Team Text Understanding and Analysis at PAN: Utilizing BERT Series Pre-training Model for Multi-Author Writing Style Analysis

Notebook for the PAN Lab at CLEF 2024

Yingzhou Huang, Leilei Kong

Foshan University, Foshan, China

Abstract

We propose a training model based on BERT series. This method uses sliding window technique to preprocess data sets to train and solve multi-author writing style analysis tasks. Our method is to combine adjacent paragraphs into a training sample, and effectively extract the characteristics of style changes between paragraphs. We conducted systematic training and evaluation on three multi-author writing style analysis datasets (easy, medium, and difficult) at different difficulty levels provided by the PAN organization. We obtained f1 scores of 0.993, 0.831 and 0.825 on the test set, respectively, which proved the effectiveness and robustness of the proposed method.

Keywords

Multi-author writing style analysis, BERT series, training sample

1. Introduction

The Multi-Author Writing Style Analysis task is designed to identify where authorship has changed in multiple author documents. In practical applications, we can analyze the author's writing style to determine the author's identity, verify whether the document has been tampered with, and whether the article is suspected of plagiarism [1] [2] [3].

Multi-author recognition research is mainly divided into two directions: traditional methods and deep learning-based methods [4]. The traditional method usually uses hand-selected features to distinguish text similarity, such as word frequency, sentence length, punctuation, etc. These methods work well in some simple scenarios, but struggle to handle more complex situations. A deep learning-based approach uses a neural network model to extract the text representation and calculate the similarity of the text representation. These methods typically include convolutional neural networks (CNNS), recurrent neural networks (RNNS), and attention mechanisms. In addition, some researchers also try to use pre-trained language models [5], such as BERT [6], GPT [7], etc to improve the performance of the models. In general, deep learning-based approaches [4] have performed well in multi-author recognition studies and are receiving increasing attention.

We created data samples using a sliding window. We encoded and classified the data using Bert [6]series models, such as Bert base, DeBERTa, ALBERT, and RoBERTa.

2. Method

In this study, we utilized four different pre-trained language models [5]: DeBERTa-base [8], DeBERTa-v3-large [8], ALBERT-large-v2 [9], and RoBERTa-base [10]. Each of them was specifically applied to training tasks of varying difficulty (easy, medium, hard). Our goal is to enhance the performance of the multi-author style detection task by leveraging the specific architectural advantages of these models. The reasons for choosing the above models for our experiments are as follows.

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

3545817606@daytoy.freeqiye.com (Y. Huang); kongleilei@fosu.edu.cn (L. Kong)
0009-0001-9239-3344 (Y. Huang); 0000-0002-4636-3507 (L. Kong)

DeBERTa-base and DeBERTa-v3-large introduce efficient attention mechanisms and improved sentence encoding strategies, making them particularly suitable for tasks that require complex language understanding [8].

ALBERT-large-v2, through parameter sharing techniques, maintains a large model capacity while reducing the number of parameters, making it suitable for processing large datasets and balancing performance and efficiency [9].

RoBERTa-base has shown excellent performance in a variety of natural language processing tasks through dynamic masked language model training, and its robustness makes it a reliable choice for experiments [10].

We designed targeted training strategies for each model to adapt to tasks of different difficulty levels. During the training process, we used precision, recall, and F1 score as the main evaluation metrics to ensure that the models can achieve optimal performance in detecting changes in author style.

3. Experiment

3.1. Dataset

The writing style change detection dataset provided by PAN@CLEF [1]includes three levels of difficulty:

1.Easy: The paragraphs of the documents cover various topics, allowing methods that utilize topic information to detect changes in authorship.

2.Medium: There is little diversity in the topics within the documents (although still present), forcing methods to focus more on style to effectively address the detection task.

3.Hard: All paragraphs in the documents are related to the same topic.

In the dataset provided by PAN, the label information available to participants includes the number of authors in the document and labels indicating whether there are changes in writing style between paragraphs. We segmented the documents in the dataset according to natural paragraphs and labels, and recalculated the quantities of the dataset. The statistical results are shown in the following table.

Table 1
Dataset size of three tasks.

Task	Easy		Medium		Hard	
	Train	Validation	Train	Validation	Train	Validation
num of documents	4200	900	4200	900	4200	900
num of text pairs	11065	2471	21913	4592	19015	4135

3.2. Data Processing

In this study, we first carried out preprocessing on the dataset. Initially, we conducted paragraph segmentation. We segmented the documents into natural paragraphs, treating each paragraph as an independent data item. Subsequently, we performed text pair extraction. Utilizing the sliding window method, we construct a text pair adjacent paragraphs. Finally, we read the corresponding label information for each text pair from the JSON file, which will be used for subsequent training and evaluation.

3.3. Experiment setting

In this experiment,We chose the CrossEntropyLoss as the loss function, which measures the error based on the similarity between probability distributions and is used in conjunction with the softmax activation function, making it suitable for classification tasks. The optimizer selected was AdamW, an improved version of the Adam optimizer, particularly suitable for weight decay, which helps reduce overfitting. The learning rate was set to 1e-5. A smaller learning rate helps the model to train stably and converge to the global optimum. The batch size was set to 2 and the number of iterations was set to 10. Smaller batch sizes and numbers of iterations help enhance the model's generalization ability and reduce overfitting.We set the maximum length of the encoder to 256, which means the total length of the text pairs is 512, since the number of tokens in most text pairs is less than 512. We established the dropout layer ratio at 0.1 to prevent overfitting during fine-tuning [11].

We then continuously replaced the pre-trained models and their corresponding neural network frameworks to compare the performance of the trained models. By comparing the accuracy on the validation set, we found that the model trained with DeBERTa [8]had the highest accuracy on the easy and hard datasets, while the model trained with RoBERTa [10]had the highest accuracy on the medium dataset.

In machine learning, the batch size is a crucial hyperparameter that can significantly affect the model's convergence rate and final performance. A larger batch size usually provides a more stable gradient estimate but also increases memory consumption and may lead the model to become trapped in local optima. Based on these considerations, we increased the batch size by tenfold, hoping to improve model performance through more stable gradient estimation.

However, after a series of experiments, we found that increasing the batch size did not lead to the expected performance improvement. Specifically, after increasing the batch size, the model's accuracy on the validation set decreased by 10%, and the training time also increased. This indicates that for the current task and the adopted model architecture, a larger batch size may not be the optimal choice. Therefore, we ultimately set the batch size to 2.

3.4. Experiment results

We carried out four experiments: the BERT series pre-training models (DeBERTa-base, Deberta-v3-large, ALBERT-large-v2 and RoBERTa-base) were used to train on the training set, and the model with the highest accuracy was selected in the verification set to fine-tune the hyperparameters. Then we implement the model with the highest F1 score on the verification sets corresponding to different difficulty levels submitted by the TIRA platform [3]. The main experiment results are shown in Table 2.

Table 2
Overview of the F1 accuracy for the multi-author writing style task in detecting at which positions the author changes for task 1, tas 2, and task 3.

Approach	Task 1	Task 2	Task 3
cloying-tournament	0.991	0.815	0.818
Baseline Predict 1	0.466	0.343	0.320
Baseline Predict 0	0.112	0.323	0.346

We also compare the performance of different Bert-based models, denoted as DeBERTa-base, Deberta-v3-large, ALBERT-large-v2 and RoBERTa-base. The experiment results concerning accuracy are shown in Table 3, where Easy, Medium and Hard denote the dataset of different difficulty levels.

Table 3
Accuracy of different Bert-based models.

model validation(accuracy)	Easy	Medium	Hard
deberta-base	99.8%	85.4%	82.1%
deberta-v3-large	95.0%	78.7%	62.7%
albert-large-v2	99.3%	78.6%	82.1%
roberta-base	99.7%	86.2%	74.7%

4. Conclusion

In this study, we successfully developed a system based on DeBERTa-base and RoBERTa-base models to effectively process the multi-author writing style analysis task by using sliding window technique. By applying sliding Windows to text sequences, our method can effectively capture local context information and extract stylistic changes between paragraphs. On three datasets of different difficulty levels provided by the PAN organization,our approach achieves excellent scores on the f1 index, demonstrating its effectiveness and robustness.

Despite the positive results of our study, there are still some limitations and issues that need to be further explored. In experiments, it was found that increasing batch size did not improve performance, but led to a decrease in accuracy. This suggests that batch size has a significant impact on model performance, but its optimal value may depend on the specific task and model architecture and requires more in-depth research to determine. We are interested in experimenting with emerging pre-trained model architectures to explore whether they can bring further performance improvements for multi-author style detection tasks.

Acknowledgments

This research was supported by the National Social Science Foundation of China (22BTQ101) .

References

[1] J. Bevendorff, X. B. Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, M. Mayerl, A. Mukherjee, A. Panchenko, M. Potthast, F. Rangel, P. Rosso, A. Smirnova, E. Stamatatos, B. Stein, M. Taulé, D. Ustalov, M. Wiegmann, E. Zangerle, Overview of PAN 2024: Multi-Author Writing Style Analysis, Multilingual Text Detoxification, Oppositional Thinking Analysis, and Generative AI Authorship Verification, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. D. Nunzio, P. Galuščáková, A. G. S. de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2024.

[2] E. Zangerle, M. Mayerl, M. Potthast, B. Stein, Overview of the Multi-Author Writing Style Analysis Task at PAN 2024, in: G. Faggioli, N. Ferro, P. Galuščáková, A. G. S. de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2024.

[3] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. doi:10.1007/978-3-031-28241-6_20.

[4] I. H. Sarker, Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions, SN computer science 2 (2021) 420.

[5] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen, J. Yi, W. Zhao, X. Wang, Z. Liu, H.-T. Zheng, J. Chen, Y. Liu, J. Tang, J. Li, M. Sun, Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models, 2022. URL: https://arxiv.org/abs/2203.06904. arXiv:2203.06904.

[6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: https://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[7] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, J. Tang, Gpt understands, too, 2023. URL: https://arxiv.org/abs/2103.10385. arXiv:2103.10385.

[8] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, 2021. URL: https://arxiv.org/abs/2006.03654. arXiv:2006.03654.

[9] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, 2020. URL: https://arxiv.org/abs/1909.11942. arXiv:1909.11942.

[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[11] X. Liang, L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, T.-Y. Liu, R-drop: Regularized dropout for neural networks, 2021. URL: https://arxiv.org/abs/2106.14448. arXiv:2106.14448.