Fine-grained domain classification using Transformers

Akshat Gahoi **Akshat Chhajer** **Dipti Mishra Sharma**

Language Technologies Research Center
International Institute of Information Technology, Hyderabad, India

{akshat.gahoi,akshat.chhajer}@research.iiit.ac.in
dipti@iiit.ac.in

Abstract

The introduction of transformers in 2017 and successively BERT in 2018 brought about a revolution in the field of natural language processing. Such models are pretrained on vast amounts of data, and are easily extensible to be used for a wide variety of tasks through transfer learning. Continual work on transformer based architectures has led to a variety of new models with state of the art results. RoBERTa (Liu et al., 2019) is one such model, which brings about a series of changes to the BERT (Devlin et al., 2018) architecture and is capable of producing better quality embeddings at an expense of functionality. In this paper, we attempt to solve the well known text classification task of fine-grained domain classification using BERT and RoBERTa and perform a comparative analysis of the same. We also attempt to evaluate the impact of data preprocessing specially in the context of fine-grained domain classification.

The results obtained outperformed all the other models at the ICON TechDOfication 2020 (subtask-2a) Fine-grained domain classification task and ranked first. This proves the effectiveness of our approach.

1 Introduction

The transformer-based language models have been showing promising progress on a number of different natural language processing (NLP) benchmarks. The combination of transfer learning methods with large-scale transformer language models is becoming a standard in modern NLP and has resulted in many state-of-the-art models.

Compared to LSTMs(Greff et al., 2015), the main limitations of bidirectional LSTMs is their sequential nature, which makes training in parallel very difficult. The transformer architecture solves that by completely replacing LSTMs by the so-called attention mechanism (Vaswani et al., 2017). With attention, we are seeing an entire sequence as a whole, therefore it is much easier to train in parallel.

Text classification is a well-known task in Natural Language Processing, which aims at automatically providing additional document-level metadata (e.g. domain, genre, author).

One of text classification tasks: domain classification can be divided into two categories:-

- Course-grained domain classification
- Fine-grained domain classification

Course-grained domain classification aims to classify the input into varied and unrelated domains such as chemistry, law, computer science etc. On the other hand, fine-grained domain classification aims to classify the input into closely related subdomains under a higher level domain. Example includes classification of text on physics into different topics like relativity, mechaincs, or quantum mechanics. The latter is found to be a significantly more challenging task due to the similarity and lack of distinction between the inputs attributed to the fact that they are under an umbrella of a common domain.

Although there is substantive work done on domain classification in general (Young-Bum Kim, 2018), there has been less emphasis on fine-grained domain classification and the various augmentations to data that can be done to achieve higher performances in that task. This paper looks at the task of fine-grained domain classification in context of transformers. It paper will provide a comparison between the widely used **BERT** and **RoBERTa** embeddings for the task as well as attempt to observe the impact of data pre-processing in the context of fine-grained domain classification.

On blind test corpora of 1929 text samples, the proposed model in this paper led to F1 score of

Proceedings of the 17th International Conference on Natural Language Processing: TechDOfication 2020 Shared Task, pages 31–34
Patna, India, December 18 - 21, 2020. ©2020 NLP Association of India (NLPAI)

0.824 at the ICON TechDOfication 2020 shared task (subtask-2a). This result helped us bag the leading position on the leaderboard.

2 Dataset

For the study, dataset from the ICON TechDOfication 2020 (subtask-2a) was used. The entire collection consisted of 14910 text samples from English spanning across 7 sub-domains of **Computer Science**. Table 1 shows the sub-domains and the distribution of data.

Sub-domain	Code	Samples
Artificial Intelligence	ai	2140
Algorithm	algo	2131
Computer Architecture	ca	2127
Computer Networks	cn	2140
Database Management System	dbms	2140
Programming	pro	2122
Software Engineering	se	2140

Table 1: Dataset distribution across domains

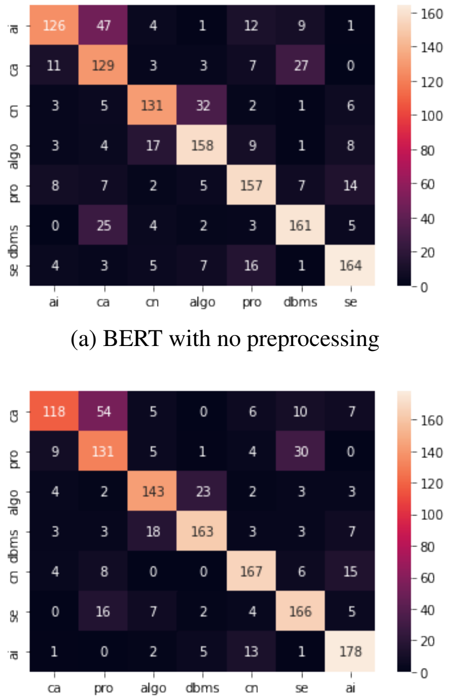
The average number of characters in the text samples was 177.6 and the average number of tokens observed in the same was 36.3.

A collection of 1929 text samples served as the blind test set for this task.

3 BERT vs RoBERTa

BERT is a bi-directional transformer for pre-training over huge amount of unlabeled textual data to learn a language representation. It can be then used to fine-tune for specific machine learning tasks like text classification. BERT outperformed the NLP state-of-the-art on several challenging tasks, attributed to the bidirectional transformer, novel pre-training tasks of Masked Language Model(Song et al., 2019) and Next Sentence Prediction(Shi and Demberg, 2019).

RoBERTa has a very similar architecture as compared to BERT with improved training methodology and more data. To improve the training, RoBERTa removes the Next Sentence Prediction task from BERT's pre-training and introduces dynamic masking so that the masked token changes during the training epochs. Originally BERT is trained for 1M steps with a batch size of 256 sequences. RoBERTa on the other hand is trained with 125 steps of 2K sequences and 31K steps with 8K sequences of batch size. Large batches are also easier to parallelize via distributed parallel training.



(a) BERT with no preprocessing

(b) RoBERTa with no preprocessing

Figure 1: Confusion matrices for dev set without preprocessing

4 System Overview

The section presents an overview of the system which was used to evaluate the scores described in the Results section of the paper.

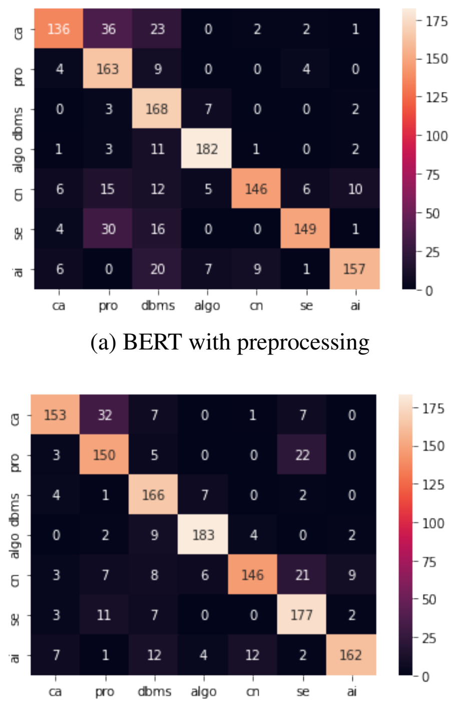
4.1 Pre-Processing

In the first approach, only one-hot-encoding for the labels was done and the raw text was fed as it is to both the transformers.

In the second approach the raw data was preprocessed keeping in mind the nature of fine-grained domain classification task. First, tokenization was done on the text using spaCy and the stop words were filtered out. Next, the tokens were passed through a counter and the top 20 tokens from the entire corpus were identified and then removed. As domain classification relies more on the keywords than the sentence structures, the data was cleaned. Lastly, the text was reconstructed from the remaining tokens. This was done to reduce the generalization amongst the sub-domains as the text had a lot of common terms from the higher level computer science domain itself.

4.2 Training

In total, 4 models were trained using BERT/RoBERTa and with/without pre-processing.



(a) BERT with preprocessing

(b) RoBERTa with preprocessing

Figure 2: Confusion matrices for dev set with preprocessing

The training was done using the concept of transfer learning. The pretrained bert-base-uncased and roberta-bert were taken and further fine tuning on it was done using the training dataset. The learning rate used was 4e-5 with 128 batch size. Each of the models were trained for 4 epochs.

5 Results and Evaluation

All the models were evaluated on the dev dataset and the results are presented in Table 2. It is clear that pre-processing indeed increases the f1 score and makes a substantive difference in fine-grained domain classification. This is because the common terms from the higher level common domains are removed and more distinction is created in the text samples for sub-domains.

In Figure 1 (results on dev set without preprocessing), we can see that RoBERTa miss classifies only slightly a less number text samples compared to BERT with both performing very similar. However, there is difference seen when preprocessing is done and frequent words are removed. In Figure 2 (results on dev set with preprocessing), we can see that BERT miss-classifies 91 text samples as '*Database Management System*' which reduces to 48 when using RoBERTa. Similarly, 87 miss-classifications done by BERT as '*Programming*' are corrected to 54 by RoBERTa. However it is seen that RoBERTa tends to miss-classify text samples as '*Software Engineering*' often.

In both the cases, RoBERTa performed better than BERT however, with a small margin. The best performing model (RoBERTa with preprocessing) was then evaluated using the ICON TechDOfication 2020 (subtask-2a) test dataset. The results obtained are shown in Table 3. It was observed that for different models also documents gets misidentified between a common pair of domains hence defining a close relation between the two domains. So this experiment can also be done to determine two closely related domains among a huge variety of domains.

Transformer	**Pre-processed**	**Precision**	**Recall**	**F1**
bert-base-uncased	no	0.761	0.752	0.756
roberta-base	no	0.788	0.783	0.781
bert-base-uncased	yes	0.832	0.812	0.810
roberta-base	yes	0.842	0.837	0.835

Table 2: Results for the dev set

Transformer	**Accuracy**	**Precision**	**Recall**	**F1**
RoBERTa	0.825	0.826	0.825	0.824

Table 3: Final model results on the test dataset

6 Conclusion

In this paper, we did a comparative analysis of BERT and RoBERTa in the context of fine-grained domain classification. Furthermore, the impact of pre-processing was also explored. It was found that pre-processing and removal of common terms from data helps the model perform better as more distinction is created between the sub-domains. The results indicate that RoBERTa performs slightly better than BERT in all the cases.

The model proposed in this paper ranked first in the ICON TechDOfication 2020 (subtask-2a) with an F1 score of 0.824.

7 Future Work

This paper shows how good transformers can perform for the task of multi class text classification. The main difference in the results comes from the embeddings being used. Thus, a very high performing multilingual model can be created if enough data and pre-trained language models are available for Indian languages. Hence, the goal can be to create BERT embeddings for other languages and extend the work done by AI4Bharat/IndicBERT (Kakwani et al., 2020). This can then not only be used for text classification tasks but also any other multi-lingual state-of-the-art natural processing task.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2015. LSTM: A search space odyssey. *CoRR*, abs/1503.04069.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Wei Shi and Vera Demberg. 2019. Next sentence prediction helps implicit discourse relation classification within and across domains. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5790–5796, Hong Kong, China. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. *CoRR*, abs/1905.02450.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Anjishnu Kumar Ruhi Sarikaya Young-Bum Kim, Dongchan Kim. 2018. Efficient large-scale neural domain classification with personalized attention.