as DeeBERT, MobileBERT, SpanBERT, and AlBERT in Bangla to strengthen the NLP phenomenon in Bangladesh. Additionally, we intend to propose a high-level API and a Python-defined module so that the developers may access and use the model in various applications. We will examine the usefulness of the Bangla-BERT model across various business applications. Besides, We will explore the encoding of multiple levels of linguistic abstraction within Bangla-BERT to properly comprehend and analyze the model's acquisition of different information. Currently, the majority of users mix Bangla and English in a variety of contexts. However, our model has only been trained in Bangla. Hence our Bangla BERT cannot be used for these applications. In the future, we will combine Bangla and English in datasets to train a model.

IX. CONCLUSION

The emergence of Transformer-based pre-trained language models rapidly expanded the accessibility of high-performing models to the typical user. However, several established multilingual BERT models include Bangla, The only Bangla-specific BERT model known to date trains on minimal website data. We used the most extensive Bangla text corpus to pre-train the language model. This paper efficiently pre-trains the Bangla-BERT model following state-of-the-art BERT architecture. We make it available to the community with the training corpus and evaluation benchmarks. Practitioners from fields other than computer science can fine-tune them for domain-specific downstream tasks. Because of the ease of use of a pre-trained NLP model, its use cases are much broader. By publishing our Bangla-BERT model, we intend to promote deep learning research and applications in Bangla-speaking nations. Additionally, the work will optimize Bangla NLP models in complexity, storage, and processing requirements.

REFERENCES

[1] A. Chernyavskiy, D. Ilvovsky, and P. Nakov, "Transformers: 'The end of history' for NLP?" 2021, *arXiv:2105.00813*.

[2] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: What we know about how BERT works," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 842–866, Dec. 2020.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[5] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XlNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[6] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.

[7] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.

[8] J. Libovický, R. Rosa, and A. Fraser, "How language-neutral is multilingual BERT?" 2019, *arXiv:1911.03310*.

[9] M. Kowsher, I. Hossen, A. Tahabilder, N. J. Prottasha, K. Habib, and Z. R. M. Azmi, "Support directional shifting vector: A direction based machine learning classifier," *Emerg. Sci. J.*, vol. 5, no. 5, pp. 700–713, Oct. 2021.

[10] M. Kowsher, A. Tahabilder, M. Z. I. Sanjid, N. J. Prottasha, M. S. Uddin, M. A. Hossain, and M. A. K. Jilani, "LSTM-ANN & BiLSTM-ANN: Hybrid deep learning models for enhanced classification accuracy," *Proc. Comput. Sci.*, vol. 193, pp. 131–140, Jan. 2021.

[11] M. Kowsher, M. A. Alam, M. J. Uddin, F. Ahmed, M. W. Ullah, and M. R. Islam, "Detecting third umpire decisions & automated scoring system of cricket," in *Proc. Int. Conf. Comput., Commun., Chem., Mater. Electron. Eng.*, Jul. 2019, pp. 1–8.

[12] M. B. Hossain, M. S. Arefin, I. H. Sarker, M. Kowsher, P. K. Dhar, and T. Koshiba, "CARAN: A context-aware recency-based attention network for point-of-interest recommendation," *IEEE Access*, vol. 10, pp. 36299–36310, 2022.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[14] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[15] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[16] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, *arXiv:1802.05365*.

[17] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018, *arXiv:1801.06146*.

[18] J. M. Gomez-Perez, R. Denaux, and A. Garcia-Silva, "Understanding word embeddings and language models," in *Practical Guide to Hybrid Natural Language Processing*. Cham, Switzerland: Springer, 2020, pp. 17–31.

[19] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," Tech. Rep., 2018.

[20] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[21] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, and S. Agarwal, "Language models are few-shot learners," 2020, *arXiv:2005.14165*.

[22] S. Wu and M. Dredze, "Beto, Bentz, becas: The surprising cross-lingual effectiveness of BERT," 2019, *arXiv:1904.09077*.

[23] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?" 2019, *arXiv:1906.01502*.

[24] G. Lample and A. Conneau, "Cross-lingual language model pretraining," 2019, *arXiv:1901.07291*.

[25] S. Rönnqvist, J. Kanerva, T. Salakoski, and F. Ginter, "Is multilingual BERT fluent in language generation?" 2019, *arXiv:1910.03806*.

[26] L. Martin, B. Müller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de la Clergerie, D. Seddah, and B. Sagot, "CamemBERT: A tasty French language model," 2019, *arXiv:1911.03894*.

[27] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, and D. Schwab, "Flaubert: Des modèles de langue contextualisés pré-entraînés pour le français," in *Proc. 6th Conférence Conjointe Journées d' Études Sur La Parole (JEP, 33e édition), Traitement Automatique Des Langues Naturelles (TALN, 27e édition), Rencontre Des Étudiants Chercheurs En Informatique Pour Le Traitement Automatique Des Langues (RÉCITAL, 22e édition)*, vol. 2, 2020, pp. 268–278.

[28] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim, "BERTje: A Dutch BERT model," 2019, *arXiv:1912.09582*.

[29] P. Delobelle, T. Winters, and B. Berendt, "RobBERT: A Dutch RoBERTa-based language model," 2020, *arXiv:2001.06286*.

[30] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," 2020, *arXiv:2003.00104*.

[31] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, and V. Basile, "AlBERTo: Italian BERT language understanding model for NLP challenging tasks based on tweets," in *Proc. 6th Italian Conf. Comput. Linguistics*, vol. 2481, 2019, pp. 1–6.

[32] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, "ParsBERT: Transformer-based model for Persian language understanding," *Neural Process. Lett.*, vol. 53, no. 6, pp. 3831–3847, 2021.

[33] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter, and S. Pyysalo, "Multilingual is not enough: BERT for Finnish," 2019, *arXiv:1912.07076*.

[34] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-training with whole word masking for Chinese BERT," 2019, *arXiv:1906.08101*.

[35] J. Canete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez, "Spanish pre-trained bert model and evaluation data," in *Proc. ICLR*, 2020, pp. 1–10.

[36] M. Masala, S. Ruseti, and M. Dascalu, "RoBERT—A Romanian BERT model," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 6626–6637.

[37] Y. Kuratov and M. Arkhipov, "Adaptation of deep bidirectional multilingual transformers for Russian language," 2019, *arXiv:1905.07213*.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[39] Y. Lin, Y. C. Tan, and R. Frank, "Open sesame: Getting inside BERT's linguistic knowledge," 2019, *arXiv:1906.01698*.

[40] A. Rush, "The annotated transformer," in *Proc. Workshop NLP Open Source Softw. (OSS)*, 2018, pp. 52–60.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[42] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," 2018, *arXiv:1808.06226*.

[43] P. Shi and J. Lin, "Simple BERT models for relation extraction and semantic role labeling," 2019, *arXiv:1904.05255*.

[44] M. Kowsher, M. S. I. Sobuj, M. F. Shahriar, N. J. Prottasha, M. S. Arefin, P. K. Dhar, and T. Koshiba, "An enhanced neural word embedding model for transfer learning," *Appl. Sci.*, vol. 12, no. 6, p. 2848, Mar. 2022.

[45] M. Kowsher, M. J. Uddin, A. Tahabilder, N. J. Prottasha, M. Ahmed, K. R. Alam, and T. Sultana, "BnVec: Towards the development of word embedding for Bangla language processing," *Int. J. Eng. Technol.*, vol. 10, no. 2, pp. 95–102, 2021.

[46] M. J. R. Rifat, S. Abujar, S. R. H. Noori, and S. A. Hossain, "Bengali named entity recognition: A survey with deep learning benchmark," in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–5.

[47] N. J. Prottasha, A. A. Sami, M. Kowsher, S. A. Murad, A. K. Bairagi, M. Masud, and M. Baz, "Transfer learning for sentiment analysis using BERT based supervised fine-tuning," *Sensors*, vol. 22, no. 11, p. 4157, May 2022.

[48] S. Sazzed, "Cross-lingual sentiment classification in low-resource Bengali language," in *Proc. 6th Workshop Noisy User-Generated Text (W-NUT)*, 2020, pp. 50–60.

[49] M. Z. Hossain, M. A. Rahman, M. S. Islam, and S. Kar, "BanFakeNews: A dataset for detecting fake news in Bangla," 2020, *arXiv:2004.08789*.

[50] M. A.-U.-Z. Ashik, S. Shovon, and S. Haque, "Data set for sentiment analysis on Bengali news comments and its baseline evaluation," in *Proc. Int. Conf. Bangla Speech Lang. Process. (ICBSLP)*, Sep. 2019, pp. 1–5.

M. KOWSHER is currently pursuing the Ph.D. degree with the Stevens Institute of Technology. He is also working as an Artificial Intelligence Scientist at Hishab Ltd., and an AI Engineer at NKSoft, USA. He is also working as a Doctorate Research Assistant at the Stevens AI Laboratory, USA. He reviewed many papers in ICCIDM, ICSECS, ICOCSIM, and *Visual Computing for Industry, Biomedicine, and Art*. He also got best paper awards from various international conferences, such as ICONCS, IC4ME2, ICCCM, NISS, and ICIET. Apart from that, he was the champion of Robi r-ventures 2.0 and got the National Basis ICT Award, in 2019. In 2021, he got the Scientist of the year award for his excellent research in the field of AI from IEM, India. He received the Provost Fellowship Award for Ph.D. degree.

ABDULLAH AS SAMI is currently pursuing the B.Sc. degree in computer science and engineering with the Chittagong University of Engineering and Technology, Chittagong, Bangladesh. He is also an Instructor in python and machine learning at a number of prestigious online learning platforms. Additionally, he works as a part-time Freelancer and a Deep Learning Enthusiast. He has worked on a variety of deep learning and natural language processing projects. He has successfully developed numerous novel approaches to machine learning problems, implemented them in production, and boasts shown writing and research abilities that contribute to attaining productivity milestones. His research interests include Bengali language processing, machine translation information retrieval (speech recognition), sentiment analysis and opinion mining, and machine learning algorithms.

NUSRAT JAHAN PROTTASHA received the B.Sc. degree in computer science and engineering from Daffodil International University, in 2021. She is currently working with Data Science Platform as a Research Assistant with several professors. In 2020, she received the Best Paper Award from the International Conference of Cyber Security and Computer Science. Besides, in recognition of scholarly publication in the reputed indexed journal has been awarded for publishing four articles in Scopus journals from her university.

MOHAMMAD SHAMSUL AREFIN (Senior Member, IEEE) received the Doctor of Engineering degree in information engineering from Hiroshima University, Japan, with the support of the scholarship of MEXT, Japan. He is in lien with the Chittagong University of Engineering and Technology (CUET), Bangladesh, and currently affiliated with the Department of Computer Science and Engineering (CSE), Daffodil International University, Dhaka, Bangladesh. Earlier, he was the Head of the Department of CSE, CUET. As a part of his Ph.D. research, he was with the IBM Yamato Software Laboratory, Japan. His research interests include privacy-preserving data publishing and mining, distributed and cloud computing, big data management, multilingual data management, semantic web, object-oriented system development, and IT for agriculture and the environment. He has more than 110 refereed publications in international journals, book series, and conference proceedings. He is a member of ACM, a fellow of IEB, and a fellow of BCS. He is the Organizing Chair of BIM 2021; the TPC Chair, ECCE 2017; the Organizing Co-Chair, ECCE 2019; and the Organizing Chair, BDML 2020. He visited Japan, Indonesia, Malaysia, Bhutan, Singapore, South Korea, Egypt, India, Saudi Arabia, and China, for different professional and social activities.

PRANAB KUMAR DHAR received the B.Sc. degree from the Chittagong University of Engineering and Technology (CUET), Chittagong, Bangladesh, in 2004, the M.Sc. degree from the University of Ulsan, Republic of Korea, in 2010, and the Ph.D. degree from Saitama University, Japan, in 2014. In 2005, he joined as a Lecturer with the Department of Computer Science and Engineering, CUET, where he is currently working as a Professor. He has published over 30 refereed journal articles and 40 conference papers. He is the author of two books, one book chapter, and one patent. His research interests include multimedia security, digital watermarking, steganography, multimedia data compression, sound synthesis, digital image processing, and digital signal processing. He is a member of the technical committee of several international conferences. He serves as a reviewer of various reputed journals, including IEEE, IEICE, Elsevier, and Springer.

TAKESHI KOSHIBA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees from the Tokyo Institute of Technology, in 1990, 1992, and 2001, respectively. He is currently a Full Professor at the Department of Mathematics, Faculty of Education and Integrated Arts and Sciences, Waseda University, Japan. His research interests include theoretical and applied cryptography, randomness in algorithms, and quantum computing and cryptography.