# Team Chen at PAN: Integrating R-Drop and Pre-trained Language Model for Multi-author Writing Style Analysis

Notebook for the PAN Lab at CLEF 2024

Zhaotian Chen, Yong Han* and Yusheng Yi

*Foshan University, Foshan, China*

**Abstract**

This paper presents our experiment in the PAN Multi-Author Writing Style Analysis task at CLEF 2024. The task is divided into three increasingly difficult subtasks according to the topic consistency between paragraphs: from detecting style changes between paragraphs with multiple topics at the easy level, to a medium level where the diversity of topics is small, forcing the method to focus more on style, finally, at the most difficult level to identify subtle style differences between paragraphs of the same topic. Therefore, the task asks for not only distinguishing different topics but also capturing obvious change in writing style with the same topic. To address the task, we select the powerful pre-trained language model, Roberta, as the foundation model and fine-tuned it to detect styles and topics of texts. Additionally, we employed R-Drop regularization to reduce overfitting during the model fine-tuning, thereby enhancing its generalization capabilities on unseen texts. Experimental results demonstrate that our model achieved F1 scores of 0.968, 0.822, and 0.807 on the test sets of the three difficulty levels, respectively.

**Keywords**

PAN 2024, Multi-Author Writing Style Analysis, Regularization, R-Drop, Pre-trained language Model

## 1. Introduction

The task of multi-author writing style analysis aims to find all positions of writing style change on the paragraph-level in a given multi-author document [1]. Therefore, detecting style changes can assist in identifying the identity of the current author, verifying the claimed authorship, and detecting the risk of plagiarism in documents. Particularly in situations where there is no comparative text, detecting style changes becomes the sole method to identify plagiarism in documents.

In recent years, PAN [2] has organized a series of tasks to detect writing style changes in text, ranging from determining the actual number of authors[3], identifying style changes between two consecutive paragraphs[4, 5], to detecting style changes at the sentence level[6](ranging from detecting style of consecutive paragraphs to consecutive sentences). In this year, the task of Style Change Detection focuses on paragraphs and detects writing style changes at every pair of consecutive paragraphs in a given text.

## 2. Related Work

Large-scale pre-trained models, such as BERT[7], RoBERTa[8], etc., often contain millions or even billions of parameters. Although larger models tend to exhibit better performance, they are highly susceptible to overfitting. During the fine-tuning process of the pre-trained model for the style change detection task, it was observed that despite a continuous decrease in training loss, the F1 score on the validation set remained unsatisfactory. Upon closer examination of both the training and validation losses, it was revealed that while the training loss was steadily declining, the validation loss was progressively increasing. To address this issue, researchers have proposed various regularization

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

*Corresponding Author

1353663548z@gmail.com (Z. Chen); hanyong2005@fosu.edu.cn (Y. Han); yiys@fosu.edu.cn (Y. Yi)

0009-0008-3734-7442 (Z. Chen); 0000-0002-9416-2398 (Y. Han); 0009-0006-7098-3681 (Y. Yi)

**Table 1**
Datasets statistics. “Dataset1”, “Dataset2”, “Dataset3”, correspond to tasks of easy, medium, and difficult levels, respectively.

| Datasets | Dataset1 | | Dataset2 | | Dataset3 | |
|---|---|---|---|---|---|---|
| | Documents | Samples | Documents | Samples | Documents | Samples |
| Training set | 4200 | 11065 | 4200 | 21914 | 4200 | 19014 |
| Validation set | 900 | 2468 | 900 | 4590 | 900 | 4132 |

methods, including weight decay [9, 10, 11], dropout [12, 13, 14, 15], normalization [16, 17, 18], adding noise [19], layer-wise pre-training and initialization [20, 21], label smoothing [22], and more. Among these methods, dropout and its variants have garnered significant attention due to their effectiveness and compatibility with other regularization techniques.

Dropout enhances the generalization capability by inhibiting the co-adaptation of neurons and implicitly creating an ensemble of multiple sub-models. As a variant of dropout, compared to the traditional dropout strategy in neural network training, The core idea of R-Drop regularization[23] lies in generating consistent predictions from models with different dropout masks during the training process.

As a modified version of dropout, R-Drop regularization, in contrast to the conventional dropout approach employed in neural network training, centers on ensuring consistent predictions from various dropout-masked networks throughout the training phase [23]. In order to do this, R-Drop employs the minimization of Kullback-Leibler divergence between the outputs of any two sub-models using different dropout masks to achieve model regularization. The method greatly improves models ability of generalization and lowers the risk of overfitting by effectively reducing the degree of freedom of model parameters. Consequently, it significantly enhances the stability and generalization capability of reasoning.

## 3. Dataset

The task presents three datasets of varying difficulty levels, categorized based on the diversity of topics and consistency within the documents. Each dataset poses specific subtasks.

- **Easy**: The paragraphs of a document cover a variety of topics, allowing approaches to make use of topic information to detect authorship changes.
- **Medium**: The topical variety in a document is small (though still present), forcing the approaches to focus more on style to effectively solve the detection task.
- **Hard**: All paragraphs in a document are on the same topic.

Each dataset is divided into three parts: training set, validation set and testing set. The training set and the validation set include ground truth data, while the testing set does not provide ground truth data. Table 1 provides statistical information about the datasets. Note that "Samples" specifically refers to data units composed of two consecutive paragraphs from the documents, used to analyze whether there is a style change between the two paragraphs. For details on how the samples were constructed, please refer to Section 4.1.

## 4. Methodology

The methodology presented in this paper encompasses three primary steps: 1) data preparation, 2) R-Drop regularization, and 3) model fine-tuning. The methodology is founded on the concept of attaining elevated precision and recall rates for classifying unseen datasets. To accomplish this goal, fine-tuning of a pre-trained language model is undertaken for specific downstream tasks. Furthermore, R-Drop regularization methods are employed to enhance the model’s generalization capabilities.

### 4.1. Data Preparation

To create the samples, we first marked the junction between two consecutive paragraphs in each document using delimiters. Subsequently, we assigned binary labels indicating whether there was a style change between the two paragraphs. This enabled us to transform the task into a binary classification problem. In order to prepare the samples for fine-tuning the pre-trained RoBERTa model, we adopted the corresponding tokenizer for RoBERTa. RoBERTa has a limit on the maximum input sequence length, typically 512 tokens. Upon analyzing the dataset, we found that only a few samples exceeded the maximum token limit. Therefore, we opted for a truncation strategy to handle samples exceeding the maximum input sequence length.

### 4.2. R-Drop Regularization

Dropout randomly drops part of units in each layer of the neural network to avoid co-adapting and over-fitting. Besides, dropout also approximately performs to combine exponentially many different neural network architectures efficiently, while model combination can always improve the model performance. Despite its simplicity and efficacy, dropout introduces a significant inconsistency between the training and inference phases, which can potentially impede model performance. To address this issue, the incorporation of the R-Drop regularization term into the training process ensures consistency in the model's predictions for identical inputs across varying dropout masks. This approach regulates the inconsistency arising from dropout during training. Specifically, for each training batch, the process involves conducting two forward passes with distinct dropout masks on the same data batch. Subsequently, the Kullback-Leibler (KL) divergence, a widely used metric for quantifying the disparity between two probability distributions, is computed between the two prediction outcomes.

Given the input $x_i$, $P_{w1}(y_i|x_i)$ and $P_{w2}(y_i|x_i)$ represent the probability distributions of $y_i$ predicted by the model under different sets of parameters (caused by dropout, such as $w1$ and $w2$), respectively. The KL divergence between $P_{w1}(y_i|x_i)$ and $P_{w2}(y_i|x_i)$ is given by:

$$D_{\mathrm{KL}}(P_{w1}(y_i|x_i)\|P_{w2}(y_i|x_i)) = \sum_{y_i} P_{w1}(y_i|x_i) \log \frac{P_{w1}(y_i|x_i)}{P_{w2}(y_i|x_i)} \tag{1}$$

To incorporate this discrepancy into the training process, R-Drop add the calculated KL divergence as an important regularization term to the loss function and use the parameter $\alpha$ to control the coefficient weight of KL divergence. The R-Drop method employs a loss function represented by the formula given below. In the formula 2, the terms $-\log P_\theta^{(1)}(y_i|x_i)$ and $-\log P_\theta^{(2)}(y_i|x_i)$ signify the negative log probabilities of accurately predicting the label $y_i$ conditional on the input $x_i$. These probabilities are obtained from two sub-models, both of which are generated by introducing dropout variations to the same neural network.

$$L = -\log P_\theta^{(1)}(y_i|x_i) - \log P_\theta^{(2)}(y_i|x_i) + \alpha D_{KL}(P_\theta^{(1)}(y_i|x_i)||P_\theta^{(2)}(y_i|x_i)) \tag{2}$$

### 4.3. Model Fine-Tuning

In the task, the RoBERTa pre-trained model was chosen as the base model to leverage the rich linguistic representations it has learned from a large corpus, enhancing the performance of downstream tasks. To reduce the risk of model overfitting, a Dropout layer was introduced after the output of the RoBERTa model. Dropout is a widely used regularization technique that randomly discards neurons to reduce model complexity and enhance model generalization. During model fine-tuning, R-Drop regularization was employed to further improve the model's generalization capability. To adapt the pre-trained model to the style change detection task, a fully connected linear output layer was added on top of the model. This output layer uses the softmax activation function to generate a probability distribution for each category, enabling the model to learn and classify whether there is a style change between consecutive paragraphs.

Algorithm 1 details the implementation of R-Drop in the model fine-tuning process. Specifically, for each training batch, two forward passes with different dropout masks are performed on the same batch of data, and the Kullback-Leibler (KL) divergence between the two prediction results is calculated. Through this approach, the model not only takes full advantage of the language representation capabilities of the RoBERTa model but also addresses the issue of overfitting. This enables the model to effectively identify and classify style changes in unseen texts, thereby enhancing the overall performance of the task.

**Algorithm 1** Fine-tuning with Integrated R-Drop Algorithm

**Require:** Training dataset $D = \{(x_i, y_i)\}$
**Require:** Neural network model $M$ with parameters $\theta$
**Require:** Number of training epochs $E$
**Require:** Learning rate $\eta$
**Require:** Balance factor $\alpha$ for KL divergence loss
**Ensure:** Trained model $M$ with optimized parameters $\theta^*$

Initialize model parameters $\theta$ randomly or with pre-trained weights
Initialize optimizer with learning rate $\eta$
**for** epoch $e = 1$ to $E$ **do**
    **for** each batch $(X_b, Y_b)$ in $D$ **do**
        Forward pass the model $M$ twice with $X_b$ to obtain two sets of outputs: $O_1$ and $O_2$
        Compute cross-entropy losses: $CE_1 = CE(Y_b, O_1), CE_2 = CE(Y_b, O_2)$
        Compute KL divergence loss: $KL = \alpha \cdot KL(O_1 \parallel O_2) + \alpha \cdot KL(O_2 \parallel O_1)$
        Compute total loss: $L = CE_1 + CE_2 + KL$
        Backpropagate the total loss $L$ to update model parameters $\theta$
    **end for**
**end for**
**return** trained model $M$ with optimized parameters $\theta^*$

# 5. Experiments

## 5.1. Experimental settings

In this paper, the RoBERTa model was chosen, comprising 12 transformer layers, 768 hidden units, and 12 attention heads. The hyperparameter settings are as follows: the maximum sequence length is set to 512, the learning rate is set at 0.00001, the batch size is configured to 32, the number of epochs is set to 7, and the dropout rate is 0.5. The coefficient weight $\alpha$ for R-Drop is set to 5.

To evaluate the effectiveness of the model for each subtask, performance is assessed by calculating the F1 score on the provided evaluation set. After conducting experiments and obtaining results on the evaluation set, the best-performing model for each subtask is selected.

## 5.2. Result

The best-performing model for each sub-task was ultimately submitted to TIRA [24] for execution, and the final performance indicators of the model were obtained. Table 2 provides the F1 scores achieved by the model on the official test set. Compared to the baseline approaches provided by Pan, which predict either no style change (all-0) or all style changes (all-1), our method achieves a minimum improvement of 1-fold and a maximum improvement of up to 7-fold.

## 5.3. Ablation experiments

To demonstrate the effectiveness of R-Drop in this task, we conducted experiments while keeping other parts of the model unchanged. We observed the performance changes of the model in the validation set by adding or not adding the R-Drop method. Table 3 presents the experimental results.

**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 |
|---|---|---|---|
| Our Method | 0.968 | 0.822 | 0.807 |
| Baseline Predict 1 | 0.466 | 0.343 | 0.320 |
| Baseline Predict 0 | 0.112 | 0.323 | 0.346 |

**Table 3**
F1 score comparison on validation datasets.Set dropout to 0.5, if the dropout parameter is available.

| | Task1 | Task2 | Task3 |
|---|---|---|---|
| Roberta With R-Drop | 0.975 | 0.833 | 0.815 |
| Roberta With dropout | 0.951 | 0.830 | 0.810 |
| Roberta Without dropout | 0.948 | 0.822 | 0.782 |

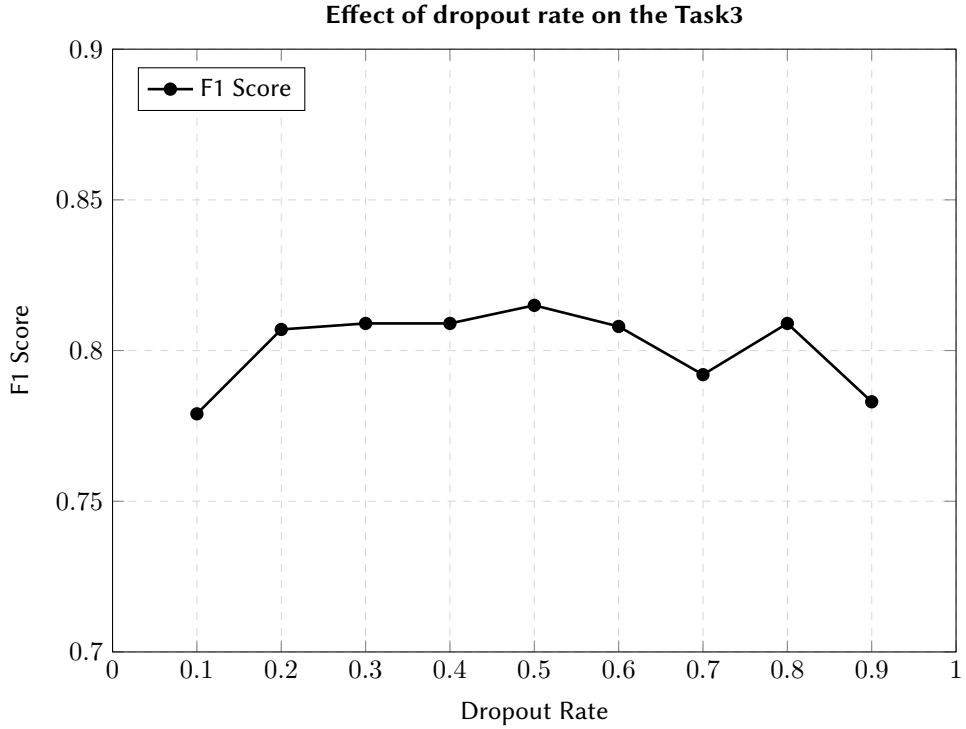


**Figure 1:** Effect of dropout rate on the Task3.

In Figure 1, we adjusted the dropout parameter in the model with other parameters unchanged, in order to optimize the model's performance on complex datasets.

## 6. Conclusion

This paper briefly introduces our work achievements on the PAN 2024 multi-author writing style analysis task. We fine-tune the RoBERTa model using the R-Drop regularization method to obtain the final results. This approach achieved promising outcomes across three subtasks of varying difficulty levels, demonstrating its effectiveness in tackling complex writing style analysis challenges. An ablation study further validated the significance of R-Drop regularization in preventing overfitting and enhancing model performance.

However, the lack of analysis on error cases limits our understanding of the model's limitations and potential for improvement. Future research should delve deeper into error case analyses to identify the model's weaknesses and devise targeted solutions.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62276064)

## References

[1] E. Zangerle, M. Mayerl, M. Potthast, et al., 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.

[2] J. Bevendorff, X. B. Casals, B. Chulvi, et al., 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.

[3] E. Zangerle, M. Tschuggnall, G. Specht, et al., Overview of the Style Change Detection Task at PAN 2019, in: L. Cappellato, N. Ferro, D. Losada, H. Müller (Eds.), CLEF 2019 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2019. URL: http://ceur-ws.org/Vol-2380/.

[4] E. Zangerle, M. Mayerl, G. Specht, et al., Overview of the Style Change Detection Task at PAN 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/.

[5] E. Zangerle, M. Mayerl, M. Potthast, et al., Overview of the Style Change Detection Task at PAN 2021, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[6] E. Zangerle, M. Mayerl, M. Potthast, et al., Overview of the Style Change Detection Task at PAN 2022, in: G. Faggioli, N. Ferro, A. Hanbury, M. Potthast (Eds.), CLEF 2022 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2022. URL: http://ceur-ws.org/Vol-3180/paper-186.pdf.

[7] J. Devlin, M.-W. Chang, K. Lee, et al., Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North, 2019. URL: http://dx.doi.org/10.18653/v1/n19-1423. doi:10.18653/v1/n19-1423.

[8] Y. Liu, M. Ott, N. Goyal, et al., Roberta: A robustly optimized bert pretraining approach, Cornell University - arXiv,Cornell University - arXiv (2019).

[9] A. Krogh, J. Hertz, A simple weight decay can improve generalization, Neural Information Processing Systems,Neural Information Processing Systems (1991).

[10] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Communications of the ACM (2017) 84–90. URL: http://dx.doi.org/10.1145/3065386. doi:10.1145/3065386.

[11] W. Wen, C. Wu, Y. Wang, et al., Learning structured sparsity in deep neural networks, Neural Information Processing Systems,Neural Information Processing Systems (2016).

[12] G. Hinton, N. Srivastava, A. Krizhevsky, et al., Improving neural networks by preventing co-adaptation of feature detectors, Cornell University - arXiv,Cornell University - arXiv (2012).

[13] L. Wan, M. Zeiler, S. Zhang, et al., Regularization of neural networks using dropconnect, International Conference on Machine Learning,International Conference on Machine Learning (2013).

[14] J. Ba, B. Frey, Adaptive dropout for training deep neural networks, Neural Information Processing Systems,Neural Information Processing Systems (2013).

[15] S. Wang, C. Manning, Fast dropout training, International Conference on Machine Learning,International Conference on Machine Learning (2013).
[16] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv: Learning,arXiv: Learning (2015).
[17] L. Huang, X. Liu, B. Liu, et al., Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks, National Conference on Artificial Intelligence,National Conference on Artificial Intelligence (2017).
[18] Y. Wu, K. He, Group normalization, International Journal of Computer Vision (2020) 742–755. URL: http://dx.doi.org/10.1007/s11263-019-01198-w. doi:10.1007/s11263-019-01198-w.
[19] S. Hochreiter, J. Schmidhuber, Simplifying neural nets by discovering flat minima, Neural Information Processing Systems,Neural Information Processing Systems (1994).
[20] D. Erhan, P.-A. Manzagol, Y. Bengio, et al., The difficulty of training deep architectures and the effect of unsupervised pre-training (2009).
[21] K. He, X. Zhang, S. Ren, et al., Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015. URL: http://dx.doi.org/10.1109/iccv.2015.123. doi:10.1109/iccv.2015.123.
[22] C. Szegedy, V. Vanhoucke, S. Ioffe, et al., Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. URL: http://dx.doi.org/10.1109/cvpr.2016.308. doi:10.1109/cvpr.2016.308.
[23] L. Wu, J. Li, Y. Wang, et al., R-drop: Regularized dropout for neural networks, Advances in Neural Information Processing Systems 34 (2021) 10890–10905.
[24] M. Fröbe, M. Wiegmann, N. Kolyada, et al., Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: 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.