Team fosu-stu at PAN: Supervised Fine-Tuning of Large Language Models for Multi Author Writing Style Analysis

Notebook for the PAN Lab at CLEF 2024

Jiajun Lv, Yusheng Yi* and Haoliang Qi

Foshan University, Foshan, China

Abstract

This paper introduces large language models and label-supervised classification to address the Multi-Author Writing Style Analysis task. Large-scale pre-training and increased parameter sizes have endowed large language models with remarkable emergent capabilities, yet their performance on specific tasks still needs to improve. Our motivation is to leverage and exploit the capabilities of large language models in natural language processing tasks, enhancing their performance on specific tasks through label-supervised classification training.

Keywords

Multi-Author Writing Style Analysis, Large language models, Low-Rank Adaptation

1. Introduction

The rapid development of the internet has made plagiarism increasingly easy. Without reference corpora, multi-author writing style analysis is an effective method for detecting plagiarism[1]. Multi-author style analysis aims to identify changes in writing style within a document attributed to different authors. Research indicates that by analyzing an author's writing style, a document can be segmented into parts written by different authors, essentially performing an intrinsic style analysis task[1].

Since 2016, the PAN committee has organized an annual multi-author writing style analysis task. Participants must identify the positions of writing style changes, using variations in style and similarities in paragraph topics as indicators. In the PAN24: Multi-Author Writing Style Analysis task, participants need to address the following intrinsic style change detection task: identify all paragraph-level positions in a given text where there are changes in writing style[2].

In this study, we employ low-rank adaptation for efficient fine-tuning of large language models to achieve labeled supervised fine-tuning to address the PAN: Multi-Author Writing Style Analysis task within the CLEF 2024 challenge. This task will be conducted on three datasets, with increasing challenges as the similarity between paragraph topics increases.

2. Related Work

Analyzing recent Multi-Author Writing Style Analysis tasks[3][4], Ye et al. [5] used supervised contrastive learning techniques with p-tuning to enhance performance. Ahmad et al.[6] adopted data augmentation and multi-model fusion to improve model performance. Huang et al. [7] employed knowledge distillation to compress the teacher model mT0-large, leveraging the generalization capabilities of large language models to improve performance metrics. From recent years' methods, it is evident that models with larger base parameters and more complex techniques generally perform better.

Since the rise of large language models (LLMs) represented by ChatGPT, LLMs have shown great potential in natural language processing [8]. Previous studies [9][10][11] have utilized LLMs' in-context

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

*Corresponding author.

lvjiajun.96@gmail.com (J. Lv); yiys@fosu.edu.cn (Y. Yi); qihaoliang@fosu.edu.cn (H. Qi)
0000-0002-8755-5310 (J. Lv); 0009-0006-7098-3681 (Y. Yi); 0000-0003-1321-5820 (H. Qi)

learning capabilities for text classification and achieved significant results. However, generation-centered architectures may not capture task-specific patterns as effectively as label-supervised BERT[12] models. Inspired by the fine-tuned BERT family models on classification tasks, this study explores label-supervised fine-tuning based on LLMs, aiming to leverage their advantages in multi-author writing style analysis tasks. We compress the model using quantization techniques and low-rank adaptation methods to reduce the cost of model training and system deployment.

3. Data processing

In the PAN24 task of writing style analysis[2], participants are required to identify changes in writing style at the paragraph level and find all the locations where these changes occur. The organizers have strictly controlled the changes in author identity and topic, and provided datasets with three levels of difficulty. To achieve this goal, given a document D, we split it into multiple text segments based on line breaks, represented as the set $\{p_1, p_2, p_3, \ldots, p_n\}$. Then, we recombine each text segment with its adjacent segment to form $n - 1$ pairs of new text pairs, represented as the set $\{(p_1, p_2), (p_2, p_3), (p_3, p_4), \ldots, (p_{n-1}, p_n)\}$. For text pairs with a sequence length exceeding 512 characters, we truncate them evenly to 512 characters.

4. Method

Our approach is illustrated in the Figure 1. We use the LLaMA-3-8B decoder [13], obtaining vector representations from the last hidden layer of the LLaMA decoder. These representations are then mapped to the label space through a feedforward layer, generating probabilities used for label classification. The model is updated by calculating the cross-entropy loss and employing low-rank adaptation for fine-tuning.

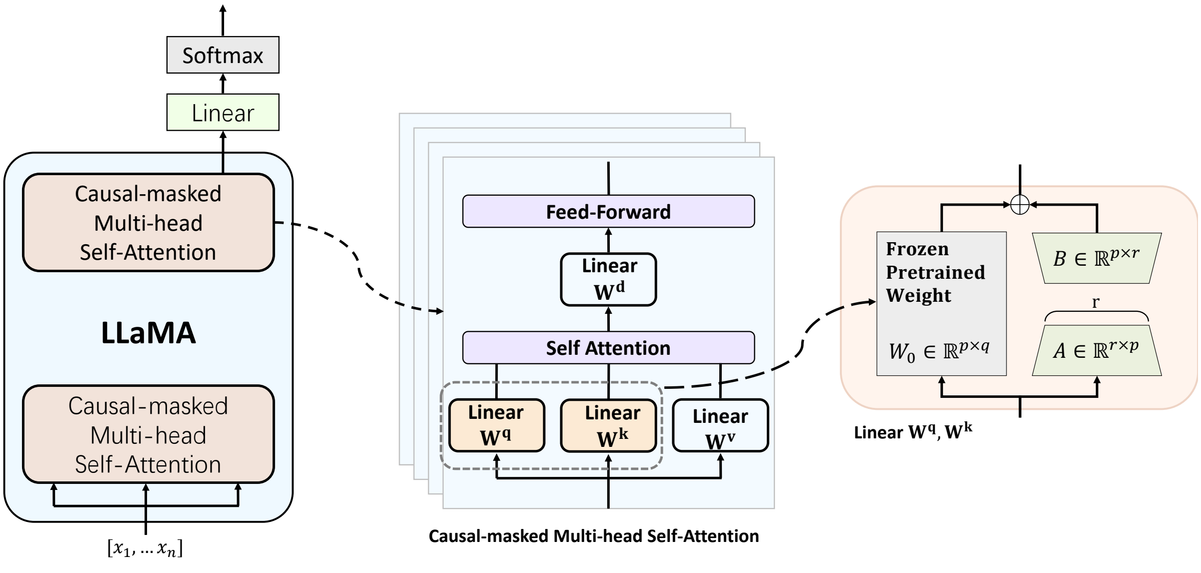
Softmax
Linear
Causal-masked Multi-head Self-Attention
LLaMA
Causal-masked Multi-head Self-Attention
$[x_1, \ldots x_n]$
Feed-Forward
Linear W^d
Self Attention
Linear W^q
Linear W^k
Linear W^v
Causal-masked Multi-head Self-Attention
Frozen Pretrained Weight
$W_0 \in \mathbb{R}^{p \times q}$
$B \in \mathbb{R}^{p \times r}$
r
$A \in \mathbb{R}^{r \times p}$
Linear W^q, W^k

Figure 1: Details of model architecture. Label supervised fine-tuning architecture for large language models.

4.1. Label supervision fine-tuning

Given the input text pairs (p_i, p_{i+1}), concatenate the two texts and feed them into the Tokenizer to perform byte-pair encoding to obtain the text encoding x_i. Then, input x_i into the decoder and extract the hidden state vector representation H_i for sequence classification.

$$x_i = Tokenizer(p_i, p_{i+1}) \tag{1}$$

$$H_i = LlamaModel(x_i) \tag{2}$$

Extract the last token vector from the hidden state vector H_i to serve as the vector representation h_i for sequence classification.

$$h_i = last(H_i) \tag{3}$$

The representation vector h_i of the sequence classification is fed into a linear layer and a softmax layer, where the vector representation$h_i$is mapped to the label space, resulting in an output probability distribution $p(y_i)$ Cross-entropy loss is calculated with the true label y_i, and the model parameters are updated.

$$p(y_i) = f_{Linear}(h_i) \tag{4}$$

$$\mathcal{L}_{ce} = -\frac{1}{N}\sum_{i=1}^{N} y_i{\cdot}log(p(y_i)) + (1 - y_i){\cdot}log(1 - p(y_i)) \tag{5}$$

4.2. Low-Rank Adaptation

The standard full fine-tuning paradigm requires thousands of GPUs working in parallel, which is very inefficient and unsustainable [14][15]. An algorithm, Parameter Efficient Fine-Tuning (PEFT), has been proposed, which aims at tuning the smallest parameters [14] to achieve better performance on full tuning of downstream tasks.

We adopted the low-rank decomposition method shown in Figure 2, where the original pretrained model weights are denoted as $W_0 \in \mathbb{R}^{d\times k}$. Through the low-rank decomposition $W_0+\Delta W = W_0+BA$, an additional parameter matrix BA is introduced into the self-attention matrices W_q and W_k, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, and the rank $r << \min(d, k)$. During training, we keep the pretrained model frozen, with only matrices A and B being updated.

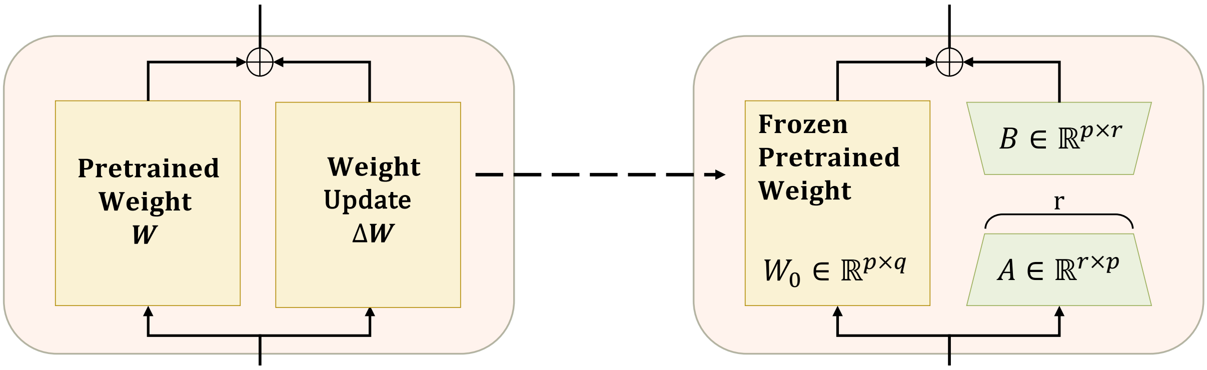
Pretrained Weight W
Weight Update ΔW
Frozen Pretrained Weight
$W_0 \in \mathbb{R}^{p\times q}$
$B \in \mathbb{R}^{p\times r}$
r
$A \in \mathbb{R}^{r\times p}$

Figure 2: Low-Rank Decomposition. For a pre-trained weight matrix W, restrict its updates to the form of low-rank decomposition.

5. Experiments

5.1. Dataset analysis

We conduct a positive and negative sample size analysis on the text pairs generated after data processing,The analysis results are shown in the Table1

Analysis reveals that the ratio of positive to negative samples in both the training and testing datasets is generally similar for each difficulty level. However, the distribution of positive and negative samples in the task1 easy dataset is unbalanced, with a ratio of 1:10.

Table 1
Statistics of the original dataset

Datasets	Training set		Validation set	
	#pos.	**#neg.**	**#pos.**	**#neg.**
task 1	10098	967	2219	252
task 2	12493	9420	2603	1989
task 3	8917	10098	1887	2248

5.2. Experience setting

In this paper, we chose Meta-Llama-3-8B as the pre-trained model and quantized it to int8. The model was trained on three different task datasets, resulting in models tailored to each task.Our hyperparameter settings are shown in Table 2:

Table 2
Fine-tuning tasks, the hyperparameters we use.

Hyperparameter	value
rank	128
alpha	128
dropout	0.1
batch size	64
max sequence length	512
initial learning rate	2e-5
epochs	3
warmup rate	0.1
target modules	w_q, w_v

We train and evaluate models on the A800 80GB GPU using the deep learning framework PyTorch and the efficient fine-tuning framework Peft[16].

6. Results

We use the fully fine-tuned deberta-base[17] as the baseline for our experiments, and the final indicators obtained by our method in the validation set are shown in Table 3

Table 3
Overview of the F1 accuracy for the multi-author writing style change detection on the validation set.

Approach	Task 1	Task 2	Task 3
our method	0.93	0.884	0.85
baseline	0.987	0.839	0.821

We finally submitted the model to the TIRA[18] platform for testing, and scored F1 for the three tasks respectively. The results are shown in Table 4 "alternating-vase" represents the fully fine-tuned deberta-base method, "quantum-ship" is the fine-tuning method based on this paper, "equilateral-commit" is a combination of both, using a voting method. The "camel-clef" involves modifying hyperparameters of target modules specifically to fine-tune the w_q, w_k, w_v, w_o weights. Our analysis reveals that the supervised fine-tuning of large language models surpasses the baseline in metrics for task2 and task3 but performs poorly on the task1 easy dataset. This poor performance may be related to the imbalance in the easy dataset distribution.

Table 4
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
presto-branch	0.944	0.887	0.834
alternating-vase	0.987	0.826	0.821
camel-clef	0.987	0.885	0.852
equilateral-commit	0.987	0.887	0.834
Baseline Predict 1	0.466	0.343	0.320
Baseline Predict 0	0.112	0.323	0.346

7. Conclusion

This paper proposes a method for detecting changes in writing style based on a large language model classifier, which uses label-supervised fine-tuning of the large language model. Additionally, we compress the model using LoRa and quantization methods to reduce training and inference costs. Experimental results show the effectiveness of supervised fine-tuning of the large language model in identifying multi-author style changes.

Acknowledgments

This research was supported by the Natural Science Foundation of Guangdong Province, China (No.2022A1515011544)

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. Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2024.

[3] J. Bevendorff, I. Borrego-Obrador, M. Chinea-Ríos, M. Franco-Salvador, M. Fröbe, A. Heini, K. Kredens, M. Mayerl, P. Pęzik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, E. Zangerle, Overview of PAN 2023: Authorship Verification, Multi-Author Writing Style Analysis, Profiling Cryptocurrency Influencers, and Trigger Detection, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, A. G. S. Vrochidis, D. Li, M. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 459–481. URL: https://doi.org/10.1007/978-3-031-42448-9_29. doi:10.1007/978-3-031-42448-9_29.

[4] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl, R. Ortega-Bueno, P. Pezik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, E. Zangerle, Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype

Spreaders, Style Change Detection, and Trigger Detection, in: A. Barrón-Cedeños, G. D. S. Martino, M. D. Esposti, F. Sebastiani, C. Macdonald, G. Pasi, A. Hanbury, M. Potthast, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. 13th International Conference of the CLEF Association (CLEF 2022), volume 13186 of *Lecture Notes in Computer Science*, Springer, 2022. URL: https://link.springer.com/book/10.1007/978-3-031-13643-6. doi:10.1007/978-3-031-13643-6.

[5] Z. Ye, C. Zhong, H. Qi, Y. Han, Supervised Contrastive Learning for Multi-Author Writing Style Analysis, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of CLEF 2023 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2023, pp. 2817–2822. URL: https://ceur-ws.org/Vol-3497/paper-237.pdf.

[6] A. Hashemi, W. Shi, Enhancing writing style change detection using transformer-based models and data augmentation, Working Notes of CLEF (2023).

[7] M. Huang, Z. Huang, L. Kong, Encoded classifier using knowledge distillation for multi-author writing style analysis, Working Notes of CLEF (2023).

[8] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, arXiv preprint arXiv:2303.18223 (2023).

[9] X. Sun, X. Li, J. Li, F. Wu, S. Guo, T. Zhang, G. Wang, Text classification via large language models, arXiv preprint arXiv:2305.08377 (2023).

[10] Y. Fei, Y. Hou, Z. Chen, A. Bosselut, Mitigating label biases for in-context learning, arXiv preprint arXiv:2305.19148 (2023).

[11] K. Margatina, T. Schick, N. Aletras, J. Dwivedi-Yu, Active learning principles for in-context learning with large language models, arXiv preprint arXiv:2305.14264 (2023).

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[13] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models (2023), arXiv preprint arXiv:2302.13971 (2023).

[14] Z. Han, C. Gao, J. Liu, S. Q. Zhang, et al., Parameter-efficient fine-tuning for large models: A comprehensive survey, arXiv preprint arXiv:2403.14608 (2024).

[15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685 (2021).

[16] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, B. Bossan, Peft: State-of-the-art parameter-efficient fine-tuning methods, https://github.com/huggingface/peft, 2022.

[17] 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.

[18] 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.