Digital Object Identifier

Training Pruned Language Model

MD KAMRUS SAMAD[1], ANAN GHOSH[2], SAJIB HOSSAIN[3] AND DR. NABEEL MOHAMMED[4]
[1]Department of Electrical and Computer Engineering, North South University (e-mail: kamrus.samad@northsouth.edu)
[2]Department of Electrical and Computer Engineering, North South University (e-mail: anan.ghosh@northsouth.edu)
[3]Department of Electrical and Computer Engineering, North South University (e-mail: sajib.hossain03@northsouth.edu)
[3]Department of Electrical and Computer Engineering, North South University (e-mail: nabeel.mohammed@northsouth.edu)

Corresponding author: Md Kamrus Samad (e-mail: kamrus.samad@northsouth.edu).

ABSTRACT Natural language processing using a large pre-trained model like Bert is expensive, time-consuming, has a large carbon footprint, and is nearly difficult to implement on a machine with minimal CPU capability. By reducing storage requirements and increasing inference computational efficiency without sacrificing comparative performance in different down-streaming tasks, a much greener, resource-saving, and comparatively smaller model can be achieved as well as a larger model. This model can also be a few-shot learner. Large models, such as BERT, need a lot of time and money to train and execute. Furthermore, it is environmentally dangerous. Dropping excess weight is a well-known method for reducing this. A model can do comparably well in various down streaming tasks after losing 90% of their weight.Furthermore, tiny models are few shot learners and may be pruned up to 90% of the network's sparsity rate. This few shot learning model may be trimmed using Lottery Ticket Hypothesis without affecting comparative performance on Bangla NLP tasks.

INDEX TERMS Bangla language processing, Benchmarks, Dataset, Ipet(Iterative Pattern Exploiting Training), Pruning, Text classifications, Token classifications, Transformer Models.

I. INTRODUCTION

NATURAL language processing research has been a matter of concern for future development of technology. The machine learning research community has spent a significant amount of time and effort scaling neural networks to enormous sizes which derive them to left enormous carbon footprint and massive energy consumption. Huge carbon emissions are becoming a threat to both the human race and the environment. Furthermore, large models require a greater amount of computational power, which is prohibitively expensive and nearly impossible to implement in a system with limited resources. Especially in Natural Language Processing (NLP) massive model like BERT having Transformers architectures [1] and pre-trained in self-supervised methods uses huge computational resources. In the race of making the model much greener a new sector of research has looked into the possibility of training smaller subnetworks instead of entire models without sacrificing performance [2]–[7]. This technique for eliminating unnecessary weights more than 90% from neural networks without harming accuracy is known as pruning [8]–[11]. Lottery ticket hypothesis(LTH) [7] is one of the pruning techniques which demonstrate that we could have trained smaller networks from the start if only we had known which subnetworks to choose. Lottery Ticket Hypothesis stated that " A randomly-initialized, dense neural network contains a subnetwork that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations." In this work, we adapt Iterative Pattern Exploiting Training for extremely small pre-trained models by combining multiple task formulations, its resilience to wordings that are hard to understand, its usage of labeled data, and characteristics of the underlying Language Model (LM). This is achieved by converting textual inputs into cloze questions that contain a task description, combined with gradient-based optimization; exploiting unlabeled data gives further improvements. [12].

- Iterative pattern exploiting training on Bangla Language can make the model a few shot learner.
- Pruning using the Lottery Ticket Hypothesis to assess whether such trainable, transferrable subnetworks exist in pre-trained BERT models having 90% sparsity and measure the performance on different down streaming task [15] which is competitive to bigger model as well as much greener and help to solve existing problem.

II. LITERATURE REVIEW

A. ITS NOT JUST SIZE THAT MATTERS: SMALL LANGUAGE MODELS ARE ALSO FEW-SHOT LEARNERS

For our study, we went through the paper "It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners." [12]. In their work, they introduced a method to optimize parameter and language models that can outperform big language models like "GPT3" which needs 175 billion parameters in the different tasks by converting textual inputs into cloze questions that contain a task description, combined with gradient-based optimization and exploiting unlabeled data gives further improvements. They used pattern exploiting training (PET) which combines the idea of reformulating tasks as cloze questions with regular gradient-based finetuning. In the paper, they also addressed PET with multiple tasks. They show that Iterative pattern exploiting training (iPET) in which several generations of models are trained on datasets of increasing size that are labeled by previous generations combined with ALBERT can outperform GPT3 in different tasks requiring only 0.1% of parameter compared to GPT3. They first compute the probability of each token at its position in the cloze question and identify the token with the highest probability then insert this token into the cloze question and compute the probability of the remaining token. They investigate the importance of several factors for few-shot performance: the choice of patterns and verbalizers, the usage of both unlabeled and labeled data, and the properties of the underlying language model. Finally, they measure how choosing different sets of training examples affects performance. It is possible to achieve a few-shot text classification performance similar to GPT-3 on SuperGLUE with LMS that have three orders of magnitude fewer parameters.

B. EARLYBERT: EFFICIENT BERT TRAINING VIA EARLY-BIRD LOTTERY TICKETS

In this paper titled "EarlyBERT: Efficient BERT Training via Early-bird Lottery Tickets". [13]. authors headed to the experiment being influenced by the Early-Bird Lottery Tickets recently studied for computer vision tasks and they proposed EarlyBERT which is a general computationally-efficient training algorithm applicable to both pre-training and fine-tuning of large-scale language models. They identify structured winning tickets in the early stage of BERT training by slimming the self-attention and fully connected sub-layers inside a transformer. Then, they applied those tickets toward efficient BERT training, and conducted comprehensive pretraining and fine-tuning experiments on GLUE and SQuAD downstream tasks. Their proposed EarlyBERT training framework consists of three steps: Searching Stage, the ticket drawing Stage, and the Efficient-training Stage. On the searching stage, they jointly trained BERT and the sparsity-inducing coefficients used to draw the winning ticket, to search for the key substructure they followed the main idea of network similar and modified it so that it can be adapted to pruning BERT by proposing to associate attention heads and intermediate layers of the fully-connected sub-layers in a transformer with learnable coefficients, which will be jointly trained with BERT but with an additional regularization to promote sparsity. In the ticket drawing Stage, they draw the winning ticket using the learned coefficients. They draw EarlyBERT using the learned coefficients with a magnitude-based metric using prune attention heads and intermediate neurons separately. They prune the attention heads whose coefficients have the smallest magnitudes and remove these heads from the computation graph on the Efficient-training stage, they train EarlyBERT for pretraining or downstream fine-tuning. In this experiment for the vanilla BERT, they fine-tune on GLUE datasets for 3 epochs with an initial learning rate of 2 × 10-5, and for 2 epochs on SQuAD with an initial learning rate of 3 × 10-5 also AdamW optimizer for both cases and for pre-training LAMB optimization technique which involves two phases of training has been adapted.

C. LANGUAGE MODEL SIZE REDUCTION BY PRUNING AND CLUSTERING

A different technique for pruning language models is followed in natural language processing. In this paper [14] authors compare different techniques like count cutoff, weighted difference models, Stolcke pruning, and clustering as well as combining them. They draw output by comparing the size reduction of the techniques at the same perplexity as none of the techniques at the same perplexity. They generalize IBM's clustering technique, combined with Stolcke pruning. Their clustering technique creates a binary branching tree with words at the leaves. By cutting the tree at a certain level, it is possible to achieve a wide variety of different numbers of clusters using two different clustering trees. Introduced two models called the "Both Clusters" technique and another model is called "predict clusters". As they described, instead of simultaneous optimization of parameters, they used an alternative technique that is significantly more efficient. This experiment includes and focuses on the fact that when count cutoffs are well optimized, they are better in some cases than previously reported. They have done the first systematic comparison of clustering techniques to pruning techniques, and shown that the pruning techniques outperform the clustering techniques when used separately. And at last, they showed that by combining clustering and Stolcke pruning techniques in a novel way, they achieve significantly better results than using Stolcke pruning alone. Actually, pruning the language model was their focused point of work.

D. A REVIEW OF BANGLA NATURAL LANGUAGE PROCESSING TASKS AND THE UTILITY OF TRANSFORMER MODELS

The author titled the paper [15]"A Review of Bangla Natural Language Processing Tasks and the Utility of Transformer Models" to describe Bangla is still considered a low-resource language in the natural language processing (NLP) field, despite being the world's sixth most frequently spoken language1 with 230 million native speakers. Bangla NLP

(BNLP) is still trailing behind after three decades of study, owing to a lack of resources and the obstacles that come with it. Work in many areas of BNLP is limited; nevertheless, a comprehensive study covering earlier work and current breakthroughs has yet to be completed. The author begins by reviewing Bangla NLP tasks, resources, and tools accessible to the research community; next, we benchmark datasets acquired from multiple platforms for nine NLP tasks using current state-of-the-art algorithms (i.e., transformer-based models). They compare monolingual vs. multilingual models of various sizes to obtain comparative outcomes for the NLP tasks under consideration. They also report our results using both individual and consolidated datasets and provide data split for future research. The authors looked at 108 articles and performed 175 different experiments. Their findings suggest that transformer-based models work well, but that there is a trade-off in terms of computing expenses. They believe that such a comprehensive study would encourage the Bangla NLP community to expand on and develop their research.

E. THE LOTTERY TICKET HYPOTHESIS FOR PRE-TRAINED BERT NETWORKS

For our experiment, we went through the paper "The Lottery Ticket Hypothesis for Pre-trained BERT Networks ." [16]. In this paper, the author described the way of the Lottery Ticket Hypothesis for pre-trained models.Huge pre-trained models like BERT have become the traditional starting point for training on a variety of downstream tasks in natural language processing (NLP), and similar patterns are emerging in other fields of deep learning. Simultaneously, research on the lottery ticket hypothesis has revealed that NLP and computer vision models contain smaller matching subnetworks capable of training to full accuracy in isolation and transferring to other tasks. In this paper, they combine these findings to see if pre-trained BERT models have trainable, transferable subnetworks. They do locate matched subnetworks from 40 percent to 90 percent sparsity for a variety of downstream activities. They discover these subnetworks at (pre-trained) initialization, in contrast to previous NLP research, which found them only after a certain amount of training.Subnetworks discovered on the masked language modeling task (the same task used to pre-train the model) transfer unconditionally; those found on other tasks transfer just partially. Their findings show that the main lottery ticket observations remain applicable in this setting, as large-scale pre-training becomes a more prominent paradigm in deep learning.

F. EMOTION CLASSIFICATION IN A RESOURCE CONSTRAINED LANGUAGE USING TRANSFORMER-BASED APPROACH

According to this paper [34], Bengali texts can be classified into one of six fundamental emotions using a transformer-based classification system. The lack of essential language processing tools and benchmark corpora makes Bengali emotion classification more difficult and more challenging. Utilizing several transformer-based methods (Bangla-BERT, m-BERT, XLM-R), experimentation was conducted. Low resource languages like Bengali and others have made little appreciable development. To categorize emotions in Bengali literature in the lack of a standard corpus, the authors created a corpus named "BEmoC". Consider the impact of several ML, DNN, and transformer-based techniques on the corpus. After preprocessing and annotation, they found 6243 text documents in the BEmoC. The amount of data that is included in BEmoC depends on the sources. To train ML and DNN models, a variety of feature extraction methods are utilized, including TF-IDF, Word2Vec, and FastText. Furthermore, they test the efficacy of transformer-based models in classifying the emotions in the Bengali text. It saw a significant rise in all scores after using transformer-based models. Bangla-BERT has the lowest f1-score 61.91% of the transformer-based models. The XLM-R model significantly outperforms Bangla-BERT and m-BERT by roughly 6% and 5%, respectively. It obtained the highest f1 score of any model, coming in at 69.73%. A performance evaluation of BEmoC revealed that, of all the techniques, the transformer model XLM-R produced the best results. In particular, XLM-R had the highest f1 score, coming in at 69.61%. Transformer-based models can also look at improving the corpus to include texts that contain mixed-emotional or sarcastic language, comparisons, or irony.

G. BANGLABERT: LANGUAGE MODEL PRETRAINING AND BENCHMARKS FOR LOW-RESOURCE LANGUAGE UNDERSTANDING EVALUATION IN BANGLA

In this paper [17], the authors present BanglaBERT, a BERT-based Natural Language Understanding (NLU) model that has been pretrained in Bangla, a language with a large user base but limited resources. In order to pretrain BanglaBERT, they crawl 110 well-known Bangla websites and collect 27.5 GB of data. These authors assess a number of downstream tasks, including text classification and sequence labeling, as well as span prediction, using special downstream datasets on question answering and natural language inference. Additionally, they subjected them to the first-ever Bangla Language Understanding Benchmark as part of this process. The majority of downstream task datasets for NLP applications are in English, so they also pretrain a model in both languages that they call BanglishBERT to allow zero-shot transfer learning between Bangla and English. They also provide two datasets on Question Answering (QA) and Natural Language Inference, two hitherto unexplored tasks in Bangla (NLI). Pre-training was carried out using the basic ELECTRA model, which comprises a 12-layer Transformer encoder with 768 embedding and hidden size, 12 attention heads, 3072 feed-forward sizes, 110M parameters, and 256 batch size for 2.5M steps. Low-resource languages with little data make it difficult to create language-specific models. Therefore, researchers are obliged to use multilingual models for languages lacking reliable pretrained models. They intro-

duced BanglaBERT and BanglishBERT, two NLU models in Bangla, a widely spoken yet low-resource language, to achieve this purpose.

H. BANGLA-BERT: TRANSFORMER-BASED EFFICIENT MODEL FOR TRANSFER LEARNING AND LANGUAGE UNDERSTANDING

The authors introduce Bangla-BERTa, a monolingual BERT model for the Bangla language, in their research study titled "Bangla-BERT: Transformer-based Efficient Model for Transfer Learning and Language Understanding." [35] The largest Bangla language model dataset, BanglaLM, which they constructed using 40 GB of text data, is used for its pre-training. Bangla-BERT outperforms multilingual BERT and other prior studies, achieving the best results across all datasets and significantly enhancing state-of-the-art performance in named entity recognition, multilabel extraction, and binary language classification. Transfer learning based on hybrid deep learning models like LSTM, CNN, and CRF in NER is also employed to assess this model. Additionally, the authors explore the pretraining architecture of the Bangla-BERT (Bangla-BERT) BERT transformer model. When compared to mBERT and other pre-trained Bangla word embedding models, this model, which was created from scratch, performs better. By embracing BERT's monolingualism and abandoning the multilingual phenomena used on a specific subset, this research will revolutionize Bangla NLP. The generation of a vocabulary using the available corpora is the first step in the pre-training process. Then, cased and uncased vocabulary are mostly produced using byte-pairencoding (BPE). They evaluated Bangla-BERT on four downstream tasks for comprehension of the Bangla language: named entity identification, cross-lingual sentiment analysis, binary text classification, and multi-class sentiment analysis. They further compared Bangla-BERT against the multilingual version of BERT, as well as other enhanced neural methods such fasttext and word2vec, to get baseline outcomes for each task.To enable access to and utilization of the model by programmers for a diverse range of applications, they also intend to provide a high-level API and a Python defined module.

I. PRUNING NEURAL NETWORKS WITHOUT ANY DATA BY ITERATIVELY CONSERVING SYNAPTIC FLOW

For our experiment, we went through the paper "Pruning neural networks without any data by iteratively conserving synaptic flow". [36] Due to possible time, memory, and energy savings both during training and during testing, tuning the parameters of deep neural networks has attracted a lot of attention. the presence of winning lottery tickets, through a costly series of training and pruning cycles, or sparse trainable subnetworks at initialization Iterative Synaptic Flow Pruning is a revolutionary pruning algorithm that they introduce (SynFlow). They develop and experimentally test a conservation law that describes layer collapse, which occurs when a layer is prematurely pruned, rendering a network untrainable, at startup in gradient-based pruning methods. Synaptic Iteration Flow The pruning procedure may be seen as conserving, subject to a sparsity requirement, the total flow of synaptic strengths across the network at the initiation. In their research of layer collapse, which is the premature pruning of a whole layer that renders a network untrainable, they develop the axiom Maximal Critical Compression, which holds that a pruning algorithm should prevent layer collapse wherever feasible. Furthermore, they demonstrate that using iterative, positive synaptic saliency ratings, a pruning method completely avoids layer collapse and meets maximum critical compression. By simply eliminating the parameters with the lowest scores, the pruning algorithms they take into consideration will always disguise the parameters. This rating procedure may be used either layer-by-layer or globally across the network. It has been demonstrated empirically that global masking outperforms layer-masking, in part because it adds fewer hyperparameters and permits adjustable pruning rates throughout the network. They experimentally compare the SynFlow algorithm's performance to that of cutting-edge algorithms SNIP and GraSP as well as baselines for random pruning and magnitude pruning. In the high compression regime, SynFlow performs better than the regime (101.5>p) and exhibits more stability as seen by its narrow intervals. In the low compression regime (p<101.5), SynFlow is also highly competitive. Despite the fact that GraSP and SNIP may somewhat beat SynFlow in this regime, both techniques exhibit layer collapse as seen by their abrupt accuracy decreases.The authors of this research proposed a unified theoretical framework to explain layer collapse in the initialization of current pruning methods. They used their concept to explain how layer collapse is solved by repeated magnitude trimming to find the winning lottery ticketinception. SynFlow, which approaches Maximum Critical Compression and demonstrably avoids layer collapse. Despite the fact that their approach is data-agnostic and doesn't require pre-training, they ensure that it consistently matches or beats other algorithms across 12 different combinations of models and datasets.

J. EXPLOITING CLOZE QUESTIONS FOR FEW SHOT TEXT CLASSIFICATION AND NATURAL LANGUAGE INFERENCE

In this paper [37], the author presents a concept that can combine the ideas (providing task descriptions to pre-trained language models and standard supervised training). To help language models in understanding a task, they introduced Pattern Exploiting Training (PET), a semi-supervised training approach that reformulates input examples as cloze-style phrases. Then, a large collection of unlabeled examples are assigned soft labels using these phrases. Finally, standard supervised training is performed on the resulting training set. In low-resource settings, PET significantly outperforms supervised training and powerful semi-supervised approaches for a number of tasks and languages. There are two drawbacks of providing task descriptions to pre-trained language

models, the first drawback of this method is the need large size models and the second one is a limited number of examples that can be provided to the models because the whole input parsed into tokens and unfortunately a number of tokens can not exceed the maximum number. Therefore, the model has already defined how many examples you may give. And here pet turns this situation and promises few-shot learning for normal-sized models and input examples as cloze-style phrases which can define the limitation for a number of examples. When the initial training data set is small, PET outperforms both traditional supervised training and strong semi-supervised approaches.

K. SENTIMENT ANALYSIS FOR BENGALI USING TRANSFORMER BASED MODELS

Internet users are coming forward to evaluate things such books, movies, videos, e-commerce items, etc. and sharing their experiences, which helps the people waiting in line to obtain feedback in advance. This type of text genre captures the core of the sentiment analysis task that aids in polarity classification, or figuring out if a text communicates positive, negative, or neutral feeling. In this paper [38], They investigated at the use of two pre-trained transformer models, multilingual BERT and XLM-Roberta (with fine-tuning), for sentiment analysis for the Bengali language. Three datasets for the Bengali language were used by them for evaluation and to yield promising results. They evaluated these three target datasets and used fine-tuned multilingual BERT and XLM-Roberta to reach the accuracy of 63%-94% and 68%-95%, respectively. This resulted in state-of-the-art results.They began their investigation with baseline models proposed by Islam et al (2020). A multilingual BERT that has been pre-trained in many languages makes up this benchmark model. GRU, LSTM, and CNN are used as extra deep neural network layers on top of BERT in order to create three distinct architectures. They claimed that finetuned XLM-R outperforms this baseline by a significant margin for the "Prothom Alo" dataset when compared to baseline models. The performances of fine-tuned BERT/XLM-R are much better than the closest baselines and will be used as the new benchmark performance for all three datasets.

L. PUNCTUATION RESTORATION USING TRANSFORMER MODELS FOR HIGH-AND LOW-RESOURCE LANGUAGES

In this paper [39], they explored different transformer-based models and propose an augmentation strategy for the Punctuation Restoration task, focusing on high-resource (English) and low-resource (Bangla) languages. They reported that their proposed augmentation strategy yields a 3.8% relative improvement in the F1 score on ASR transcriptions for English. They used bidirectional Long Short-Term Memory on top of the pre-trained transformer network. They obtained a dimensional embedding vector from the pre-trained language model for each token. This is used as input for a BiLSTM layer, consisting of hidden units. This allows the network to make effective use of both past and future contexts for prediction. The outputs from the forward and backward LSTM layers are concatenated at each time step and fed to a fully connected layer with four output neurons, which correspond to 3 punctuation marks and one O token. The accuracy of Automatic Speech Recognition (ASR) systems has increased significantly. This is because state-of-the-art NLP models are mostly trained using punctuated texts. They explored different architectures and finetune pre-trained models for this task focusing on English and Bangla. They observed that for ASR transcriptions, a high proportion of cases Question and Comma are predicted as O and Period. The best result is obtained using the Roberta model with augmentation. For manual (Ref.) and ASR transcriptions, They had gained while merging the number of classes from four to two. They obtained the best result using the XLM-Roberta model as it is trained with more texts for low-resource languages like Bangla and has a larger vocabulary for them. The performance gain from augmentation is marginal on state human labelers to reduce their time and effort and make the manual annotation process faster. For English, they obtained state-of-art results for manual and ASR transcriptions using our augmentation technique coupled with the RoBERTa-large model.

M. STRUCTURED PRUNING OF LARGE LANGUAGE MODELS

By parameterizing each weight matrix using its low-rank factorization and adaptively removing rank-1 components during training, the authors of this paper which is named "Structured Pruning of Large Language Models" [40] provided a generic, structured pruning approach. Additionally, they showed how their technique can be used to prune the BERT model on a number of downstream fine-tuning classification benchmarks as well as adaptive word embeddings in large language models. This approach reduces model size significantly, but it produces unstructured sparse matrices that are hard to support on common hardware, making it difficult to achieve training and inference speedups. Minimizing the expected training loss can be stated as the optimization goal throughout training. They introduced FLOP, an enhanced technique for organized pruning. Using low-rank factorization, FLOP suggests an alternative parameterization of the weight matrices. Pruning each individual parameter weight is a simple solution, but it might be difficult to speed up computations when dealing with unstructured sparse matrices. As a solution, structured pruning opts to eliminate sets of related parameters. The best results are obtained with FLOP -l0 at all pruning settings. At 70% compression, they got a perplexity of 1.25, which is almost the same as the uncompressed model's 1.24. Their findings show that factorization-based pruning significantly beats block-structured pruning and even surpasses unstructured pruning, with the performance being further improved in most situations by utilizing their improved l0 regularization. Their approach only loses 0.8 perplexity while achieving 50% compression. They trained

a large model with hidden size 2048 that contained 90% more parameters and used the NP-l0 baseline, and they were still able to preserve almost 99% of the performance while reducing the number of parameters by 35%. Finally , they presented a generalized structured pruning method based on adaptive low-rank factorization and systematically assessed its performance on large language models. Additionally, they showed that this method can significantly speed up and compress large models with little performance loss when compared to other methods, such as unstructured magnitude pruning.

III. PROPOSED METHODOLOGY

After completing iterative pattern exploiting training up to 90% prune of a model can be done using Pruning technique according to Lottery tickets hypothesis.

A. ITERATIVE PATTERN EXPLOITING TRAINING

We have done our Pattern exploiting training on Bengali dataset using Pre trained model having customized architecture. For this training we followed the technique mentioned on [size doesn't matter]. At this technique, Let M be a masked language model and T it's vocabulary. All token sequence is considered as T^*. For some $z \epsilon T^*$ containing at least k masks and t ϵ T, we denote with q_M^k (t | z) the probability that M assigns to t at the K^{th} masked position in z; the model's logits before applying soft-max are denoted with s_M^k (t | z) task of mapping inputs $x \epsilon X$ to outputs $y \epsilon Y$ and for this PET needs a set of pattern-verbalizer pairs (PVPs). Each PVP p = (P, v) consists of a pattern P : X $\rightarrow$ T^* that maps inputs to cloze questions containing a single mask and a verbalizer v : Y $\rightarrow$ T that maps each output to a single token representing its task-specific meaning in the pattern. The core idea of PET is to derive the probability of y being the correct output for x from the probability of v(y) being the "correct" token at the masked position in P(x). Based on this intuition, a conditional probability distribution q_p of y given x is defined as

$$q_p(y|x) = \frac{exp\ s_p(y|x)}{\sum_{y'} exp\ s_p(y'|x)} \tag{1}$$

Where $s_p(y|x) = s_M^1(v(y)|p(x))$ is the raw score of v(y) at the masked position in P (x). For a given task, identifying PVPs that perform well is challenging in the absence of a large development set. Therefore, PET enables a combination of multiple PVPs P = $\{p_1, \ldots, p_n\}$ as follows: For each PVP p, a MLM is finetuned on training examples (x, y) by minimizing the cross entropy between y and $q_p(y|x)$ and the ensemble of fine tuned MLMs is used to annotate a set of unlabeled examples; each unlabeled example $x \epsilon X$ is annotated with soft labels based on the probability distribution where w_p is a weighting term that is proportional to the accuracy achieved with p on the training set before training. The resulting soft-labeled dataset is used to train a regular sequence classifier by minimizing cross entropy between its output and q_p

$$q_p(y|x) \alpha exp \sum_{p \epsilon P} w_p . s_p(y|x) \tag{2}$$

PET is an iterative technique in which successive generations of models are trained on increasingly large datasets that have been labeled by previous generations. This is accomplished in the following way: First, as in normal PET, an ensemble of MLMs is trained. For each model M_i, a random subset of other models is used to generate a new training set t_i by assigning labels to those unlabeled examples for which the selected subset of models is most confident in its prediction. Each M_i is then retrained on t_i; this process is repeated several times, each time increasing the number of examples in t_i by a constant factor.

B. PRUNING

For pruning we have used the technique of lottery ticket network f(x; , ·), a subnetwork is a network f(x; m , ·) with a pruning mask $m \epsilon \{0, 1\}^{d1}$ (where is the element-wise product). That is, it is a copy of f(x; , ·) with some weights fixed to 0. Let $A_t^T(f(x; \theta_i, \gamma_i))$ be a training algorithm for a task T that trains a network $(f(x; \theta_i, \gamma_i))$ on task T for t steps, creating network $(f(x; \theta_{i+t'}, \gamma_{i+t}))$. Let θ_0 be the BERT-pre-trained weights. Let $\epsilon^T(f(x; \theta))$ be the evaluation metric of model f on task T.

Matching subnetwork. A subnetwork f(x; m , γ) is matching for an algorithm A_t^T if training f(x; m , γ) with algorithm A_t^T results in evaluation metric on task T no lower than training f(x; 0,γ) with algorithm A_t^T.

$$\epsilon^T(A_t^T(f(x; m, \Phi, \theta, \gamma))) \geq \epsilon^T(A_t^T(f(x; \theta_0, \gamma))) \tag{3}$$

Winning ticket: A subnetwork f(x; m ,) is a winning ticket for an algorithm A_t^T if it is a matching subnetwork for A_t^T and = 0.

Universal subnetwork: Universal subnetwork. A subnetwork f(x; m , γt_0) is universal for tasks $\{T_i\}_{i=1}^N$ if it is matching or each $A_{t_i}^{T_i}$ for appropriate, task-specific configurations of γT_i

C. STRATEGIES

We used the method portrayed in Figure 1.First and foremost, using simple transformers with an NLP library and our heterogeneous dataset, we fine-tuned a tiny Bert model named ckiplab/albert-tiny-chines [18] from scratch. We calculated the perplexity for our varied dataset and fine-tuned the model for the down streaming assignment. Following that, for a few shots learning, we train our model using iterative pattern exploitation training, assess the perplexity, and fine-tune for the down streaming job. For ipet, we used this fine-tuned model. We use 22 parameters to apply this approach. We utilize ipet as a method and provide the relevant dataset to the method. Declared the pattern ids as well. Essentially, we constructed a custom data processor and a custom pattern for this task. By providing these arguments for these procedures,

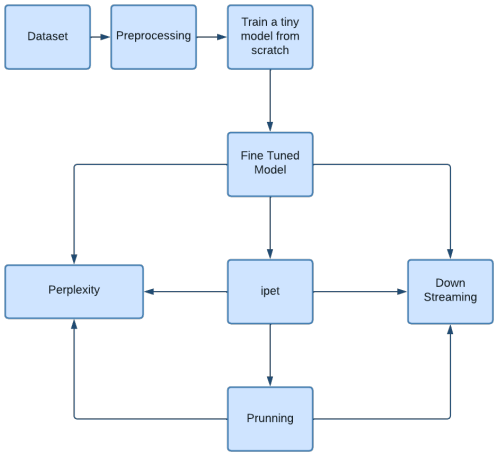


FIGURE 1. Methodology

FIGURE 2. Procedure of Dataset Construct

we also ensure lm training. For pruning our language model we have fine tuned our model on Masked language modeling using our diverse dataset and customized training parameter for exploring the whole iteration in a single pass. After a complete single pass iteration every time 10 percent of the network has been dropped and rewind for the next steps consecutively from the last checkpoint until it reaches 90 percent sparsity rate. We used 32 batch sizes for the pruning and logging outputs after every 5093 steps. We also measured perplexity after iterative pattern exploitation, and then fine-tuned for down streaming tasks for Bangla language model and compared the outcome with other models.

IV. EXPERIMENT

A. DATASET

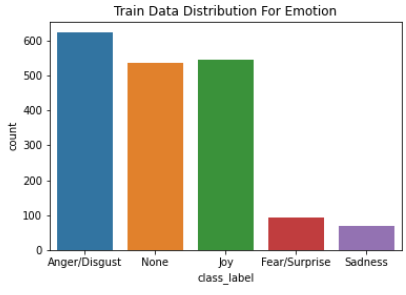
Pretraining a language model requires a large amount of high-quality text data. BERT, for example, is pretrained on 3.3 billion tokens from the English Wikipedia and the Books corpus. However, Bangla is a resource-constrained language. We fine-tuned our pre-trained model using our own corpus compiled from several literature sources. To obtain our data, we collected a number of PDF files and extracted the Bangla text using OCR. We replaced contaminated and useless text with white-space to eliminate polluted and unnecessary content from our data. The data was then prepossessed to BERT format, which is one sentence per line. After prepossessing, the overall size of our corpus is 80 MB, and it contains three types of data. To diversify the data, social media, Bangla newspapers, and Bangla Shadhu text have been mixed. We divided the data for training and testing at an 8:2 ratio and fed it into our model. In Figure 2 the procedure of making a diverse data-set is shown.

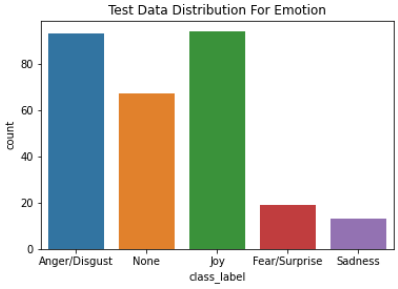
B. DOWNSTREAMING TASKS DATASET

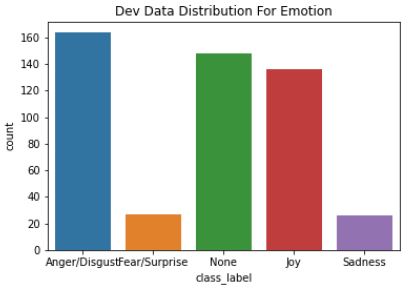
To evaluate the performance of our models, a number of downstreaming tasks are applied. Datasets from this study [16] are used for those downstream tasks. Take into account sentiment, emotion, authorship attribution, and news categorization as a text classification problem, where text might be a document, an article, or a social media post. There are also PoS tagging and punctuation restoration are examples of token classification problems.These models are end-to-end fine-tuned for those downstreaming tasks utilizing the dataset throughout the entire network.

1) Emotion Classification

The dataset of emotion includes five emotion labels: fear/surprise, joy, sadness, anger/disgust, and none [28], [31]. Using the data split method 10% of the dataset specifically for testing. The remaining amount were further divided into 20% for development and 80% for training. 2890 YouTube comments in Bangla, English, and romanized Bangla are included in the dataset.

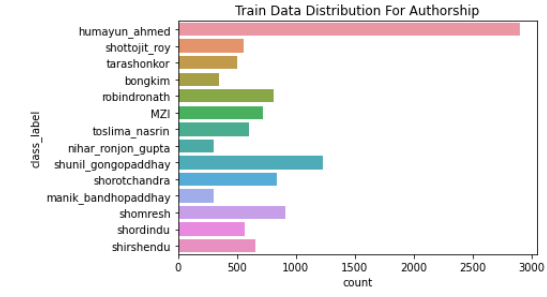


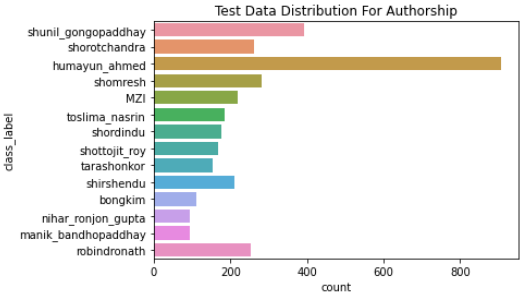


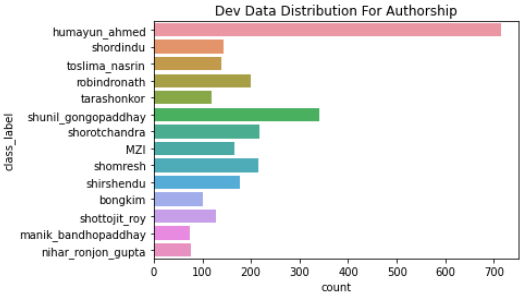


2) Authorship Classification

The writings of 14 distinct writers are included in the authorship dataset [31], [33]. The dataset contains documents with a fixed number of words of 750. The 14047, 3,511, and 750 writings for the train, development, and test divides, respectively, compose of the data splits.

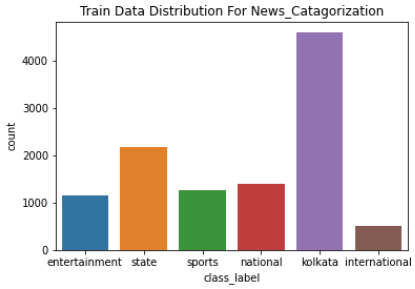


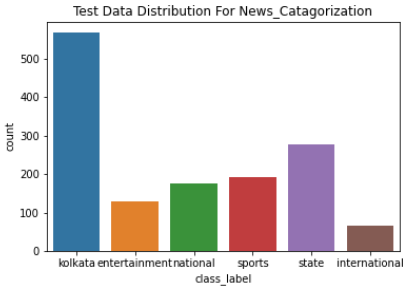


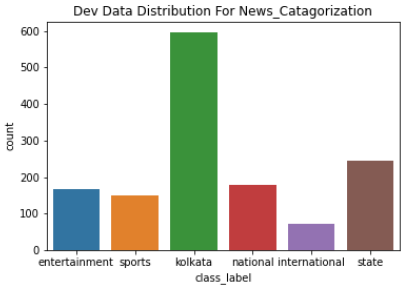


3) News Categorization

This news classification dataset, which has training, development, and test splits with 11109, 1408, and 1407 news articles, respectively, comprises six different class labels [31], [32]. The dataset contains a low distribution of the international news category.





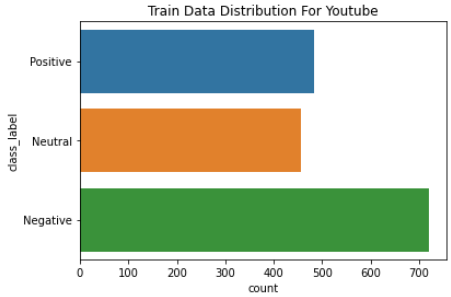


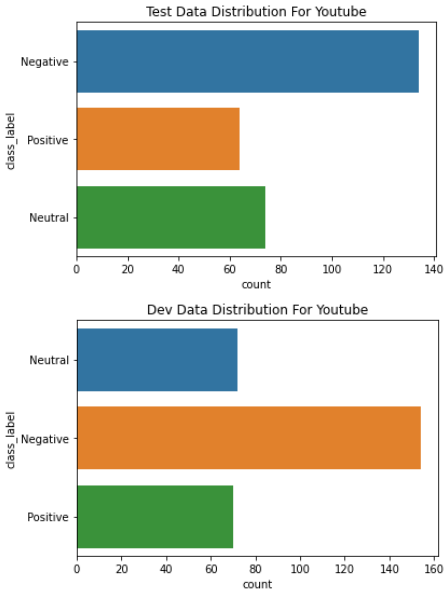
4) Sentiment Classification

For the research work, we combined and used separate publicly accessible sentiment analysis datasets [25]. In the section that follows, we provide such a succinct overview of each dataset.

- Youtube Comments Dataset [28]

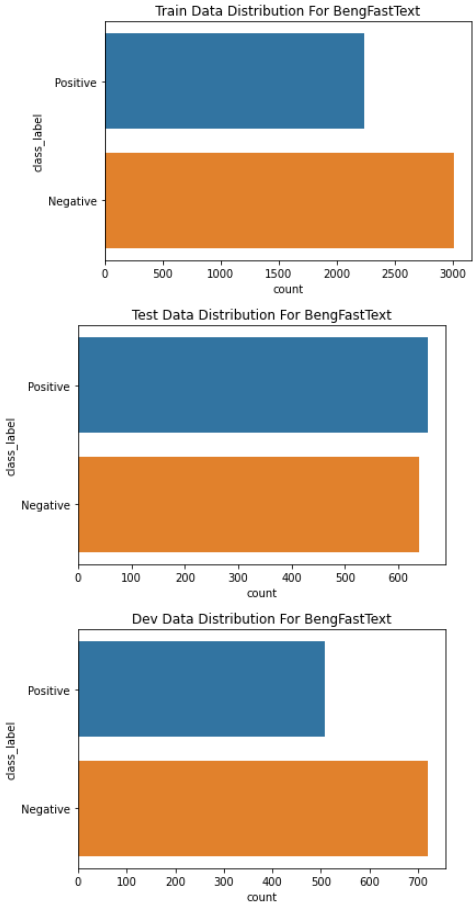
The YouTube dataset has been created by combining a variety of comments posted on the site. There are 1660 comments for training, 273 comments for testing, and 297 comments for development in its three classes.





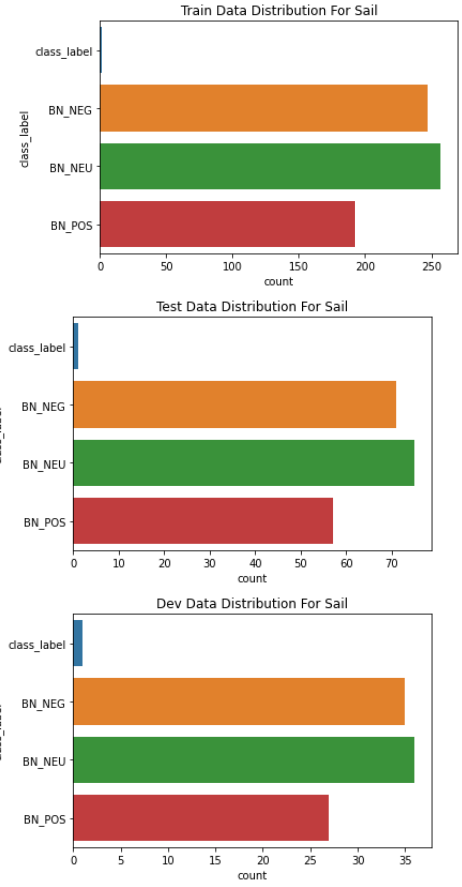
- BengFastText Dataset [29]

A combination of newspapers, TV news, books, blogs, and social media sites have been used to assemble the BengFastText dataset. The total quantity of data in this dataset is 7776, which is divided into 5253 training data, 1228 development data, and 1295 test data.



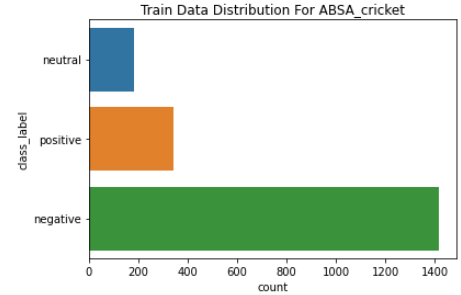
- Sail Dataset [26]

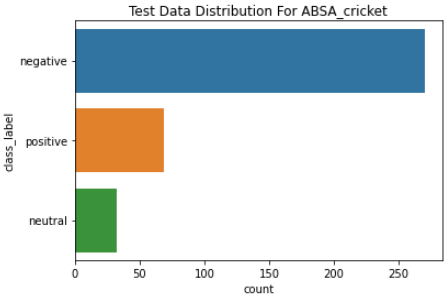
For the Shared Task on Sentiment Analysis in Indian Languages (SAIL), a dataset named SAIL was constructed, most of which are tweets. The dataset includes a total of 1000 tweets, which are broken down into 697 posts for training, 204 posts for testing, and 99 posts for development, respectively.

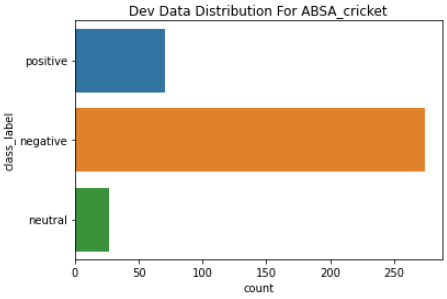


- ABSA Cricket Dataset [27]

This dataset, which was created using data from Facebook, BBC Bangla, and Prothom Alo, comprises a total of 2688 entries, of which 1943 are for training, 373 are for development, and 372 are for testing.

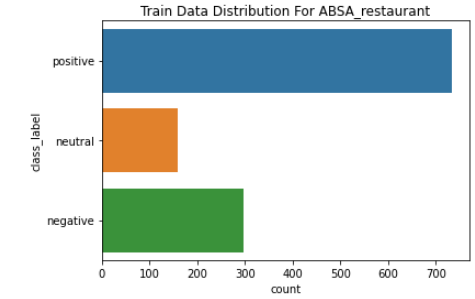


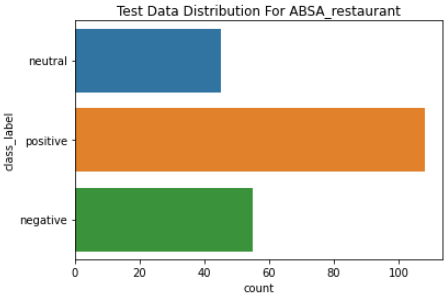


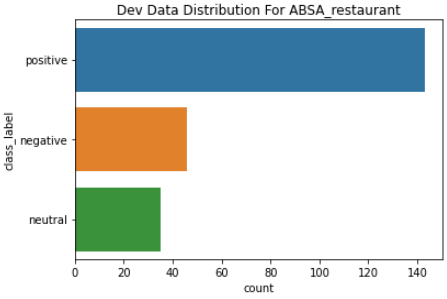


- ABSA Restaurent [27]

To perform aspect-based sentiment analysis in Bangla, the ABSA dataset was assembled. Only the restaurant category of data is contained in the dataset. The dataset includes 225 data for development, 209 data for testing, and 1188 data for training. This dataset contains three classes.

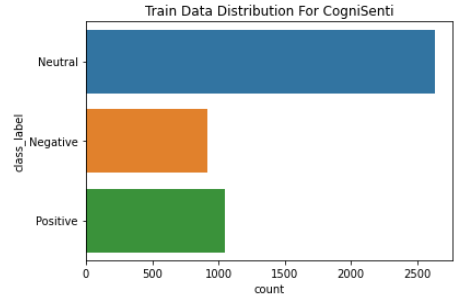


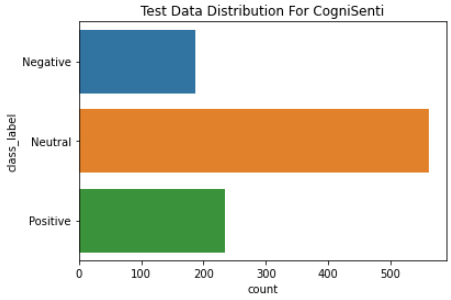


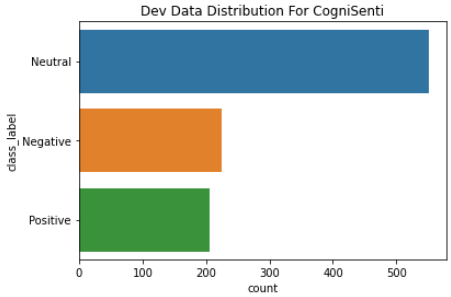


- CogniSenti Dataset [30]

The CogniSenti dataset, which is composed of posts on social media, has 5,628 tweets from Twitter and 942 posts from Facebook. This dataset divided the data into training, development, and test sets, each of which had 4,599, 985, and 986 data, accordingly, in order to evaluate performace.

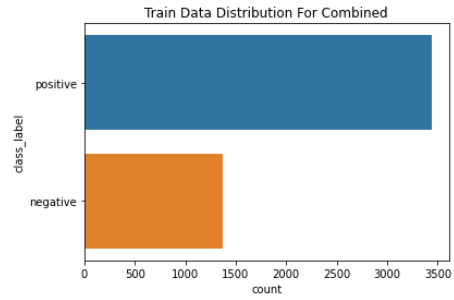


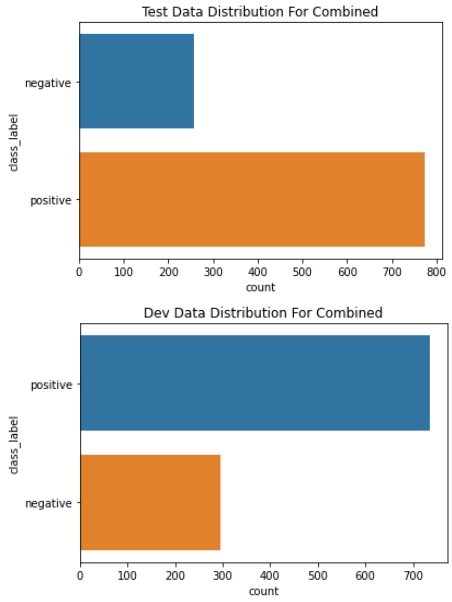




- Combined

This dataset separately combines all training datasets, test datasets, and development datasets of sentiment classification. It includes 4807 training data, 1031 test data, and 1031 development data.





5) Pos Tagging

For training and evaluating the models for the POS tagging task, we used three datasets: LDC Corpus [19], [20], IITKGP POS Tagged Corpus [21], and CRBLP POS Tagged Corpus [22]. We used a three-step for training and testing the models.

- LDC Corpus

In the first procedure, we used the LDC Corpus, which has a total of 7,393 sentences, or 102,937 tokens. It was developed by Microsoft Study (MSR), India, for linguistic research. The corpus's text was compiled from blogs, Wikipedia articles, and other sources to provide it diversity.

- LDC+IITKGP Corpus

The IITKGP POS Tagged Corpus and LDC Corpus were integrated in the next phase. There are 72,400 tokens and 5,473 phrases in the IITKGP POS Tagged Corpus. Microsoft Research created this corpus in conjunction with IIT Kharagpur and numerous other Indian institutes.

- LDC+IITKGP+CRBLP Corpus

The last approach was combining the LDC+IITKGP and CRBLP POS Tagged Corpus from before. 20000 tokens are distributed throughout 1176 phrases in this CRBLP POS Tagged Corpus.

6) Punctuation Restriction

The train, development, and test splits were created using a publically available corpus of Bangla articles as the dataset for the punctuation restoration task [23], [24]. ASR and manual excerpts of texts were used to create two test datasets by the authors. These were compiled from 65 minutes of voice snippets taken from four short tales in Bangla. There are four labels, comprising the comma, period, questions, and other tokens, as well as three punctuation marks. This dataset consists 1,379,986 tokens for Train, 179,371 tokens for Dev, and 87,721 tokens, 6,821 tokens, and 6,417 tokens for Test(news), Test(Ref.) and TEST(ASR) respectively.

C. CONFIGURATIONS

We fine tuned a tiny Bert model from scratch named ckiplab/albert-tiny-chines [18] using basic transformers, an NLP package, and our heterogeneous dataset. We had considered our fine tuned model as our base model with 312 hidden size and 30522 vocab size which has 4 hidden layers and 12 attention heads. Additionally, we used the iPET and pruned methodologies to consider two additional models with the same configuration figure. To compare our models we had selected 7 models from our study. The configuration for those models are shown in the Table 3.

D. METRICS

Perplexity is a metric that measures how effectively a probability model predicts a given sample. Perplexity is one technique to evaluate language models in the context of Natural Language Processing. The optimal language model is one that uses data that has never been seen before. We evaluate our model depending on the perplexity.

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(W_i|W_{i-1})}} \tag{4}$$

We measured perplexity from our dataset using this method. Also, we measured for each categories from the dataset. The results are shown in Table 1.

TABLE 1. Perplexity on diverse dataset

Dataset	Eval loss	Perplexity
Full Dataset	2.28	9.43
Sadhu	2.29	9.88
Social Media	2.36	10.52
Newspaper	2.24	9.29

TABLE 2. Perplexity of all three model

Model	Perplexity
Base Model	3.8722
iPET Model	1.0460
Pruned Model	46.8558

V. RESULTS

Using our diverse dataset, we fine-tuned a tiny Bert model from scratch, named ckiplab/albert-tiny-chines [18]. After that, we applied iPET to that model. Finally, we pruned the

TABLE 3. Model Configurations

Model	Attention heads	Hidden layers	Hidden size	Vocab size
Bnbert(ours)	12	4	312	30522
Bnbert ipet(ours)	12	4	312	30522
Bnbert ipet Pruned(ours)	12	4	312	30522
BanglaBert [17]	12	12	768	32000
Bangla-Electra [16]	4	12	256	29898
Indic-Bert [16]	12	12	768	100000
Bert-bn [16]	12	12	768	102025
XLM-RoBERTa [16]	16	24	1024	250002
Indic-DisrilBert [16]	12	6	768	30522
DisrilBERT-m [16]	12	6	768	119547

model that the iPET methodology.All three models' levels of perplexity were measured and are shown in Table 2. Performing iPET methodology on base model is better compared to Base model and Pruned model. The same downstreming tasks were applied to each model to evaluate how well it performed. On another model called BanglaBert, all the same downstreming tasks were carried out for comparing with our models. Additionally, we compared the performance of our model with other known models from this research [16]. We used weighted average for Precision, Recall, and F1 to evaluate the performance of our model since it can handle the problem of dataset imbalance.

A. EMOTION CLASSIFICATION

In the table 4, We evaluate how well our models performed at this specific task. According to the F1 score, our Bnbert ipet Pruned and Bnbert outperform Bangla Electra at emotion classification. Furthermore, our Bnbert ipet Pruned model performs better than BERT-bn and nearly as well as Indic-Bert.

B. AUTHORSHIP CLASSIFICATION

Table 5 compares the performance of our models against that of XLM-RoBERTa, Indic Bert, and Bert-bn. Bnbert, Bnbert ipet, and Bnbert ipet Pruned each achieve an F1 score of 97.11%, 97.73%, and 97.68%, correspondingly. Additionally, our models provide outcomes that are equivalent to BanglaBert.

C. NEWS CATEGORIZATION

Table 6 contains the findings for the categorization of news task. Our Bnbert ipet Pruned model works almost as well as Indic-DistilBERT for categorizing news, and three other models of ours outperform Bangla Electra and DistilBERT-m. In terms of accuracy, Bnbert ipet Pruned also performs on able to compete with the larger BanglaBert model.

D. POS TAG

In table 7 demonstrates the our models performances in comparison to Bangla Electra, DistilBERT-m, BERT-bn, and BanglaBert. Take into account Bnbert ipet Pruned's accuracy and F1 score are on comparable with those of Bangla Electra and DistilBert-m. It performs F1 score 82.01% against the three merged datasets, whereas Bangla Electra performs 74.7% and DistilBERT-m performs 78.20% on the merge of three datasets.

E. SENTIMENT CLASSIFICATION

For both individual and combined dataset, we have included a summary of the sentiment classification results from several transformer models in table 8. Bnbert, Bnbert ipet, and Bnbert ipet Pruned are the models that perform better than Bangla Elcetra and DistilBERT-m based on Accuracy and F1 score for the combined dataset. Bangla Electra and DistilBERT-m perform 72.0% and 74.3% respectively among all of our models, whereas Bnbert ipet performs at 79.52% accuracy. Bert-bn and our models may be compared utilizing aggregated dataset, too. In this specific task, BanglaBerT outperforms all the other models.

F. PUNCTUATION RESTRICTION

We only consider the overall results by ignoring the no punctuation entries (O tokens) for this task when presenting the outcomes of the punctuation restriction in table 9 following. Bnbert, Bnbert ipet, and Bnbert ipet Pruned outperform XLM-RoBERTa, Indic Bert, Bangla Electra, and BanglaBert over the three datasets with this punctuation restriction. For the News dataset, Ref dataset, and ASR dataset, respectively, Bnbert ipet Pruned perform 96.39%, 96.19%, and 91.28% according to F1 score.

VI. INFERENCE TIME COMPARISION

According to the study[16], models have a large computational complexity, and the training and inference times are significantly higher than using traditional algorithms (such SVM, RF). The combined sentiment dataset, authorship classification dataset, emotion classification dataset, and news categorization dataset were utilized to measure the training and inference times for our Bnbert ipet Pruned model. The

TABLE 4. Emotion Classification

Model	Acc	P	R	F1
Bnbert(ours)	45.51	43.06	45.51	41.21
Bnbert ipet(ours)	43.71	41.08	43.71	39.84
Bnbert ipet Pruned(ours)	50.30	46.86	50.30	48.10
Bangla Electra [16]	43.8	38.3	43.8	36.3
Indic Bert [16]	50.6	52.1	50.6	49.1
Bert-bn [16]	49.1	46.7	49.1	46.9
BanglaBert [17]	71.05	66.46	71.05	68.62

TABLE 5. Authorship Classification

Model	Acc	P	R	F1
Bnbert(ours)	97.11	97.14	97.11	97.11
Bnbert ipet(ours)	97.72	97.76	97.72	97.73
Bnbert ipet Pruned(ours)	97.67	97.69	97.67	97.68
XLM-RoBERTa [16]	93.8	94.1	93.8	93.8
Indic Bert [16]	95.2	95.3	95.2	95.2
Bert-bn [16]	90.2	90.3	90.2	90.2
BanglaBert [17]	98.85	97.71	97.88	98.71

TABLE 6. News Categorization

Model	Acc	P	R	F1
Bnbert(ours)	82.51	83.03	82.51	81.50
Bnbert ipet(ours)	84.29	84.35	84.29	84.08
Bnbert ipet Pruned(ours)	88.48	88.24	88.48	88.19
Bangla Electra [16]	80.4	78.5	80.4	79.2
Indic-DisrilBert [16]	89.0	90.2	89.0	89.4
DisrilBERT-m [16]	79.5	79.4	79.5	79.0
Banglabert [17]	94.52	94.55	94.52	94.53

TABLE 7. POS Tag

	Bnbert(ours)				Bnbert ipet(ours)				Bnbert ipet Pruned(ours)			
	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
LDC	75.30	69.04	65.06	65.78	75.87	69.08	66.80	67.37	81.40	74.94	73.39	73.85
LDC+IITKGP	79.04	72.81	70.66	71.09	78.87	72.59	70.82	71.28	83.27	77.47	76.72	76.93
LDC+IITKGP+CRBLP	78.08	70.91	69.38	69.30	77.73	70.36	69.15	69.24	82.01	75.51	74.78	82.01
	Bangla Electra [16]				DistilBERT-m [16]				BERT-bn [16]			
	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
LDC	79.0	73.4	71.1	72.2	83.9	78.8	78.0	78.4	85.8	81.5	80.7	81.1
LDC+IITKGP	80.2	74.8	72.2	73.7	83.7	78.5	77.9	78.2	85.5	81.5	80.9	81.2
LDC+IITKGP+CRBLP	80.4	75.1	73.0	74.7	83.8	78.4	78.0	78.20	85.4	81.0	80.2	80.6
	BanglaBert [17]											
	Acc	P	R	F1								
LDC	91.54	88.40	89.04	88.67								
LDC+IITKGP	92.45	89.75	90.24	89.93								
LDC+IITKGP+CRBLP	92.08	88.97	89.59	89.22								

TABLE 8. Sentiment Classification

	Bnbert(ours)				Bnbert ipet(ours)				Bnbert ipet Pruned(ours)			
	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
Youtube comments	63.85	67.83	63.85	65.28	57.43	70.29	57.43	58.76	64.52	65.42	64.52	64.91
BengFastText	81.41	81.81	81.41	81.51	79.86	80.02	79.86	79.47	79.62	80.38	79.62	79.76
Sail	54.08	66.25	54.08	55.33	46.93	55.70	46.93	45.26	54.08	52.99	54.08	52.60
ABSA Cricket	75.53	69.48	75.53	72.35	77.41	70.46	77.41	72.17	72.31	71.66	72.31	71.62
ABSA Restaurent	58.92	48.49	58.92	53.09	58.03	61.35	58.03	57.09	60.71	58.05	60.71	58.32
CogniSenti	65.85	58.23	65.85	58.15	64.62	49.45	64.62	56.00	65.54	66.02	65.54	60.22
Combined	77.91	79.21	77.91	77.62	79.52	79.75	79.52	79.10	78.52	78.49	78.52	78.41
	Bangla Electra [16]				DisrilBERT-m [16]				BERT-bn [16]			
	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
Youtube comments	67.4	66.2	67.4	66.6	70.0	70.4	70.0	70.1	73.3	72.6	73.3	72.9
BengFastText	68.7	73.0	68.7	66.8	69.1	74.6	69.1	66.9	69.0	75.5	69.0	66.5
Sail	53.3	51.9	53.3	49.8	57.3	57.0	57.3	57.0	60.7	59.8	60.7	59.5
ABSA Cricket	71.3	63.5	71.3	67.0	74.5	66.5	74.5	69.9	71.1	67.8	71.1	69.2
ABSA Restaurent	50.2	35.7	50.2	40.9	60.3	57.7	60.3	57.9	60.7	59.3	60.7	58.5
CogniSenti	63.9	58.3	63.9	59.3	59.2	58.6	59.2	58.9	62.4	62.1	62.4	62.2
Combined	72.0	72.4	72.0	71.8	74.3	74.8	74.3	74.2	79.3	79.3	79.3	79.2
	BanglaBert [17]											
	Acc	P	R	F1								
Youtube comments	75.33	75.27	75.33	75.21								
BengFastText	84.75	84.70	84.75	84.69								
Sail	71.42	72.11	71.42	70.80								
ABSA Cricket	80.64	82.50	80.64	76.02								
ABSA Restaurent	72.76	71.87	72.76	72.20								
CogniSenti	70.03	72.58	70.03	69.14								
Combined	95.43	95.76	95.43	95.50								

TABLE 9. Punctuation Restriction

	News			Ref.			ASR		
	P	R	F1	P	R	F1	P	R	F1
Bnbert	94.32	98.38	96.31	85.04	97.11	90.67	86.55	96.71	91.35
Bnbert ipet	87.94	95.20	93.76	79.0	90.13	88.27	80.46	90.11	89.17
Bnbert ipet Pruned	94.55	98.23	96.36	86.99	96.19	91.36	88.51	95.73	91.98
XLM-RoBERTa [16]	87.8	86.2	87.0	67.6	70.2	68.6	60.3	66.4	63.2
Indic-Bert [16]	73.9	70.5	72.2	60.7	54.1	57.2	55.9	53.4	54.7
Bangla Electra [16]	64.8	49.7	56.3	59.5	39.2	47.2	54.8	38.7	45.3
BanglaBert [17]	94.77	98.21	96.46	68.07	40.64	50.86	68.93	56.21	61.93

training times for all of these tasks were 1 hour 40 minutes 39 seconds, 1 hour 8 minutes 34 seconds, 3 minutes 14 seconds, and 58 minutes 8 seconds, respectively. Those tasks took 15, 16, almost 4, and 5 seconds, correspondingly, in inference time. SVM required 2 hours 32 minutes 30 seconds for training and 4 seconds for inference. For XLM-RoBERTa, the training phase took 3 hours 42 minutes 41 seconds, while the inference phase took 1 minute 21 seconds. The inference time for our model was 3.75 time higher than that of SVM but 5.4 times less than that of XLM-RoBERTa. Our research shows that our Bnbert ipet Pruned model performs on able to compete with large models like XLM-RoBERTa and BanglaBert in several downstreaming tasks for the inference time. This model was fine tuned from a small model with 12 attention heads and 4 layers using our diversified dataset.

VII. CONCLUSION

As a consequence of our research, we came up with an understanding that a smaller model with a small number of layers can be a few-shot learner. This few-shot learnt model can outperform the pretrained base model in perplexity due to iterative pattern exploitation training. Although this pattern exploitation training has minimal influence on down streaming performance, it does allow the model to learn a few shots. The Lottery Ticket Hypothesis is used to prune the model, which decreases its size over time and removes superfluous weights, making it lighter, greener, and more efficient to train and deploy. As a result, the model is comparable with the base model in the down-streaming task while having

just 10% of the weight of the main pretrained model. After everything we've done thus far, we've exceeded our goals for this experiment. Furthermore, we want to do future studies to increase the efficacy and efficiency of this strategy.

REFERENCES

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

[2] Lee, N., Ajanthan, T. and Torr, P.H., 2018. Snip: Single-shot network pruning based on connection sensitivity. arXiv preprint arXiv:1810.02340.

[3] Dettmers, T. and Zettlemoyer, L., 2019. Sparse networks from scratch: Faster training without losing performance. arXiv preprint arXiv:1907.04840.

[4] Wang, C., Zhang, G. and Grosse, R., 2020. Picking winning tickets before training by preserving gradient flow. arXiv preprint arXiv:2002.07376.

[5] Evci, U., Gale, T., Menick, J., Castro, P.S. and Elsen, E., 2020, November. Rigging the lottery: Making all tickets winners. In International Conference on Machine Learning (pp. 2943-2952). PMLR.

[6] You, H., Li, C., Xu, P., Fu, Y., Wang, Y., Chen, X., Baraniuk, R.G., Wang, Z. and Lin, Y., 2019. Drawing early-bird tickets: Towards more efficient training of deep networks. arXiv preprint arXiv:1909.11957.

[7] Frankle, J. and Carbin, M., 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. arXiv preprint arXiv:1803.03635.

[8] Le Cun, Y., Denker, J.S. and Solla, S.A., 1990. Optimal brain damage, Advances in Neural Information Processing. Vol. 2, DS Touretzky, ed.

[9] Hassibi, B. and Stork, D., 1992. Second order derivatives for network pruning: Optimal brain surgeon. Advances in neural information processing systems, 5.

[10] Han, S., Pool, J., Tran, J. and Dally, W., 2015. Learning both weights and connections for efficient neural network. Advances in neural information processing systems, 28.

[11] Li, H., Kadav, A., Durdanovic, I., Samet, H. and Graf, H.P., 2016. Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710.

[12] Schick, T. and Schütze, H., 2020. It's not just size that matters: Small language models are also few-shot learners. arXiv preprint arXiv:2009.07118.

[13] Chen, X., Cheng, Y., Wang, S., Gan, Z., Wang, Z. and Liu, J., 2020. Early-bert: Efficient bert training via early-bird lottery tickets. arXiv preprint arXiv:2101.00063.

[14] Chen, X., Cheng, Y., Wang, S., Gan, Z., Wang, Z. and Liu, J., 2020. Early-bert: Efficient bert training via early-bird lottery tickets. arXiv preprint arXiv:2101.00063.

[15] Alam, F., Hasan, A., Alam, T., Khan, A., Tajrin, J., Khan, N. and Chowdhury, S.A., 2021. A Review of Bangla Natural Language Processing Tasks and the Utility of Transformer Models. arXiv preprint arXiv:2107.03844.

[16] Chen, T., Frankle, J., Chang, S., Liu, S., Zhang, Y., Wang, Z. and Carbin, M., 2020. The lottery ticket hypothesis for pre-trained bert networks. Advances in neural information processing systems, 33, pp.15834-15846.

[17] Bhattacharjee, A., Hasan, T., Uddin, W.A., Mubasshir, K., Islam, M.S., Iqbal, A., Rahman, M.S. and Shahriyar, R., 2022. Banglabert: Lagnuage model pretraining and benchmarks for low-resource language understanding evaluation in bangla. Findings of the North American Chapter of the Association for Computational Linguistics: NAACL.

[18] Huggingface.co. 2022. ckiplab/albert-tiny-chinese · Hugging Face. [online] Available at: <https://huggingface.co/ckiplab/albert-tiny-chinese> [Accessed 23 August 2022].

[19] Sankaran Baskaran, Kalika Bali, Tanmoy Bhattacharya, Pushpak Bhattacharyya, Girish Nath Jha, et al. 2008. A common parts-of-speech tagset framework for indian languages. In Proc. of LREC 2008. Citeseer.

[20] Monojit Choudhury Kalika Bali and Priyanka Biswas. 2010. Indian Language Part-of-Speech Tagset: Bengali LDC2010T16. Technical Report. Philadelphia: Linguistic Data Consortium.

[21] A Kumaran. 2007. A Part of Speech Tagger for Indian Languages (POS tagger). Technical Report. Microsoft Research.

[22] Rabia Sultana Ummi and Fahmina Huda. 2008. Developing language resources for English machine translation. Technical Report. BRAC University.

[23] Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Punctuation Restoration using Transformer Models for High-and-Low-Resource Languages. In Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020). Association for Computational Linguistics, 132–142.

[24] Aisha Khatun, Anisur Rahman, Hemayet Ahmed Chowdhury, Md. Saiful Islam, and Ayesha Tasnim. 2019. A Subword Level Language Model for Bangla Language. CoRR abs/1911.07613 (2019). arXiv:1911.07613 http://arxiv.org/abs/1911.07613

[25] Md. Arid Hasan, Jannatul Tajrin, Shammur Absar Chowdhury, and Firoj Alam. 2020. Sentiment Classification in Bangla Textual Content: A Comparative Study. In 2020 23rd International Conference on Computer and Information Technology (ICCIT). 1–6.

[26] Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. Shared Task on Sentiment Analysis in Indian Languages (SAIL) Tweets - An Overview. In Mining Intelligence and Knowledge Exploration, Rajendra Prasath, Anil Kumar Vuppala, and T. Kathirvalavakumar (Eds.). Springer International Publishing, 650–655.

[27] Md Rahman, Emon Kumar Dey, et al. 2018. Datasets for Aspect-Based Sentiment Analysis in Bangla and Its Baseline Evaluation. Data 3, 2 (2018), 15.

[28] Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from Bangla youtube comments. In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP). IEEE, 1–6.

[29] Md Rezaul Karim, Bharathi Raja Chakravarthi, Mihael Arcan, John P McCrae, and Michael Cochez. 2020. Classification Benchmarks for Under-resourced Bengali Language based on Multichannel Convolutional-LSTM Network. arXiv (2020), arXiv–2004.

[30] Md. Arid Hasan, Jannatul Tajrin, Shammur Absar Chowdhury, and Firoj Alam. 2020. Sentiment Classification in Bangla Textual Content: A Comparative Study. In 2020 23rd International Conference on Computer and Information Technology (ICCIT). 1–6.

[31] Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Bangla Text Classification using Transformers. arXiv preprint arXiv:2011.04446 (2020).

[32] Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. AI4Bharat-IndicNLP Corpus: Monolingual Corpora and Word Embeddings for Indic Languages. arXiv preprint arXiv:2005.00085 (2020).

[33] Aisha Khatun, Anisur Rahman, Md Saiful Islam, et al. 2019. Authorship Attribution in Bangla literature using Character-level CNN. In 2019 22nd International Conference on Computer and Information Technology (ICCIT). IEEE, 1–5.

[34] Das, A., Sharif, O., Hoque, M.M. and Sarker, I.H., 2021. Emotion classification in a resource constrained language using transformer-based approach. arXiv preprint arXiv:2104.08613.

[35] Kowsher, M., Sami, A.A., Prottasha, N.J., Arefin, M.S., Dhar, P.K. and Koshiba, T., 2022. Bangla-BERT: Transformer-based Efficient Model for Transfer Learning and Language Understanding. IEEE Access.

[36] Tanaka, H., Kunin, D., Yamins, D.L. and Ganguli, S., 2020. Pruning neural networks without any data by iteratively conserving synaptic flow. Advances in Neural Information Processing Systems, 33, pp.6377-6389.

[37] Schick, T. and Schütze, H., 2020. Exploiting cloze questions for few shot text classification and natural language inference. arXiv preprint arXiv:2001.07676.

[38] Bhowmick, A. and Jana, A., Sentiment Analysis For Bengali Using Transformer Based Models.

[39] Alam, T., Khan, A. and Alam, F., 2020, November. Punctuation restoration using transformer models for high-and low-resource languages. In Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020) (pp. 132-142).

[40] Wang, Z., Wohlwend, J. and Lei, T., 2019. Structured pruning of large language models. arXiv preprint arXiv:1910.04732.