Supervised Contrastive Learning for Multi-Author Writing Style Analysis

Zhanhong Ye, Changle Zhong*[1], Haoliang Qi, Yong Han

Department of Electrical Engineering Foshan University, China

Abstract

This paper proposes supervised contrastive learning with a p-tuning method to solve the Multi-Author Writing Style Analysis task. The motivation is to capture more detail of the variation between the two paragraphs and exploit the potential performance of the pre-trained models. Therefore, we combine two methods, the Rdrop method and the supervised contrastive learning (SupCon) method, with p-tuning technology. Then trained and evaluated the three challenging Multi-Author Writing Style Analysis datasets (easy, medium, hard), which the PAN gave, and we achieved a score of 98.280, 83.035, 82.081 on each of the three difficulty tests set on the f1 metric. In addition, we conducted ablation experiments, which proved that supervised contrastive learning was beneficial in capturing more detailed changes in text and stimulating the potential of pre-trained models performance.

Keywords

Supervised Contrastive Learning, Multi-Author Writing Style Analysis, Soft Prompt, and Pre-trained Models

1. Introduction

Multi-author style identification is a task to discern whether two authors' writing styles are consistent. In detail, the style change detection task aims to identify text positions within a given multi-author document at which the author switches. If multiple authors have written a text, the task is to find evidence that we can detect variations in the writing style. Multi-Author Writing Style is widely used in plagiarism detection and author identification. In addition, style change detection can help to uncover gift authorships, verify a claimed authorship, or to develop new technology for writing support.

Supervised contrastive learning(SupCon)[1] method, which purpose of introducing label information into contrastive learning is to be able to use label information to use the same label as a positive sample and vice versa as a negative sample, and it has been applied in image classification and compared with the traditional unsupervised contrastive learning method at the time reached SOTA in the computer vision field. P-tuning[2], a soft-hard template method, is proposed, hoping that the model can learn how to represent the embedding of some words in the template through downstream tasks. Recently study[5] combines discrete prompts[4] with contrastive learning, using discrete prompts to help construct positive and negative examples in supervised contrastive learning. However, due to discrete prompts, the paper[2] has proved that in the context of manually setting prompts, different prompts will lead to different performances. Hence, there are better solutions than manually setting prompts. Therefore, based on the above, we combine p-tuning with supervised comparative learning to improve performance in completing the current task---the multi-author writing style analysis dataset released by PAN[6].

*[1] Corresponding author

[2]CLEF 2023 – Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

EMAIL: chinwang.yip@gmail.com (A. 1); 103409103@qq.com (A. 2); qihaoliang@fosu.edu.cn (A. 3); hanyong2005@fosu.edu.cn (A. 4)
ORCID: 0009-0001-4094-006X (A. 1); 0009-0008-3044-2383 (A. 2); 0000-0003-1321-5820 (A. 3); 0000-0002-9416-2398 (A. 4)

CEUR Workshop Proceedings (CEUR-WS.org) Proceedings

In this paper, we introduce a method that combines the P-tuning, Rdrop[3], and contrastive learning technology to aim for as simple an improvement as possible but deliver an excellent performance. In detail, the model has three-part. The first part is an lstm[7] model employed to learn the optimal prompt for the current downstream task. The second part is the deberta-v3[8] model, which handles the current task. Finally, the third part is classifier with contrastive learning loss.

2. P-tuning with supervised contrastive learning

2.1. Network Architecture

First, we regard the current task as a binary classification task. Given the model input, the goal is to use deberta-v3 to implement text classification tasks. Then the core part of the model introduces the p-tuning method to learn a soft-hard template according to the current downstream task and introduces supervised contrastive learning to use label information to make better feature representation.

According to the model shown in figure1, it consists of encoding, classification, contrastive learning, and p-tuning tasks. The first is the encoding part. We use the deberta-v3[8] model to encode the model, which is the transformer block and dropout layer shown in the figure. Noticed that the green and red spots indicate that the same model input was encoded by the pre-trained model, but the encoded result is different because of the existence of the dropout layer. There will be a difference between the output of two identical inputs because the dropout layer will randomly drop something, making the output of the two inputs have some subtle differences. Next comes the classification part, where we use linear layers and classifiers to classify the encoded content, making it possible to complete the current downstream task. Then is the comparative learning part, which is the part that encloses $\mathcal{L}_{out}^{sup}$ in the frame, as shown in the figure. After obtaining the hidden state, we perform supervised contrastive learning calculations. In addition, we also use the rdrop method on the obtained hidden state to calculate the situation when two identical samples are used as a positive pair or a negative sample pair (this situation is not calculated in the SupCon method). That is the part including $\mathcal{L}_{RD}$, as described in Figure 1. See section 2.2 for details. Finally is the p-tuning method, as shown in Figure 1. The part including p-tuning is the method of using p-tuning. Details are in section 2.3.

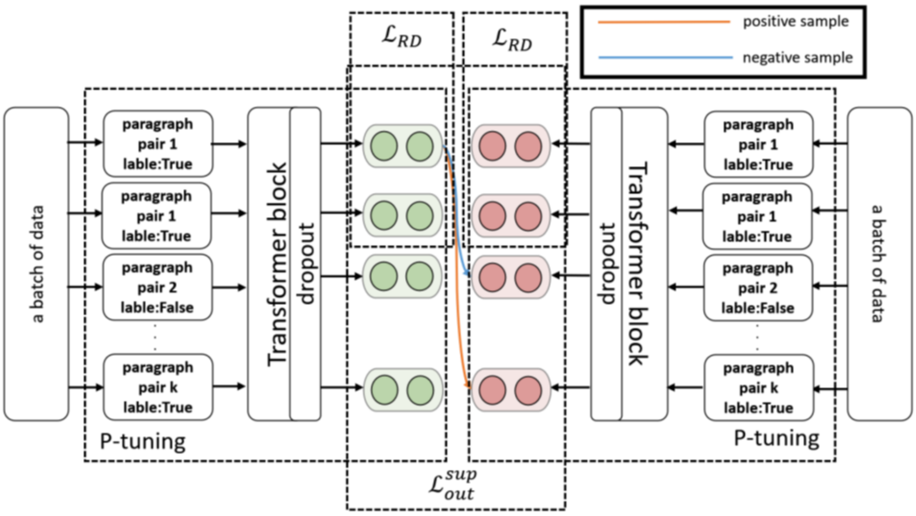
$\mathcal{L}_{RD}$
$\mathcal{L}_{RD}$
positive sample
negative sample
a batch of data
paragraph pair 1 lable:True
paragraph pair 1 lable:True
paragraph pair 2 lable:False
paragraph pair k lable:True
Transformer block
dropout
P-tuning
dropout
Transformer block
paragraph pair 1 lable:True
paragraph pair 1 lable:True
paragraph pair 2 lable:False
paragraph pair k lable:True
a batch of data
P-tuning
$\mathcal{L}_{out}^{sup}$

Figure 1 Model Architecture

Overall the primary correlate loss function can be defined as follow.

$$\mathcal{L} = \mathcal{L}_{out}^{sup} + \mathcal{L}_{RD} \tag{1}$$

The loss $\mathcal{L}_{out}^{sup}$ means contrastive learning loss and the loss $\mathcal{L}_{RD}$ represents Rdrop method loss.

2.2.Supervised contrastive learning

Firstly, the PAN has given three complex data sets for Multi-Author Writing Style Analysis. Then, we directly form a paragraph pair of two paragraphs and combine them with the soft template(section 2.3) as input to the model. Then we counted the token length of each paragraph pair which did not include the soft template in the dataset for all difficulties, including the training set and development set. The statistical results show that the length of most of the data is less than 512.

Give a batch name as $\mathcal{B}$. The contents of $\mathcal{B}$ can define as $\{(S_1, y_1), (S_2, y_2) \ldots (S_k, y_k)\} \in \mathrm{B}$, where S_k means the paragraph pair, and y_k is the correlate label. Then we extend the batch by copying each paragraph in the batch, naming it as $\mathcal{B}'$ $\mathcal{B}'$ can define as $\{(S_1, y_1), (S_1, y_1), (S_2, y_2), (S_2, y_2) \ldots (S_k, y_k)\} \in \mathcal{B}'$.Then we combine the expanded batch with the soft-hard template to get $\widehat{\mathrm{B}}$, $\widehat{\mathrm{B}}$ can be defined as $\{(S_1 + r_1, y_1), (S_1 + r_1, y_1), (S_2 + r_2, y_2), (S_2 + r_2, y_2) \ldots (S_k + r_\mathrm{k}, y_k)\} \in \mathcal{B}'$, where r_k is a soft-hard template.Then we feed the $\widehat{\mathrm{B}}$ into the pre-training model, which is composed of the transformer[8] block and dropout layer in the figure for encoding to get the corresponding hidden state $\mathcal{H}$. $\mathcal{H}$ can be defined as $\{h_1, h_1', h_2, h_2' \ldots h_k, h_k'\}$,$h_k$ *and* h_k' mean the hidden state of paragraph S_k and copied sentence S_k. Once $\mathcal{H}$ is obtained, we calculate $\mathcal{L}_{out}^{sup}$ and $\mathcal{L}_{RD}$.

After getting hidden state $\mathcal{H}$, we calculate the supervised contrastive learning loss(equation 2). First, we define the same label as a positive sample and vice versa as a negative example. However, the difference is that for S_k and its corresponding copied S_k, although their labels are the same, they do not participate in the calculation of supervised contrastive learning in this case. Instead, we use the Rdrop[3] method to calculate the loss (equation 3). After defining the positive and negative samples, we use formula 2 to calculate the supervised contrastive learning loss.

$$\mathcal{L}_{out}^{sup} = \sum_{i=1}^{k} \frac{-1}{K} \sum_{j \in J(i)} log \frac{exp\,(h_i \cdot h_j / \tau)}{\sum_{p \in A(i)} exp\,(h_i \cdot h_p / \tau)} \tag{2}$$

The index $j \in J(i)$ means the paragraph corresponding to index j should have the same label as the paragraph corresponding to the current index i and index i ≠ j. This is done to form a sample of positive examples. Similarly, the part p∈A(i) indicates that the paragraph pair corresponds to the index of p ≠ i. This is done to form a sample of negative examples. The numerator part of the fraction calculates the distance of each positive example with an exponential function, and the denominator part is the sum of the distance with the exponential function of each negative example in the batch.

Next is the rdrop formula,

$$\mathcal{L}_{RD} = -\log p_1(y_k|(S_k + \mathrm{p'}_\mathrm{k})) - \log p_2(y_k|(S_k + \mathrm{p'}_\mathrm{k})) + \frac{\alpha}{2}[D_{kl}(p_1(y_k|(S_k + \mathrm{p'}_\mathrm{k})) \mid\mid p_2(y_k|(S_k + \mathrm{p'}_\mathrm{k}))] + D_{kl}(p_2(y_k|(S_k + \mathrm{p'}_\mathrm{k})) \mid\mid p_1(y_k|(S_k + \mathrm{p'}_\mathrm{k})) \tag{3}$$

where α is a hyperparameter, kl means Kullback–Leibler divergence[10]. Then $p_1(y_k|(S_k + \mathrm{p'}_\mathrm{k}))$ and $p_2(y_k|(S_k + \mathrm{p'}_\mathrm{k}))$ represent the probability distribution of the i-th paragraph and the probability distribution after copying the i-th paragraph. Among them, $\mathrm{p'}_\mathrm{k}$ represents the soft-hard template, and the plus sign in $S_k + \mathrm{p'}_\mathrm{k}$ represents the original input of the model combined with the soft-hard prompt template. For the combination method, see section 2.3, and the part $-\log p_1(y_k|(S_k + \mathrm{p'}_\mathrm{k}))$ is used to calculate the loss on the current task.

2.3.P-tuning

We directly use the method described by p-tuning to build the soft-hard template. Given a manual prompt, we define it as p . Then p consists of the following words $\{w_1\ w_2\ \ldots\ w_\mathrm{k}\}$, w_k means one of the words in the prompt. Then we manually replace the words in p with learnable tokens, and the replaced result is p' , which consists of the following words $\{r_1\ w_2\ \ldots\ r_\mathrm{i}\}$, r_i means we manually replace the token at position i with a learnable token r_i. After getting p', we combine paragraph S_k and p', and the combination method can be a cloze form or prompt as a prefix for model input. We use a cloze-style approach to combine prompt and model inputs, and we mark this paradigm as $S_k + \mathrm{p}'_\mathrm{k}$. Then we get the final input of the model and then feed this input to the transformer block to get the corresponding hidden state. Finally, we map the hidden state onto a vocabulary space through the linear layer and get the probabilities of 'yes' and 'no'.

3. Experiments and Result

3.1. Data statistics

The PAN provides all data. The data is available in three difficulty levels: easy, medium, and hard. Each difficulty data set is divided into a training set, a development set, and a test set. The distribution of each dataset is 70%, 15%, and 15%, respectively. Organizing the data according to the method mentioned in section 2.2 will result in the following number of paragraph pairs, as Table 1 shows.

Table 1
the statistical result

	dataset1	dataset2	dataset3
Train-set	12904	28216	19113
Dev-set	2828	7042	4112

3.2. Experience setting

In this work, the deberta-v3-base model is selected for use in the p-tuning technique. It concludes with 12 transformers[9] layers and 12 attention heads and its hidden size is 768. Table 2 shows the detail of the hyperparameter. We set the early stopping to 10, setting the learning rate to 2e-5 and the Rdrop alpha coefficient to 4, and the supervised contrastive learning temperature coefficient to 70.

3.3. Results

We will present two experiments, which are the main experiment and the ablation experiment. The main experiment is the best result achieved so far, and the ablation experiment is to investigate how different settings affect the performance of the model. We use the method of section 2.2 to construct the model input for our proposal, but the difference is that for fine-tune Bert, our input does not use the template.

Table 2
the best score in different difficulty development set

	dataset1@ F1-SCORE	dataset2@ F1-SCORE	dataset3@ F1-SCORE
Our method	99.088	83.034	82.0
Fine-tune bert	96.446	79.574	78.051

In this experiment, we report an ablation experiment on our model. The following three experiments show the performance of our proposed method with the SupCon method removed. The performance obtained by removing the SupCon and Rdrop methods, i.e., the performance of the plain p-tuning method.

Table 3
the score in the different difficulty test set

	dataset1@ F1-SCORE	dataset2@ F1-SCORE	dataset3@ F1-SCORE
Test set	98.280	83.035	82.081

Table 4
ablation experiment in different difficulty development sets

	dataset1 @ F1-SCORE	dataset2 @ F1-SCORE	dataset3 @ F1-SCORE
Our method	99.078	83.034	82.0
Without SupCon	96.902	81.016	78.623
Without SupCon & Rdrop	96.849	81.407	73.448

4. Conclusion

In this paper, we have accomplished the tasks mentioned by PAN[11][12], and we propose a method that combines three types of technology to solve the multi-Author Writing Style Analysis task. To solve this task, we propose a method that combines Rdrop, supervised contrastive learning, and p-tuning technology. The proposed method obtains 98.280, 83.035, 82.081 on three datasets of varying difficulty. This validates the ability of our proposed method to accomplish the Multi-Author Writing Style Analysis task.

5. Acknowledgments

This work is supported by the National Social Science Foundation of China (No. 22BTQ101).

6. References

[1] P. Khosla, P. Teterwak, C.Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, Advances in neural information processing systems 33 (2020) 18661–18673.

[2] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, J. Tang, Gpt understands, too, arXiv preprint arXiv:2103.10385 (2021).

[3] L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, T.-Y. Liu, et al., R-drop: Regularized dropout for neural networks, Advances in Neural Information Processing Systems 34 (2021) 10890 – 10905.

[4] T. Schick, H. Schütze, Exploiting cloze questions for few shot text classification and natural language inference, arXiv preprint arXiv:2001.07676 (2020).

[5] Y. Jian, C. Gao, S. Vosoughi, Contrastive learning for prompt-based few-shot language learners, arXiv preprint arXiv:2205.01308 (2022).

[6] E. Zangerle, M. Mayerl, M. Potthast, B. Stein, Overview of the Multi-Author Writing Style Analysis Task at PAN 2023, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of CLEF 2023 - Conference and Labs of the Evaluation Forum, CEUR-WS,2023.

[7] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997)1735 – 1780.

[8] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, arXiv preprint arXiv:2111.09543 (2021).

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[10] J. M. Joyce, Kullback-leibler divergence, in: International encyclopedia of statistical science, Springer, 2011, pp. 720 – 722.

[11] 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. Stefanos 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, 2023.

[12] 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.