Technical Domain Classification of Bangla Text using BERT

Koyel Ghosh*, Dr. Apurbalal Senapati

Department of Computer Science and Engineering, Central Institute of Technology, Assam.

*Corresponding author

doi: https://doi.org/10.21467/proceedings.115.16

ABSTRACT

Coarse-grained tasks are primarily based on Text classification, one of the earliest problems in NLP, and these tasks are done on document and sentence levels. Here, our goal is to identify the technical domain of a given Bangla text. In Coarse-grained technical domain classification, such a piece of the Bangla text provides information about specific Coarse-grained technical domains like Biochemistry (bioche), Communication Technology (com-tech), Computer Science (cse), Management (mgmt), Physics (phy) Etc. This paper uses a recent deep learning model called the Bangla Bidirectional Encoder Representations Transformers (Bangla BERT) mechanism to identify the domain of a given text. Bangla BERT (Bangla-Bert-Base) is a pretrained language model of the Bangla language. Later, we discuss the Bangla BERT accuracy and compare it with other models that solve the same problem.

Keywords: Coarse-grained Technical Domain Classification, Technical Domain Classification, BERT, Bangla BERT, Bangla language, Transformers.

1 Introduction

This sentiment classification, text classification, domain classification Etc., are the most popular NLP tasks since old time. So, there have been huge ups and downs happening on the progress of these research topics. Most of the approaches have focused on binary classification, mainly because the IMDB dataset [1] is readily available, but the entire game is changing slowly. Now lots of datasets are published and available. Multiclass, multilabel, fine-tuning Etc., are becoming the new challenge day by day. Coarse-grained technical domain classification (document or sentence level) usually is the task to identify the technical domain from a text. Here, we are working with Bangla TechDOfication 2020 dataset. Recently, a deep neural network architecture, i.e. transformer, has been widely used in Natural Language Processing (NLP) tasks and surprisingly produces high scores in the English dataset. Jacob Devlin proposed the BERT model [2] at Google AI Language. BERT is a transformers model pre-trained on a large corpus data in a self-supervised fashion.

This paper aims to identify the technical domain of a given Bangla text. We will discuss the Bangla TechDOfication 2020 dataset in Section 3. We will shortly discuss BERT and Bangla BERT in Section 4, which is used for the classification. Finally, we present the result and comparisons of models in section 5.

2 Related Work

Previously several classical machine-learning algorithms were used on text classification. This task demanded manually extracting features like the bag of words, bi-gram, tri-gram or n-grams. Then those selected features are used as inputs to classification algorithms such as Hidden Markov Models (HMM), Naive Bayes (NB), Support Vector Machine (SVM), random forests Etc, as discussed in [3]-[5]. After that, various neural models like convolution models [6]-[11], recurrent models [12]-[13] and attention mechanisms [14]-[15] have been

© 2021 Copyright held by the author(s). Published by AIJR Publisher in the "Proceedings of Intelligent Computing and Technologies Conference" (ICTCon2021) March 15th–16th, 2021. Jointly organized by Assam Science and Technology University (ASTU), and Central Institute of Technology Kokrajhar (CITK).

Proceedings DOI: 10.21467/proceedings.115; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-947843-5-7

used widely. Pretrained models on large corpus are effective for classifications and other NLP tasks, reducing new models' training time. Very popular word-embeddings, like [16] and GloVe [17] or ELMo [18] are the example of pre-trained models. A fine-tuning method for a pre-trained language model, i.e. VLM-FiT, has been proposed by Howard and Ruder (2018). Pre-trained language models like BERT are trained on a large amount of unlabeled data, which have helped for classifications. There are a lot of work already done successfully [19] in the coarse-grained with sentiment [20] and emotion analysis [21] dataset. Often, in the classification task, Word2Vec or fasttext or GloVe or all-combined approach [22] is used to utilize different word embedding algorithms' effectiveness. ESIM with SuBiLSTM (Ensemble) and ESIM with SuBiLSTM-Tied (Ensemble) approaches [23] performed well on the Stanford Sentiment Treebank dataset [24], both in its binary (SST-2) and fine-grained (SST-5) forms. A similar approach is applied to the question classification i.e TREC dataset [25], both in its 6 class(TREC-6) and 50 class (TREC-50) forms. Bidirectional dilated LSTM with attention [26] is used for another fine-grained dataset [27]. Paper [28] proposed a BiLSTM neural network architecture based on Word2vec's CBOW architecture. Some very old approaches on domain classification are in [29]. Now, this [30] is based on accident causes classification with the approach of deep learning and Word2Vec. They compare their model with others where bi-gram, n-gram was used for text representation. In [31], the author added an attention layer with BiLSTM for short text fine-grained sentiment classification to get better accuracy.

3 Dataset

Here,Technical Domain Identification dataset[6] in its Coarse-grained Domain Classification - Bangla (Subtask-1b) form has been used. Table 1 shows the details of the dataset. Total samples are 53,574, and we split the dataset into 80% training set and 20% test set. Coarse-grained Domain Classification is a piece of text that provides information about specific Coarse-grained domains. The first column contains text, and the second column has information regarding the label (domain). Number of the domains or classes are Biochemistry (bioche), Communication Technology (com-tech), Computer Science (cse), Management (mgmt), Physics (phy). Table 2 shows the distribution of the domains in the dataset.

Table 1: Bangla TechDOfication2020 *Dataset*

Dataset	**Total**	**Train**	**Test**	**label**
Bangla TechDOfication2020	53,574	42,859	10,715	5

Table 2: Bangla TechDOfication2020 *Dataset statistics*

Domain	**Train**	**Test**
Biochemistry (bioche)	2,759	741
Communication Technology (com-tech)	10,968	2,774
Computer Science (cse)	11,050	2,782
Management (mgmt)	8,054	1,946
Physics (phy)	10,028	2,472

[6] https://ssmt.iiit.ac.in/techdofication.html

4 Methodology

Coarse-grained Domain Classification uses a natural language text as input and produces output among the domains {bioche, com-tech, cse, mgmt, phy}. After text preprocessing is performed, we use the pretrained Bangla BERT model to build the classification model. In this section, we briefly illustrate BERT and then explain Bangla BERT model. The architecture of the used method is shown in Figure 1.

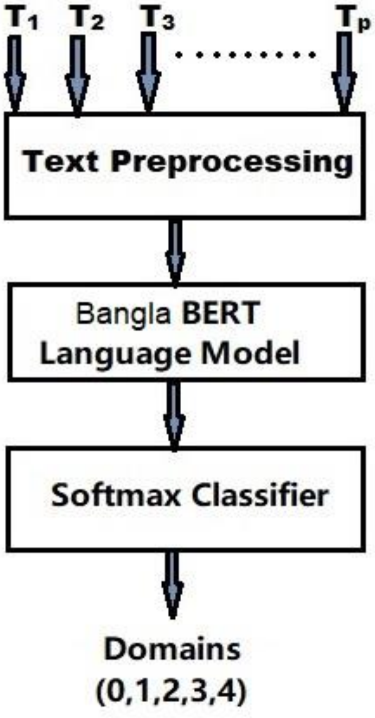


Figure 1: *The methodology followed for Technical Domain Classification*

4.1 BERT

Multi-layer bidirectional Transformer encoder, based on the original implementation described in "Attention is all you need" [32] is the BERT's model basic architecture. The BERT model is described as self-supervised learning. There are two BERT models, BERT-base and BERT-large. The numbers of layers, numbers of hidden units, number of self-attention heads and total trainable parameters are listed for both in Table 3.

Table 3: *BERT-base vs. BERT-large*

	BERT-base	**BERT-large**
Numbers of layers	12	24
Numbers of hidden units	768	1024
Number of self-attention heads	12	16
Total trainable parameters	110M	340M

Input representation:

A [CLS], i.e. classification token, is inserted at the beginning of the first sentence, and a [SEP], i.e. separation token, is inserted at the end of each sentence.

There are two steps in this architecture:

a. Pre-training

The model is trained on unlabeled texts(such as large unsupervised text corpus comprising the Toronto Book Corpus and Wikipedia dump) over various pre-training tasks. The pre-training task combines two unsupervised tasks Masked Language Modeling(Masked LM) and Next Sentence Prediction(NSP).

- **Masked Language Modeling (Masked LM) :**
  Here, 15% of the WordPiece tokens in each input sequence is masked out, i.e. replaced with [MASK] token at random. After feeding the entire sequence to a deep bidirectional Transformer encoder, the model learns to predict the actual value of the masked tokens based on the context produced by the other non-masked words in the sequence.
- **Next Sentence Prediction(NSP) :**
  It takes two sentences to learn the relationship between sentences and classify, whether the second sentence follows the first sentence or just a random sentence. 50% of the time the second sentence is the following actual sentence that follows the first sentence (labeled as IsNext), and it is a random sentence from the corpus (labeled as NotNext) 50% of the time. To minimise the combined loss function of the two strategies, Masked LM and Next Sentence Prediction are trained together.

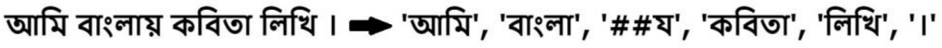
b. Fine-tuning

The BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks.

4.2 Preprocessing

Text preprocessing is a crucial step to get high accuracy in NLP tasks. Here, we perform the following steps before the classification model.

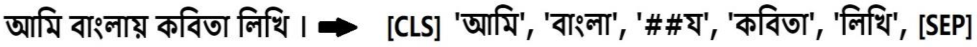
- **Tokenization :**
  We used AutoTokenizer to divide the texts into words from a pretrained Bangla BERT model, as mentioned in [33]. It breaks the words into their prefix root and suffix, handling the unseen words. Figure 2 shows the tokenization example.



আমি বাংলায় কবিতা লিখি। ➡ 'আমি', 'বাংলা', '##য', 'কবিতা', 'লিখি', '।'

Figure 2: *Tokenization Example*

- **Special token Addition :**
  [CLS] and [SEP] tokens have been added to their correct position, as mentioned earlier. After this step, sentences will be looked like, as shown in Figure 3.



আমি বাংলায় কবিতা লিখি। ➡ [CLS] 'আমি', 'বাংলা', '##য', 'কবিতা', 'লিখি', [SEP]

Figure 3: *After the addition of Special tokens ([CLS] and [SEP])*

- **Label encoding :**
  As labeled domains on the texts are words so, we need to encode them into a unique number. Like, bioche- 0, com-tech- 1, cse- 2, mgmt- 3, phy- 4 for subtask 1b.

4.3 Bangla BERT

For the Coarse-grained Technical Domain Classification, we used Bangla BERT[7], which Sagor Sarker[8] proposes. As described in [33], bangla-bert-base is a pretrained language model of Bengali language using Masked Language Modeling described in BERT [2]. **Bengali commoncrawl corpus from OSCAR** and **Bengali Wikipedia Dump Dataset** are used as Pre-training Corpus in this case.

They preprocessed the downloaded corpus according to the BERT format with one sentence per line and an extra newline for new documents. BNLP[9] package, i.e. natural language processing toolkit for the Bengali Language, is used for training the Bengali SentencePiece model to build vocabulary with vocab size 1,02,025 and vocabulary file is converted according to the BERT format.

Bangla-BERT was trained on a single Google Cloud TPU with code provided in Google BERT's GitHub repository[10]. The currently available model follows the bert-base-uncased model architecture using a Masked LM, whose parameters are similar to BERT-base as mentioned in Table 3. This model is uncased as it does not make a difference between uppercase and lowercase.

5 Experiments and Result

In this section, we discuss the Bangla Bert model result and compare it with Bengali Electra, Multilingual BERT and Indic Transformers Bangla BERT models that solve the same problem, i.e. technical domain classification on the Bangla Coarse-grained technical domain classification dataset.

5.1 Comparison Models

- **Bengali Electra:**
  bangla-electra[11] is trained on OSCAR and Bengali Wikipedia Dump Dataset. Its vocabulary size is 29,898.
- **Multilingual BERT:**
  In the BERT multilingual base model (uncased)[2], the top 102 languages with the largest Wikipedia are used for the Pretrained model with Masked LM objective.
- **Indic Transformers Bangla BERT:**
  Indic-transformers-bn-bert[12] is a BERT language model which is pre-trained on ~3 GB of the monolingual training corpus. The pre-training data was majorly taken from Open Super-large Crawled ALMAnaCH coRpus (OSCAR).

[7] https://huggingface.co/sagorsarker/bangla-bert-base

[8] https://github.com/sagorbrur

[9] https://github.com/sagorbrur/bnlp

[10] https://github.com/google-research/bert

[11] https://huggingface.co/monsoon-nlp/bangla-electra

[12] https://huggingface.co/neuralspace-reverie/indic-transformers-bn-bert

5.2 Evaluation Metric

Here, we use the accuracy measure to evaluate Bangla BERT's performance and compare it with the other three. Firstly, we found the number of wrong predictions (W). Then we find out the number of correct predictions subtracting the number of the wrong predictions (W) to the total number of predictions (T). The accuracy is defined as follows:

$$Accuracy(\%) = \frac{T - W}{T} \times 100 \qquad (1)$$

5.3 Results

All the results and comparisons are shown in Table 4. It gives the accuracy of all models on the Bangla Coarse-grained technical domain classification dataset. We can see that, among all, Bangla BERT achieved the highest accuracy.

Table 4: *Accuracy(%) of Bangla BERT on Bangla TechDOfication 2020 dataset*

Model	Accuracy
Bengali Electra	75.3
Multilingual BERT	74.5
Indic Transformers Bangla BERT	67.8
Bangla BERT	84.05

6 Conclusions

It is challenging to deal with any regional languages in India like Bangla, Assamese, Bodo, Tamil, Telugu Etc. Most of the work has been done with the European languages. So, regional languages are yet to be explored more. This paper used the pretrained Bangla BERT model for technical domain classification on Bangla TechDOfication 2020 dataset. We can see from Table 4 Bangla BERT performs better than other models. In the future, challenges are handling the mixed data, as this dataset has some English and regional language's word in it. In most studies, people keep only the primary language alphabets and remove the other but doing this performance decreases slightly as removed words also have some extra meaning in sentences.

References

[1] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[3] L. Manevitz, M. Yousef, N. Cristianini, J. Shawe-Taylor, and B. Williamson, "One-class svms for document classification", *Journal of machine learning research 2 (2001) 139-154 submitted 3/01; published 12/01 one-class svms for document classification*, 01 2002.

[4] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002).* Association for Computational Linguistics, Jul. 2002, pp. 79–86.[Online]. Available: https://www.aclweb.org/anthology/W02-1011

[5] G. Forman, "Forman, g.: An extensive empirical study of feature selection metrics for text classification. journal of machine learning research 3, 1289-1305,"*Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, 01 2003.[Online]. Available: https://dl.acm.org/doi/10.5555/944919.944974

[6] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Baltimore, Maryland: Association for Computational Linguistics, June. 2014, pp. 655–665. [Online]. Available: https://www.aclweb.org/anthology/P14-1062

[7] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification,"*CoRR*, vol. abs/1509.01626, 2015. [Online].Available: http://arxiv.org/abs/1509.01626

[8] A. Conneau, H. Schwenk, L. Barrault, and Y. LeCun, "Very deep convolutional networks for natural language processing," *CoRR*, vol. abs/1606.01781,2016. [Online]. Available: http://arxiv.org/abs/1606.01781

[9] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp.562–570. [Online]. Available: https://www.aclweb.org/anthology/P17-1052

[10] Y. Zhang, D. Shen, G. Wang, Z. Gan, R. Henao, and L. Carin, "Deconvolutional paragraph representation learning," *CoRR*, vol. abs/1708.04729, 2017.[Online]. Available: http://arxiv.org/abs/1708.04729

[11] D. Shen, Y. Zhang, R. Henao, Q. Su, and L. Carin, "Deconvolutional latent-variable model for text sequence matching," *CoRR*, vol. abs/1709.07109, 2017. [Online]. Available: http://arxiv.org/abs/1709.07109

[12] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning,"*CoRR*, vol. abs/1605.05101, 2016. [Online].Available: http://arxiv.org/abs/1605.05101

[13] D. Yogatama, C. Dyer, W. Ling, and P. Blunsom, "Generative and Discriminative Text Classification with Recurrent Neural Networks," *arXiv e-prints*, p. arXiv:1703.01898, Mar. 2017.

[14] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June. 2016, pp. 1480–1489. [Online]. Available: https://www.aclweb.org/anthology/N16-1174

[15] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *CoRR*, vol. abs/1703.03130, 2017. [Online]. Available: http://arxiv.org/abs/1703.03130

[16] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2,* ser. NIPS'13. Red Hook, NY, USA: Curran Associates Inc., 2013, p. 3111–3119. [Online]. Available: https://dblp.org/rec/journals/corr/MikolovSCCD13.bib

[17] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online].Available: https://www.aclweb.org/anthology/D14-1162

[18] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).* New Orleans, Louisiana: Association for Computational Linguistics, June. 2018, pp. 2227–2237. [Online]. Available: https://www.aclweb.org/anthology/N18-1202

[19] M. S. Akhtar, A. Ekbal, and E. Cambria, "How intense are you? predicting intensities of emotions and sentiments using stacked ensemble [application notes],"*IEEE Computational Intelligence Magazine,* vol. 15, no. 1, p. 64–75, Feb. 2020. [Online]. Available: https://doi.org/10.1109/MCI.2019.2954667

[20] K. Cortis, A. Freitas, T. Daudert, M. Huerlimann, M. Zarrouk, S. Handschuh, and B. Davis, "SemEval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017).* Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 519–535. [Online]. Available: https://www.aclweb.org/anthology/S17-2089

[21] S. Mohammad and F. Bravo-Marquez, "WASSA-2017 shared task on emotion intensity," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis.* Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 34–49. [Online]. Available: https://www.aclweb.org/anthology/W17-5205

[22] M. Salur and I. Aydin, "A novel hybrid deep learning model for sentiment classification," *IEEE Access*, vol. 8, pp. 58 080 – 58 093, 04 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9044300

[23] S. Brahma, "Improved sentence modeling using suffix bidirectional lstm," *arXiv*: Learning, 2018. [Online]. Available: http://arxiv.org/abs/1805.07340

[24] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,* vol. 1631, pp. 1631–1642, 01 2013. [Online]. Available: https://www.aclweb.org/anthology/D13-1170

[25] E. Voorhees, “The trec question answering track,” *Nat. Lang. Eng,* vol. 7, pp. 361–378, 01 2006, [Online]. Available:*https://doi.org/10.1017/S1351324901002789*

[26] A. M. Schoene, A. P. Turner, and N. Dethlefs, “Bidirectional dilated lstm with attention for fine-grained emotion classification in tweets,” in *AffCon@AAAI*, 2020

[27] R. Klinger, O. de clercq, S. Mohammad, and A. Balahur, “Iest: Wassa-2018 implicit emotions shared task,” in *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 09 2018. [Online]. Available: https://www.aclweb.org/anthology/W18-6206

[28] O. Melamud, J. Goldberger, and I. Dagan, “context2vec: Learning generic context embedding with bidirectional lstm,” in *Proceedings of The 20th {SIGNLL} Conference on Computational Natural Language Learning, 05,* 2016, pp. 51–61 [Online]. Available: https://www.aclweb.org/anthology/K16-1006

[29] G. Bernier-Colborne, C. Barriere, and P. A. Menard, “Fine-grained domain classification of text using TERMIUM plus,” in *Proceedings of Language, Ontology, Terminology and Knowledge Structures Workshop ({LOTKS} 2017), Association for Computational Linguistics,* 09 2017. [Online]. Available: https://www.aclweb.org/anthology/W17-7005

[30] F. Zhang, “A hybrid structured deep neural network with word2vec for construction accidents causes classification,” *International Journal of Construction Management,* pp. 1–21, 11 2019.

[31] J. Xie, B. Chen, X. Gu, F. Liang, and X. Xu, “Self-attention-based bilstm model for short text fine-grained sentiment classification,”*IEEE Access,* vol. 7,pp.180558-180570, 12 2019

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA 2017. [Online].Available: https://arxiv.org/pdf/1706.03762.pdf

[33] S. Sarker, “Banglabert: Bengali mask language model for bengali language understanding,” 2020. [Online]. Available: https://github.com/sagorbrur/bangla-bert