Alternative non-BERT model choices for the textual classification in low-resource languages and environments

Syed Mustavi Maheen, Moshiur Rahman Faisal, Rafakat Rahman, Md. Shahriar Karim
Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh
{mustavi.maheen, moshiur.faisal, rafakat.rahman}@northsouth.edu
shahriar.karim@northsouth.edu

Abstract

Natural Language Processing (NLP) tasks in non-dominant and low-resource languages have not experienced significant progress. Although pre-trained BERT models are available, GPU-dependency, large memory requirement, and data scarcity often limit their applicability. As a solution, this paper proposes a fusion chain architecture comprised of one or more layers of CNN, LSTM, and BiLSTM and identifies precise configuration and chain length. The study shows that a simpler, CPU-trainable non-BERT fusion CNN + BiLSTM + CNN is sufficient to surpass the textual classification performance of the BERT-related models in resource-limited languages and environments. The fusion architecture competitively approaches the state-of-the-art accuracy in several Bengali NLP tasks and a six-class emotion detection task for a newly developed Bengali dataset. Interestingly, the performance of the identified fusion model, for instance, CNN + BiLSTM + CNN, also holds for other low-resource languages and environments. Efficacy study shows that the CNN + BiLSTM + CNN model outperforms BERT implementation for Vietnamese languages and performs almost equally in English NLP tasks experiencing artificial data scarcity. For the GLUE benchmark and other datasets such as Emotion, IMDB, and Intent classification, the CNN + BiLSTM + CNN model often surpasses or competes with BERT-base, TinyBERT, DistilBERT, and mBERT. Besides, a position-sensitive self-attention layer role further improves the fusion models' performance in the Bengali emotion classification. The models are also compressible to as low as $\approx 5\times$ smaller through pruning and retraining, making them more viable for resource-constrained environments. Together, this study may help NLP practitioners and serve as a blueprint for NLP model choices in textual classification for low-resource languages and environments.

1 Introduction

Many developed nations are now considering deep learning approaches for tackling textual toxicity in social media. But countries lacking substantial socio-economic capacity and technological infrastructures are lagging. The current trend of NLP research evolves mainly around a few dominant languages, leaving NLP research for many low-resource languages unattended or less explored (Joshi et al., 2020). The NLP tasks in low-resource languages generally suffer from exceptionally scarce resources, ranging from lack of annotated data to insufficient computational facilities. In contrast, most NLP breakthroughs that achieve high accuracy are computationally intensive, making it more challenging for societies suffering from inadequate technological infrastructures. For instance, while the bidirectional transformer BERT has about 340 millions parameters (Devlin et al., 2018), a more advanced model GPT-3 (Brown et al., 2020), has about 170 billions parameters, requiring extensive GPU/TPU support and memory storage that may be unaffordable for low-resource societies. As a result, low-resource languages and environments are frequently left out with little attention from the NLP community (Joshi et al., 2020).

Further complicating matters, the serverless free deployment of deep learning models, as commonly done using Amazon Web Services (AWS) and Google Cloud Platform (GCP), is restrictive for larger model size (Han et al., 2015a,b). Also, latency increases with increasing memory requirement and model size, suggesting memory-intensive device GPU/TPU for faster inference and response. These additional financial costs limit access to BERT models for NLP community works in resource-constrained environments (Strubell et al., 2019). One intriguing question thus arises: could computationally less-expensive non-BERT models reduce GP/TPU dependency and associated fi-

Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing, pages 192 - 202

nancial cost without affecting the classification accuracy for textual classification in a low-resource context?

The multilingual-BERT (mBERT) (Devlin et al., 2018; Pires et al., 2019) and its reduced versions (Abdaoui et al., 2020), other compressed BERT modifications, such as TinyBERT (Jiao et al., 2019), MobileBERT (Sun et al., 2020), are a few viable models proposed for many languages and contexts, including the low-resource ones. Nevertheless, these models require additional fine-tuning and training for target-specific NLP tasks, requiring GPU/TPU support even in a resource-constrained context. Also, size of these models may not be optimal for deployment in low-end devices. So, textual classification in many non-dominant languages remains rudimentary, leaving the communities unequipped against the increasing toxicity and abusive comments on social platforms. Besides, many textual classification tasks do not require a rigorous use of linguistic semantics. So, models that are structured well against the semantics, for instance, the BERT models, may not always be the most optimal choice in NLP tasks less dependent on language semantics. Thus, a viable trade-off between the deployability, scarce resources, and DNN models' accuracy in NLP tasks for low-resource languages and environments needs unraveling.

As a solution, this study integrates local and global dependencies in sentences by bringing alternative DNN models into a hybrid model structure, namely the fusion chain models. Subsequently, a rigorous architecture search identifies deployable DNN models for low-resource languages, with an improved understanding on a few intriguing questions such as:

- How effective are the homogeneous (of similar layers) and heterogeneous (of different layers) form of fusion of one or more DNN layers in textual classification tasks?
- What chain length is optimal to maintain accuracy and reduce the difference between training and validation loss?
- How helpful the self-attention is for fusion models, and what is its optimal position?

We identify that classification accuracy is sensitive to fusion chain length, beyond which classification accuracy deteriorates considerably. Subsequent exploration of the identified fusion models reveals a position-sensitive performance of the self-attention layer for the newly annotated six-class Bengali emotion dataset.

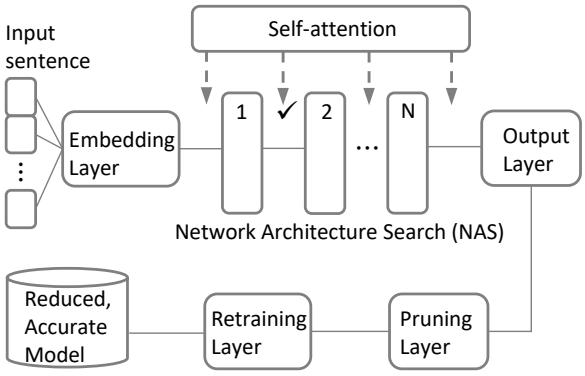

Figure 1: The word embedding layer acts as the input for the fusion of DNN layers during the NAS process. In the NAS, CNN, LSTM or BiLSTM layer are all considered as the initial layer, however the subsequent layers depended on the type of initial layer chosen finally resulting the three alternative chain-structures. The output from the DNN fusion requires pruning and retraining to generate the deployable models.

2 Related Works

Previous works attempted alternative deep learning models in NLP tasks for low-resource languages and environments. For instance, using a teacher-student framework, the BERT distillation with simpler models such as CBoW + FFN and BiLSTM as the student models for the limited availability of labeled data (Wasserblat et al., 2020). While such models are more deployable in low-end devices, the training still relies on a memory-hungry and costly setup requiring GPU/TPU as well as large unlabelled data for student model training. Alternative approaches consider freezing the BERT-layer outcomes by assessing their roles in the classification process (Grießhaber et al., 2020), requiring GPU/TPU support to train. Also, the sequence of frozen layers may vary across alternative datasets, and hence, the accuracy for a particular set of frozen layers becomes context-dependent. Instead, we investigate if a simple, CPU-trainable CNN and RNN fusion layer stack can achieve textual classification accuracy in NLP tasks where syntactical knowledge is less influential than the keywords or sentiment-based phrases. To find out such alternative non-BERT models, we propose fusion-chain architecture comprising one or more CNN and RNN layers and perform a rigorous network architecture search (NAS). Interestingly, the NAS process identifies a few optimal candidate mod-

els capable of achieving accuracy comparable to the baseline models, as elaborated further in the subsequent sections.

The emergence of more advanced deep neural networks capable of learning the word orders and information dependency in sentences replaces the classical machine learning models (Mikolov et al., 2013) in many NLP tasks. Precisely, the neural network models of the form of RNN (LSTM, BiLSTM) or CNN independently, or in combination with a pre-trained word embedding facility such as word2vec (Mikolov et al., 2013), fasttext (Joulin et al., 2016), have become the standard alternatives. For instance, Dynamic CNN architecture (DCNN) performs semantic modeling to identify words' short and long-range relations in sentences (Kalchbrenner et al., 2014). Whereas the CNN-based models are good at local and position-invariant feature extraction, the LSTM/BiLSTM models explicitly treat sentences as a sequence of words and capture sentence-level (for instance, syntactical (Zhu et al., 2015)) dependencies. Also, a few alternative attempts integrate local and global textual dependencies using CNN and RNN architectures (known as hybrid models) to improve accuracy of textual classification reviewed thoroughly in (Minaee et al., 2021).

Intriguingly, the hybrid models also appear promising for target-specific sequential analysis, as evident from quantifying the function of specific DNA sequences (Quang and Xie, 2016). Named Entity Recognition (NER) tasks also employ a hybrid approach by merging BiLSTM and CNN models (Chiu and Nichols, 2016). One of the initial works leveraging the advantages of both CNN and RNN architectures for textual classification is the Convolutional-LSTM (C-LSTM). Precisely, in C-LSTM, n-gram features extracted by a CNN layer are fed to the LSTM layer for learning the intra-sentence sequential dependence of words (Zhou et al., 2015). Authors in (Zhang et al., 2016) also tried a hybrid model with LSTM outputs fed to a CNN layer in document modeling. Alternative models include an attention mechanism with either CNN or RNN architecture to optimize textual classification performance further. For instance, Attention-Based Bidirectional Long Short-Term Memory Networks (Att-BLSTM) capture the position variant semantic information from the sentences (Basiri et al., 2021). Another study implements an attention-based Convolutional Neural Network (ABCNN) to model a pair of sentences (Yin et al., 2016). However, most of the studied hybrid models are single and two-layer models and did not explore the relevance of a larger stacking depth in textual classification tasks. The optimal fusion length and the order of the layers are still debatable and context-dependent. Besides, these CPU-implementable models facilitate the exploration and deployment of DNN models in low-resourced environments devoid of adequate advanced computing devices and facilities.

Algorithm 1 Fusion chain generation in NAS

Require: Input and Embedding Layer
Require: N = Max. fusion chain length
Require: RNN = LSTM | BiLSTM
Require: Initial Fusion Layer = CNN | RNN
Ensure: i = RandomNumber $(1$ to $N-1)$
  Fusion Model = Initial Fusion Layer
  for
    $x \leftarrow 0$ `to` i **do**
    if x is even **then**
      `Layer` $\leftarrow$ RNN
    else if x is odd **then**
      `Layer` $\leftarrow$ CNN
    end if
    Append `Layer` to Fusion model
    Append GlobalMaxPooling, Output Layer
    Return: Fusion model
    if Fusion chain length $> N$ **then**
      `BREAK`
    end if
  end for

3 Models and Methods

3.1 Proposed fusion chain models

Alternative DNN versions possess different strengths in NLP tasks. For instance, CNN (LeCun et al., 1998) models are good at position invariant text classification tasks, whereas the RNN (Elman, 1990) models are more pertinent for sequential processing of the input texts. However, the basic RNN structure frequently suffers from vanishing gradient problems, and the improved RNN variants are—Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014). Many NLP tasks such as sentiment analysis, emotion detection, have striking similarity, as the attributes are largely keywords dependent. Because of the sequential structures of

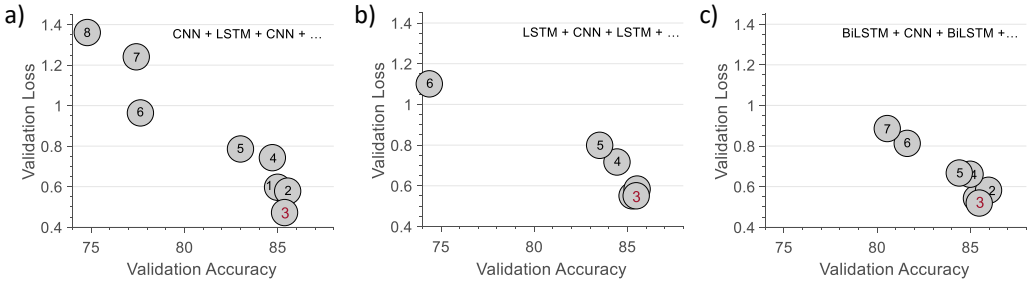

Figure 2: a, b, c) Optimal chain length for the three alternative fusion chain models studied extensively as part of the NAS.

LSTM and GRU, and their ability to remember previous text sequences, they perform well where context-dependencies are crucial (Yin et al., 2017). Another variant of LSTM, the Bidirectional LSTM (BiLSTM), comprises two LSTMs taking input sequence in forward and reverse directions, exhibits improved performance over single-LSTM in many applications (Huang et al., 2015). While each deep learning variant has its strength, a legitimate question thus arises—if a fusion model, formed with the DNN variants in a fusion chain, enhance performance of textual classification. An immediate next interesting question thus becomes the optimal chain length of the proposed fusion model.

3.2 Optimal length of the fusion chain

Textual classification accuracy depends on the context length of a word in a sentence. Fusing multiple DNN layers can increase the context length, but the optimal stacking depth for the DNN layers remains elusive and requires unravelling. The proposed fusion architecture follows a generic structure— it starts with an input layer, followed by an embedding layer that generates an embedding matrix for the given input sentence. A DNN layer is introduced immediately next to the embedding layer. Subsequently, additional DNN layers are added to form a fusion chain model of DNN layers, as schematically shown in Fig. 1. We performed random search for an the optimal fusion chain length, using several performance objectives, including the higher classification accuracy. The network architecture search (NAS) for an optimal chain length randomly generates even and odd numbers to decide if the next stacking to be done by an LSTM/BiLSTM (for even) or CNN (for odd) layer. The current fusion process does not consider similar DNN layers to be stacked together. The maximum length of fusion chain considered in the NAS is eight, beyond which the classification accuracy becomes considerably low (data not shown). The NAS process for optimal fusion chain length is summarized in algorithm 1.

3.3 Generalized random search

We implemented a generalized random search for a set of hyper-parameters in Keras (Chollet et al., 2018) and used it in all the experiments conducted for the analysis of fusion chain models. Interestingly, the random search process needs manual tuning of only one parameter, namely the maximum word length of a sentence that affects the shape of attention and LSTM layers. With this little tuning, the search process as developed in this study remains applicable for other similar textual classification tasks. Each layer in the random search is accompanied by an activation layer, a batch normalization layer, and a dropout layer to minimize the overfitting error. The CNN and RNN layers here also include kernel, bias, and activity regularizers (see the supplemental data for details).

3.4 Metrics used for comparison

The initial architecture search uses classification accuracy on the test dataset and the loss difference (LD = validation loss - training loss) as the performance metrics. The classification accuracy is defined as (TP + TN)/(TP + TN + FP + FN) with TP, TN, FP, FN are true positive, true negative, false positive, and false negative, respectively. The random search also considers early stopping to control the overfitting error [1]. For a comparison between the baseline models and the CNN + BiLSTM + CNN fusion model, we also considered other metrics, such as the number of parameters (# params), number of floating point operations (# FLOPs). Generally, experiments conducted in this study consider a 80% (training) and 20% (testing)

[1]Data and codes are available here in this link

Table 1: Performance of alternative fusion models for the new 6-class emotion Bengali dataset.

Model structure	Accuracy (T)	LD
Classical Machine Learning Models		
1. SVM	41.93	NA
2. KNN	72.79	NA
3. Random Forest	81.43	NA
Fusion models		
4. CNN + CNN + CNN	85.62	0.491
5. LSTM + LSTM	85.43	0.541
6. CNN + LSTM + CNN	86.61	0.283
7. LSTM + CNN + LSTM	85.74	0.483
8. BiLSTM + BiLSTM	86.54	0.126
9. BiLSTM + CNN + CNN	85.25	0.143
10. CNN + BiLSTM + CNN	84.54	-0.058
11. BiLSTM + LSTM	85.14	0.206
12. BiLSTM + LSTM + BiLSTM	85.49	0.057
13. BiLSTM + CNN + BiLSTM	85.86	-0.005
Fusion models + attention		
14. CNN + attn. + BiLSTM + CNN	86.83	
15. CNN + attn. + LSTM + CNN	86.91	
BERT models		
16. mBERT	86.62	0.457
17. Bangla BERT	86.17	0.177

split, and use fasttext (Joulin et al., 2016) as word embedding method.

3.5 Datasets

The study considers datasets across different languages and contexts for the efficacy demonstration of CNN + BiLSTM + CNN fusion. We developed a new Bengali corpus for 6-class emotion classification, as well as used other previously developed Bengali datasets for different NLP tasks– i) Six-class emotion Bengali dataset (Das et al., 2021), ii) Hate Speech Bengali dataset (Romim et al., 2021), and iii) DeepHateExplainer Bengali dataset (Karim et al., 2020). As examples of non-Bengali languages that relate the low-resource contexts, we consider the **Vietnamese** (Ho et al., 2019) and **Indonesian** (Saputri et al., 2018) datasets. The low-resource contexts in English considers an artificial data scarcity for the Stanford Sentiment Treebank 2 (**SST-2**), (Socher et al., 2013), emotion classification dataset (**Emotion**) (Saravia et al., 2018), and the Internet Movie Database (**IMDB**) review dataset (Maas et al., 2011). Finally, the efficacy study of the CNN + BiLSTM + CNN fusion model also considers evaluating the model on the on the General Language Understanding Evaluation the **GLUE benchmark** (Wang et al., 2018); however, we used randomly chosen 250 samples only from each classes to mimic artificial data scarcity.

3.6 Baseline models

We compare CNN + BiLSTM + CNN and other fusion models as identified against the models previously introduced for resource-constrained environments. A few such models are BERT-base (uncased) (Devlin et al., 2018), mBERT (Abdaoui et al., 2020), DistilBERT (Sanh et al., 2019), and TinyBERT (Jiao et al., 2019). The chosen models are all BERT related, and a few of which, for instance, DistilBERT, and TinyBERT, come with reduced size and additional fine-tuning for the resource-constrained environments and low-end devices. Besides the GLU benchmark, the mBERT is also used for the textual classification in Bengali.

4 Results and Discussion

Optimal fusion chain length of fusion models: The NAS process identifies (see Fig. 2a, b, c) that stacking unlimited DNN layers do not improve performance of the fusion models. Instead, the accuracy and LD of the textual classification deteriorate after the chain length attains an optimal value. Interestingly, chain-structure of length three or fewer layers yield the optimal performance (shown in Fig. 2a, b, c) irrespective of the fact whether fusion models start with any of the CNN, LSTM, BiLSTM layers. The NAS considers three fusion chains:

- CNN + LSTM + CNN + LSTM + . . . + CNN
- LSTM + CNN + LSTM + CNN + . . . + LSTM
- BiLSTM + CNN + BiLSTM + . . . + BiLSTM

A comparison between the competing models for our newly developed corpus of emotion classification reveals that accuracy deteriorates as the chain length goes beyond three. As it appeared, the accuracy gradually reduces to lower values as the length increases beyond three (shown in Fig. 2a, b, c). Among the models with a chain length of three or less, a model with a chain length of three is the smallest in LD values among the three allowed chains. A fusion chain that starts with a CNN layer attains the lowest validation loss and is explored further in subsequent analysis by replacing the LSTM layer with a BiLSTM layer.

GLUE benchmark with artificial data scarcity: The GLUE benchmark datasets have different sentence classification tasks. The performance evaluation of CNN + BiLSTM + CNN for all the categories has been done by assuming an artificial data scarcity. Precisely, the artificial scarcity considers only 250 samples from each class. As reported, the proposed CNN + BiLSTM + CNN model frequently outperforms baseline

Table 2: Efficacy study of CNN + BiLSTM + CNN fusion model considers GLUE benchmark datasets. Here, M and B stand for Millions and Billions, respectively. Only 250 samples were collected randomly to mimic a low-resource setup artificially for each class, among which 80% and 20% were for training and testing purposes. Here, accuracy colored in red is the highest, whereas the bold black is the next highest accuracy attained. The baseline models are all pre-trained versions available in https://huggingface.co/models

Model	# Params	# FLOPs	CoLA	WNLI	QQP	QNLI	RTE
BERT-base	109M	22.04B	63	46	61	70	75
mBERT	110M	22.04B	64	49	66	**73**	71
DistilBERT	52.2M	22.04B	**65**	47	65	74	76
TinyBERT	14.5M	0.119B	48	39	49	53	57
CNN + BiLSTM + CNN	0.4M	1.50M	**64**	**65**	**71**	**73**	**81**
CNN + LSTM + CNN	0.37M	1.43M	60	**64**	69	**74**	**81**
CNN + BiLSTM	0.38M	1.47M	62	62	**70**	71	**79**

Table 3: Comparison between CNN + BiLSTM + CNN model and BERT with frozen layers as in (Grießhaber et al., 2020) for 1000 randomly selected samples from SST-2 dataset (Socher et al., 2013).

Methods	**Model structure**	**SST-2**
BERT	no frozen layer	0.78 ± 0.059
	layer 1,2,3 frozen	0.80 ± 0.045
	layer 9,10,11 frozen	0.84 ± 0.013
Fusion	CNN + BiLSTM + CNN	0.80

models and approximates the rest for all different classification tasks available in GLUE benchmark (shown in Table 2). For instance, the comparison considers both the SST-2 (Socher et al., 2013) and CoLA (Warstadt et al., 2019) datasets for the single sentence classification task, and the CNN + BiLSTM + CNN model achieves the second-highest accuracy (64% for CoLA) marked as bold black with Distilled BERT accuracy at the top with 65% accuracy. Interestingly, in 4 sentence inference task (dataset RTE (Bentivogli et al., 2009)), the CNN + BiLSTM + CNN model achieves 81% accuracy exceeding all the other baseline models in the presence of artificial scarcity. In another inference task dataset, QNLI (Rajpurkar et al., 2016), the fusion model CNN + LSTM + CNN attains the maximum accuracy (74%) with CNN + BiLSTM + CNN and mBERT following it with an accuracy of 73%. The GLUE benchmark also includes three-sentence similarity tasks, and the CNN + BiLSTM+ CNN performed equally well for datasets such as QQP (Chen et al., 2018) with the highest and immediate next best performances with 71% and 70%, respectively. These experiments on different NLP tasks of the GLUE benchmark demonstrate the ability of CNN + BiLSTM + CNN models to perform better in data scarcity and low-end computational facilities.

Fusion and BERT models have comparable accuracy for a newly developed Bengali corpus: A few fusion chain models perform closely with BERT models for Bengali 6-class emotion corpus we developed (see supplemental information). Precisely, the Bangla BERT and mBERT models achieve 86.17% and 86.62%, whereas the CNN + LSTM + CNN fusion model reports an accuracy 86.61% (Table 1, row 6). The accuracy further improves for the same dataset with a self-attention layer added immediately after the initial CNN layer with an accuracy of 86.83% and 86.91% respectively (Table 1, row 14, 15). We primarily emphasized on minimizing overfitting error by lowering the difference between the validation loss and training. As observed, fusion models containig BiLSTM layers demonstrate a tendency of lowering the LD (Table 1, row: 8, 9, 10, 12, 13), and in fact, obtains the lowest LD = 0.057 among alternative fusion models. Interestingly, the fusion models performed very closely with the mBERT model, and in fact, outperformed mBERT in lowering the generalization error. For instance, reported mBERT LD = 0.457 (Table 1, row 16), whereas the CNN + LSTM + CNN model has a low LD = 0.28. The fusion models also perform well across other Bengali text classification datasets. For instance, CNN + BiLSTM + CNN model outperforms mBERT and BanglaBERT implementation for the reported dataset in (Das et al., 2021). In another dataset of Bengali hate speech detection (Romim et al., 2021), the fusion model with self-attention CNN + attn. + BiLSTM + CNN outperforms all the previous DNN and ML implementations, as evident from Table 4. However, for the dataset in (Karim et al., 2020), the fusion models fail to match the BERT-variants' performance (see Table 4) and surpass only the other DNN models. However, these datasets generally contain few thousands of samples for each classes, and do not necessarily represent data scarcity. Fur-

Table 4: Performance comparison between fusion models and alternative DNN and BERT models for various NLP-tasks in Bengali language. Here, A ≡ self-attention layer.

Group	Model structure	Accuracy (%)	Ref.
Six-class emotion Bengali dataset (Das et al., 2021)			
DNN	CNN + A + LSTM + CNN	**64.26**	Ours
	CNN + A + BiLSTM + CNN	**65.24**	
	CNN + A + GRU + CNN	**64.73**	
	CNN + BiLSTM	55.68	(2021)
	BiLSTM	58.08	(2021)
BERT	mBERT	64.63	
	Bangla-BERT	62.24	(2021)
	XLM-R	69.61	
Hate Speech Bengali dataset (Romim et al., 2021)			
ML	SVM	87.80	(2021)
DNN	fasttext + LSTM	84.30	
	fasttext + BiLSTM	86.55	(2021)
	word2vec + LSTM	83.85	
	CNN + A + BiLSTM + CNN	**88.65**	Ours
DeepHateExplainer (Karim et al., 2020)			
DNN	LSTM	75	(2020)
	BiLSTM	78	
	CNN + A + BiLSTM + CNN	**83.56**	Ours
BERT	Bangla-BERT	86	
	mBERT-cased	85	(2020)
	XML-Roberta	87	

ther exploration of the fusion models for other low-resources languages and contexts reveal the resilience of the identified models. For instance, the IMDB dataset (Maas et al., 2011) and the Emotion dataset (Saravia et al., 2018) were randomly reduced to mimick low-resource contexts. Subsequently, mBERT performance for the reduced datasets (5%, 10% for IMDB and 0.01%, 0.02% for Emotion) was compared against the fusion models' performance.

As appeared in Table 4, fusion models outperformed in all instances; in fact, it performed significantly better for the smaller dataset size considered. Ability of fusion models also remain equally competitive in other English NLP tasks, as demonstrated from classification accuracy comparison (see Table 6) between the fusion models and other BERT, DNN based implementation as in (Larson et al., 2019). Specifically, the fusion models attain a comparable accuracy of 93.62%, 93.28% as opposed to BERT-base's 94.4% reported in (Larson et al., 2019). Interestingly, the proposed method also perform competitively with the other low-resource fine-tuning, for instance, the freezing of BERT-layer approach as in (Grießhaber et al., 2020). Precisely, the CNN + BiLSTM + CNN model achieves higher accuracy than the BERT-base model reported, and almost equally perform to other tuned BERT-models of frozen layers, for a randomly selected 1000 samples from the SST-2 dataset (see Table 3).

Table 5: Training cost comparison between the baseline and fusion models using the average time per epoch for all the GLUE benchmark datasets studied.

Model	Average Time per Epoch (second)				
	CoLA	WNLI	QQP	QNLI	RTE
BERT-base	1286	1321	895	1421	783
mBERT	2540	1721	1296	2671	1026
DistilBERT	783	982	662	941	386
TinyBERT	19.6	24.4	19.8	24.4	18.8
CNN + BiLSTM + CNN	1.92	3.36	3.33	3.36	2.21
CNN + LSTM + CNN	1.25	3.26	2.25	3.18	1.11
CNN + BiLSTM	1.23	4.21	3	4.16	2.58

Table 6: Performance comparison between fusion models and alternative DNN and transformers models across different languages and datasets. Here, A ≡ attn.

Method	Model structure	Accuracy (%)	Ref.
Artificial scarcity: (5%, 10%) of **IMDB** dataset (Maas et al., 2011)			
Fusion	CNN + A + BiLSTM + CNN	(**84.79**, **85.10**)	Ours
BERT	mBERT	(81.40, 84.79)	-
Scarcity: (0.01%, 0.02%) **Emotion** dataset (Saravia et al., 2018)			
Fusion	CNN + A + LSTM + CNN	(**84.65**, **89.87**)	Ours
BERT	mBERT	(79.5, 89.57)	-
100% of **Intent Classification** dataset (Larson et al., 2019)			
BERT	BERT-base	94.3	
Others	CNN	89.8	(2019)
	MLP	90.1	
Fusion	CNN + BiLSTM + CNN	**93.62**	Ours
	CNN + LSTM + CNN	**93.28**	
100% of the **Vietnamese** dataset (Ho et al., 2019)			
Fusion	CNN + LSTM + CNN	54.76	
	CNN + BiLSTM + CNN	54.54	Ours
BERT	BERT-base	53.18	
100% of the **Indonesian** dataset (Saputri et al., 2018)			
Fusion	CNN + LSTM + CNN	54.76	
	CNN + BiLSTM + CNN	54.54	Ours
BERT	BERT-base	53.18	

Position-sensitive self-attention role of fusion models in new Bengali corpus: An attention layer may aid in capturing the necessary information for a sequence to sequence model. We also investigated how adding a self-attention layer to the fusion model affects the accuracy of the the newly developed 6-class Bengali emotion corpus. However, an immediate question arises—what the optimal position of the attention layer be within a fusion chain. To answer this, we execute four different experiments, utilizing a self-attention layer in four alternative places: between the embedding and the first CNN layer, between the first CNN layer and the first LSTM layer, between the first LSTM layer and the second CNN layer, and between the second CNN layer and the final output

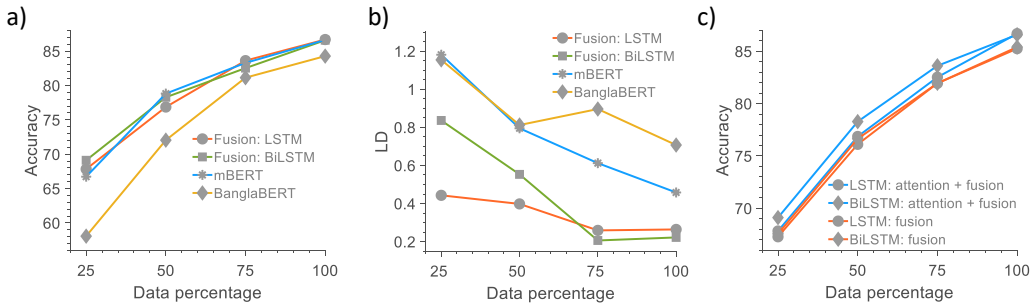

Figure 3: Performance comparison between the fusion (CNN + attn. + LSTM/BiLSTM + CNN) and mBERT model on 25%, 50%, 75% and 100% of a new 6-class Bengali emotion dataset. The dataset was split randomly to produce an artificial scarcity. In Fig. 3a-b, the green (square), red (circle), blue (asterisks), and yellow (diamond) lines represent CNN + attn. + LSTM + CNN (Fusion: LSTM), CNN + attn. + BiLSTM + CNN (Fusion: BiLSTM), mBERT and BanglaBERT models' performance, respectively. a) Accuracy comparison of all the four models for varying data size. b) The loss difference (LD) progression for different data sizes– the smaller the loss, the better the performance is. c) An inclusion of a self-attention layer improves fusion models' performance (blue lines).

Table 7: Deployable form for a few DNN-based fusion models before and after the pruning and retraining for the six-class Bengali emotion dataset developed in this study.

Serial	Fusion architecture	Retrained Accuracy	Accuracy		Size (zip, MB)	
			Before pruning	After Pruning	Before Pruning	After Pruning
1	LSTM + LSTM	86.19	85.43	85.18	33.45	6.32
2	**CNN + LSTM + CNN**	**86.36**	86.61	85.54	34.81	6.60
3	LSTM + CNN + LSTM	85.28	85.74	84.24	34.27	6.45

layer. As observed, the model provides an accuracy of 85.79% and a loss difference of 0.205 if the attention layer is placed between the embedding and the first DNN layer. Interestingly, the accuracy increased to 86.68%, and the loss difference reduced to 0.164 if the attention layer posits between the first CNN and first LSTM layer. It was the highest accuracy produced and the lowest loss difference of 0.164 among the alternative self-attention position tried. An attention layer between the LSTM and the second CNN layers generates shape mismatch and stops the model from training. Final experiment that places attention between the second CNN and output layer produces an accuracy of 85.79% with a 0.285 loss difference. These experiments show that for the 6-class Bengali emotion classification, a position-sensitive attention layer makes a difference in classification accuracy and reduces overfitting error. The accuracy improvement because of the self-attention layer still holds if an artificial scarcity for the new corpus is produced by considering 25%, 50%, 75% of the complete dataset, as shown in Fig. 3c. However, further analysis with other datasets and languages would clarify whether self-attention layer roles, as observed here in Fig. 3, are context-dependent or generic, and are beyond the scope of this study.

Fusion models robustly perform in data scarcity: One intriguing query on the fusion model would be to assess its ability to perform in data scarcity. An experiment designed to compare how the proposed fusion models and mBERT perform in data scarcity randomly segregates the Bengali 6-class emotion dataset into 25%, 50%, 75%, and 100% categories. The artificial data scarcity is analogous to the low-resource contexts, mimicking the lack of sufficient annotated data common for many low-resource languages. The comparison considers CNN + attn. + LSTM + CNN and CNN + attn. + BiLSTM + CNN and compare with mBERT. The fusion models perform better for the 25% case and match or surpass the mBERT performance in other scarce data cases (shown in Fig. 3a). Besides, the fusion models decrease LD in all the artificially produced scarce cases studied. A close comparison (Fig. 3b) shows that the LD of mBERT (blue line) remains way above the LDs reported by the fusion models. For the 25% case, the LD value is doubled for mBERT, indicating an advantage of fusion models in low-resource contexts.

Fusion models are computationally less expensive: Along with other factors, the computational cost of an NLP model also depends on its size and the FLOPs count. A comparison of these metrics between the baseline models and the fusion models exhibits that fusion models are more advantageous for a small number of annotated samples (shown in Table 2). For instance, the fusion model CNN

+ BiLSTM/LSTM + CNN roughly does 100 times fewer FLOPs. Also, for most GLUE datasets, the fusion model outperforms the TinyBERT in the presence of data scarcity. Some of the BERT models demonstrate equal accuracy for some GLUE benchmark datasets. However, these models are computationally extensive because of their high #Params and #FLOPs. Although costs related to FLOPs are decreasing, it requires hardware upgradation from GPU to TPU. Whereas the GPU itself is a computationally extensive device in low-resource environments, let alone the use of TPU. So, the low #FLOPs requirement in CNN + BiLSTM + CNN provides an edge over the memory-hungry BERT models in low-resource contexts. Besides, the possibility of a low computational cost of the CNN + BiLSTM + CNN model can also be predicted by comparing the average time per epoch calculation, an ensemble representation of all the individual times per epoch for alternative GLUE benchmark data considered. The average time per epoch over GLUE benchmark data is about 3 seconds for the CNN + BiLSTM + CNN model. In contrast, the same becomes as high as 1000 seconds or more for the different baseline models implemented in the experiment.

Besides, pruning and retraining reduce the fusion models further and increase their deployability in low-end devices and web platforms. Precisely, the CNN + LSTM + CNN model achieves almost a $5\times$ reduction in size from 34.81MB to a model size of 6.60MB, as in Table 7. The TinyBERT model may be as small as about 16MB, but it is pre-trained in the English language requiring further tuning in other languages for better accuracy. For instance, in experiments on a Bengali 6-class emotion dataset, the TinyBERT, pre-trained in English, achieves an accuracy of 33.42%. This accuracy drops to 24% if annotated data is reduced to 25%. So, TinyBERT requires training of the pre-trained model and suffers because of data scarcity. Whereas, for the proposed fusion model CNN + BiLST/LSTM + CNN, the initial accuracy (86.61) is almost retrievable (86.36) upon pruning and retraining (data shown in Table 7). Also, the model size reduces to around 5MB after pruning compared to the 16MB of the pre-trained TinyBERT.

5 Conclusion

Generally, the RNN and CNN models are computationally less intensive but compromise accuracy in textual classification. In contrast, BERT-variants and other advanced transformer-based implementations demonstrate improved performance but are computationally intensive. This study analyzed a few low-resource textual classification contexts to identify CPU-trainable and comparatively smaller deployable DNN models sufficiently accurate in textual classification tasks. These identified less-intensive DNN fusion models attained accuracy that frequently surpasses BERT performance in low-resource contexts. Interestingly, the efficacy of CNN + BiLSTM + CNN remains equally applicable in other alternative languages, tasks. This study also demonstrates that the fusion models are all CPU-trainable, making them easily accessible for communities suffering from an infrastructural deficiency. Moreover, low-resource languages always suffer from smaller corpus, infrequent research initiatives, and a lack of intensive computational facilities. These hinder the potential deployment of DNN models to monitor toxic and abusive elements in the ever-increasing social media platforms. Because of its relatively small size and acceptable classification accuracy, the fusion models are a suitable alternative to computationally intensive BERT variants for deployment in low-end devices.

Further improvement of the fusion models may consider a multichannel word-embedding technique, equipping the models better for out of vocabulary words now common in the era of social media platforms, POS-tagging to exploit the key phrases of the sentiment better. Such extensions, alone or in a cohort, can improve the fusion models to tackle the long-term dependencies analysis by forming phrases from the dependent and related words in longer sentences. Overall, this work provides sufficiently accurate, computationally less intensive CPU-trainable DNN models for NLP tasks for low-resource languages and may serve as the blueprint to identify the deployable NLP models for low-resource languages and environments.

Acknowledgements

We express our gratitude to Dr. Shafin Rahman, Department of Electrical and Computer Engineering at the North South University, Bangladesh and all the anonymous reviewers for their sincere comments, suggestions, and criticisms.

References

Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load what you need: Smaller versions of multilingual bert. *arXiv preprint arXiv:2010.05609*.

Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, and U Rajendra Acharya. 2021. Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems*, 115:279–294.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs. *University of Waterloo*, pages 1–7.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

François Chollet et al. 2018. Keras: The python deep learning library. *Astrophysics source code library*, pages ascl–1806.

Avishek Das, Omar Sharif, Mohammed Moshiul Hoque, and Iqbal H Sarker. 2021. Emotion classification in a resource constrained language using transformer-based approach. *arXiv preprint arXiv:2104.08613*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Daniel Grießhaber, Johannes Maucher, and Ngoc Thang Vu. 2020. Fine-tuning bert for low-resource natural language understanding via active learning. *arXiv preprint arXiv:2012.02462*.

Song Han, Huizi Mao, and William J Dally. 2015a. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

Song Han, Jeff Pool, John Tran, and William J Dally. 2015b. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*.

Vong Anh Ho, Duong Huynh-Cong Nguyen, Danh Hoang Nguyen, Linh Thi-Van Pham, Duc-Vu Nguyen, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. 2019. Emotion recognition for vietnamese social media text. In *International Conference of the Pacific Association for Computational Linguistics*, pages 319–333. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Md Karim, Sumon Kanti Dey, Bharathi Raja Chakravarthi, et al. 2020. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. *arXiv preprint arXiv:2012.14353*.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Daniel Quang and Xiaohui Xie. 2016. Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic acids research*, 44(11):e107–e107.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Nauros Romim, Mosahed Ahmed, Hriteshwar Talukder, and Md Saiful Islam. 2021. Hate speech detection in the bengali language: A dataset and its baseline evaluation. In *Proceedings of International Joint Conference on Advances in Computational Intelligence*, pages 457–468. Springer.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Mei Silviana Saputri, Rahmad Mahendra, and Mirna Adriani. 2018. Emotion classification on indonesian twitter dataset. In *2018 International Conference on Asian Language Processing (IALP)*, pages 90–95. IEEE.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Moshe Wasserblat, Oren Pereg, and Peter Izsak. 2020. Exploring the boundaries of low-resource bert distillation. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 35–40.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272.

Rui Zhang, Honglak Lee, and Dragomir Radev. 2016. Dependency sensitive convolutional neural networks for modeling sentences and documents. *arXiv preprint arXiv:1611.02361*.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*.

Xiaodan Zhu, Parinaz Sobihani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *International Conference on Machine Learning*, pages 1604–1612. PMLR.