Multilingual Pre-Trained Transformers and Convolutional NN Classification Models for Technical Domain Identification

Suman Dowlagar
LTRC
IIIT-Hyderabad
suman.dowlagar@research.iiit.ac.in

Radhika Mamidi
LTRC
IIIT-Hyderabad
radhika.mamidi@iiit.ac.in

Abstract

In this paper, we present a transfer learning system to perform technical domain identification on multilingual text data. We have submitted two runs, one uses the transformer model BERT, and the other uses XLM-ROBERTa with the CNN model for text classification. These models allowed us to identify the domain of the given sentences for the ICON 2020 shared Task, TechDOfication: Technical Domain Identification. Our system ranked the best for the subtasks 1d, 1g for the given TechDOfication dataset.

1 Introduction

Automated technical domain identification is a categorization/classification task where the given text is categorized into a set of predefined domains. It is employed in tasks like Machine Translation, Information Retrieval, Question Answering, Summarization, and so on.

In Machine Translation, Summarization, Question Answering, and Information Retrieval, the domain classification model will help leverage the contents of technical documents, select the appropriate domain-dependent resources, and provide personalized processing of the given text.

Technical domain identification comes under text classification or categorization. Text classification is one of the fundamental tasks in the field of NLP. Text classification is the process of identifying the category where the given text belongs. Automated text classification helps to organize unstructured data, which can help us gather insightful information to make future decisions on downstream tasks.

Traditional text classification approaches mainly focus on feature engineering techniques such as bag-of-words and classification algorithms (Yang, 1999). Nowadays, the sate-of-the-art results on text classification are achieved by various NNs such as CNN (Kim, 2014), LSTM (Hochreiter and Schmidhuber, 1997), BERT (Adhikari et al., 2019), and Text GCN (Adhikari et al., 2019). Attention mechanisms (Vaswani et al., 2017) have been introduced in these models, which increased the representativeness of the text for better classification. Transformer models such as BERT (Devlin et al., 2018) uses the attention mechanism that learns contextual relations between words or sub-words in a text. Text GCN (Yao et al., 2019) uses a graph-convolutional network to learn a heterogeneous word document graph on the whole corpus, which helped classify the text. However, of all the deep learning approaches, transformer models provided SOTA results in text classification.

In this paper, We present two approaches for technical domain identification. One approach uses the pre-trained Multilingual BERT model, and the other uses XLM-ROBERTa with CNN model.

The rest of the paper is structured as follows. Section 2 describes our approach in detail. In Section 3, we provide the analysis and evaluation of results for our system, and Section 4 concludes our work.

2 Our Approach

Here we present two approaches for the TechDOfication task.

2.1 BERT for TechDOfication

In the first approach, we use the pre-trained multilingual BERT model for domain identification of the given text. Bidirectional Encoder Representations from Transformers (BERT) is a transformer encoder stack trained on the large corpora. Like the vanilla transformer model (Vaswani et al., 2017), BERT takes a sequence of words as input. Each layer applies self-attention, passes its results through a feed-forward network, and then hands

Proceedings of the 17th International Conference on Natural Language Processing: TechDOfication 2020 Shared Task, pages 16–20
Patna, India, December 18 - 21, 2020. ©2020 NLP Association of India (NLPAI)

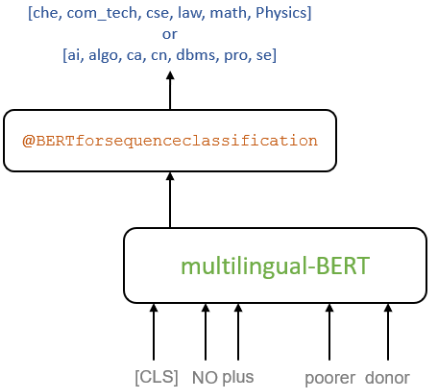

[che, com_tech, cse, law, math, Physics]
or
[ai, algo, ca, cn, dbms, pro, se]
@BERTforsequenceclassification
multilingual-BERT
[CLS]
NO plus
poorer
donor

Figure 1: The architecture of the BERT model for sentence classification.

it off to the next encoder. The BERT configuration model takes a sequence of words/tokens at a maximum length of 512 and produces an encoded representation of dimensionality 768.

The pre-trained multilingual BERT models have a better word representation as they are trained on a large multilingual Wikipedia and book corpus. As the pre-trained BERT model is trained on generic corpora, we need to finetune the model for the given domain identification tasks. During finetuning, the pre-trained BERT model parameters are updated.

In this architecture, only the [CLS] (classification) token output provided by BERT is used. The [CLS] output is the output of the 12th transformer encoder with a dimensionality of 768. It is given as input to a fully connected neural network, and the softmax activation function is applied to the neural network to classify the given sentence.

2.2 XLM-ROBERTa with CNN for TechDOfication

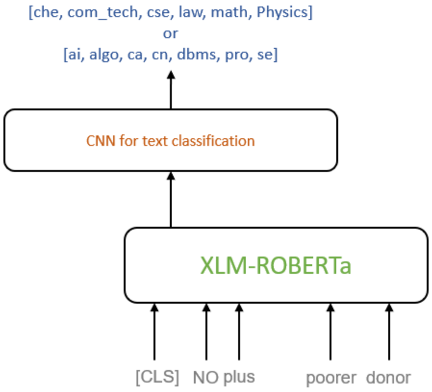

[che, com_tech, cse, law, math, Physics]
or
[ai, algo, ca, cn, dbms, pro, se]
CNN for text classification
XLM-ROBERTa
[CLS]
NO plus
poorer
donor

Figure 2: The architecture of the XLM-ROBERTa with CNN for sentence classification.

XLM-ROBERTa (Conneau et al., 2019) is a transformer-based multilingual masked language model pre-trained on the text in 100 languages, which obtains state-of-the-art performance on cross-lingual classification, sequence labeling, and question answering. XLM-ROBERTa improves upon BERT by adding a few changes to the BERT model such as training on a larger dataset, dynamically masking out tokens compared to the original static masking, and uses a known preprocessing technique (Byte-Pair-Encoding) and a dual-language training mechanism with BERT in order to learn better relations between words in different languages. The given model is trained for the language modeling task, and the output is of dimensionality 768. It is given as input to a CNN (Kim, 2014) because convolution layers can extract better data representations than Feed Forward layers, which indirectly helps in better domain identification.

3 Experiment

This section presents the datasets used, the task description, and two models' performance on technical domain identification. We also include our implementation details and error analysis in the subsequent sections.

3.1 Dataset

We used the dataset provided by the organizers of TechDOfication ICON-2020. There are two subtasks, one is coarse-grained, and the other is fine-grained. The coarse-grained TechDOfication dataset contains sentences about Chemistry, Communication Technology, Computer Science, Law, Math, and Physics domains in different languages such as English, Bengali, Gujarati, Hindi, Malayalam, Marathi, Tamil, and Telugu. Whereas the fine-grained English dataset focuses on the Computer-Science domain with sub-domain labels as Artificial Intelligence, Algorithm, Computer Architecture, Computer Networks, Database Management system, Programming, and Software Engineering.

3.2 Implementation

For the implementation, we used the transformers library provided by HuggingFace[1]. The HuggingFace contains the pre-trained multilingual BERT, XLM-ROBERTa, and other models suitable

[1] https://huggingface.co/

for downstream tasks. The pre-trained multilingual BERT model used is *"bert-base-multilingual-cased"* and pre-trained XLM-R model used is *"xlm-roberta-base"*. We programmed the CNN architecture as given in the paper (Kim, 2014). We used the PyTorch library that supports GPU processing for implementing deep neural nets. The BERT models were run on the Google Colab and Kaggle GPU notebooks. We trained our classifier with a batch size of 128 for 10 to 30 epochs based on our experiments. The dropout is set to 0.1, and the Adam optimizer is used with a learning rate of 2e-5. We used the hugging face transformers pre-trained BERT tokenizer for tokenization. We used the BertForSequenceClassification module provided by the HuggingFace library during finetuning and sequence classification for the multilingual-BERT based approach.

3.3 Baseline models

Here, we compared the BERT model with other machine learning algorithms.

SVM with TF_IDF text representation We chose Support Vector Machines (SVM) with TF_IDF text representation for technical domain identification. SVM classifier and TF_IDF vector representation is obtained from the scikit-learn library (Pedregosa et al., 2011).

CNN: Convolutional Neural Network (Kim, 2014). We explored CNN-non-static, which uses pre-trained word embeddings.

3.4 Results

The results are tabulated in Table 1. We evaluated the performance of the method using macro F1. The multilingual-BERT model performed well when compared to the other SVM with TF-IDF and CNN models. Given all the languages, we have observed an increase of 7 to 25% in classification metrics for BERT compared to the baseline SVM classifier, it showed a 2 to 5% increase in classification metrics compared to the CNN classifier on the validation data. On the test data, multilingual BERT showed better performance in subtasks 1a, 1b, 1c, 1h and 2a whereas XLM-ROBERTa with CNN showed better performance in the subtasks 1d, 1e, 1f, 1g. This increase in classification metrics is due to the transformer model's and convolutional NN's capability, which learned better text representations from the generic data than other models.

4 Error Analysis

The multilingual-BERT model's confusion matrix is compared with the poorly performed model for languages, Hindi, and Tamil languages are shown in Figure 3. We chose Hindi and Tamil languages because, here, the difference in performance is more significant. For the Hindi subtask, the SVM classifier confused between "cse", "com_tech", and "mgmt" labels, whereas the BERT model performed better. For the Tamil subtask, the SVM classifier confused between "com_tech" and "mgmt" labels, whereas the BERT model performed better than the other models. This is because both the approaches (pre-trained multilingual-BERT and pre-trained XLM-ROBERTa with CNN) learned better representation of the above data than the other models that helped in technical document identification.

5 Conclusion and Future work

We used pre-trained bi-directional encoder representations using multilingual-BERT and XLM-ROBERTa with CNN technical domain identification for English, Bengali, Gujarati, Hindi, Malayalam, Marathi, Tamil, and Telugu languages. We compared the approaches with the baseline methods. Our analysis showed that pre-trained multilingual BERT and XLM-ROBERTa with CNN models and finetuning it for text classification tasks showed an increase in macro F1 score and accuracy metrics compared to baseline approaches.

Some datasets are large, like for the Hindi, Tamil, and Telugu, we can train the BERT and XLM-ROBERTa models from scratch and consider its hidden layer representation, and concatenate this with the representation of the pre-trained model. It might help to classify the datasets even better.

References

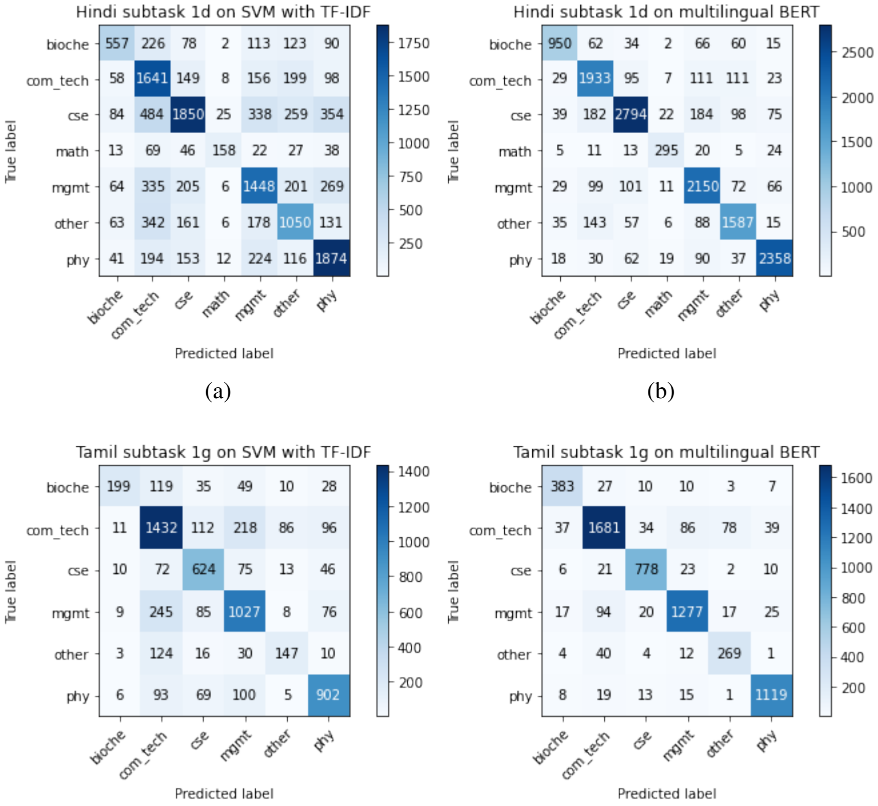
Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

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*.

	Classifier Models					
Dataset	**Validation**				**Test**	
	SVM	**CNN**	**M-Bert**	**XLM-R+CNN**	**M-Bert**	**XLM-R+CNN**
English subtask-1a	81.48	83.05	**88.87**	87.09	**79.84**	73.57
Bengali subtask-1b	66.35	85.78	**86.81**	85.71	**80.35**	78.17
Gujarati subtask-1c	69.63	86.27	**87.21**	86.89	**68.67**	66.73
Hindi subtask-1d	58.21	81.03	**83.40**	82.13	59.89	**60.44**
Malayalam subtask-1e	80.60	92.51	**94.72**	93.40	34.47	**34.86**
Marathi subtask-1f	73.32	86.89	**87.42**	86.37	59.52	**59.89**
Tamil subtask-1g	65.95	85.75	**87.50**	86.54	49.24	**51.34**
Telugu subtask-1h	71.98	88.07	**90.28**	89.43	**67.17**	62.26
English subtask-2a	70.24	72.53	**77.36**	76.77	**78.98**	78.07

Table 1: macro F1 on validation and test data for all the subtasks



(a) (b)

(c) (d)

Figure 3: Confusion matrix on the given validation data for the Hindi and Tamil languages

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1-2):69–90.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.